Cloud Computing -- It is the use of remote servers on the internet to store,manage and process data rather than a local server or your personal computer.

so , you do three things ; you store , you manage, and your process data

Store - you are storing a file, say on a file system on the Cloud

Manage - you are managing a data using database there on the cloud

Process- you are using computing power on the cloud to process your data

for huge data you can always rent a server from AWS, with the right kind of configuration and you and use that machine to process the data or process the file that you want to process and once you are done with that you can always terminate your machine and will pay AWS according to the number of hours that you have used for this.

you can rent that server from AWS , use it and pay according to your usage.

AWS is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help business scale and grow

Put your data on Cloud Servers . No more buying expensive servers

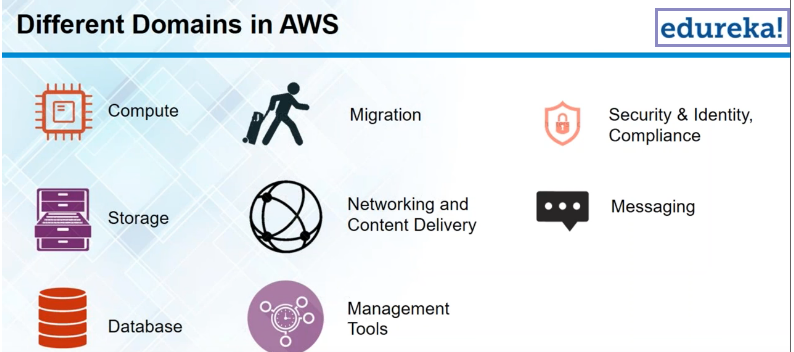
Scalablity - Your server capacity will vary accroding to traffic

Your cloud provider will manage your servers , hence no worries about the underlying infrastructure

Aws - 2006 Azzure- 2010

AMI (Amazon Machine Image) is Operating System

Storage for Windows minimum of 30 GB and we can expend based on the requirement



**Compute:**

**EC2 Elastic Compute Cloud:** Its just like a raw server, So we can configure this server to be anything , you can use it host a website , work at a environment , its just like a new PC that you buy , So install a fresh Operating System on your PC and then you can configure it to be anything and solve any software you want and then it can serve you as you require.

L**ambda :** Lambda is advanced version of EC2 **,** its based on EC2 but the difference between the Lambda & EC2 is , Lambda cannot be used to host in your application , Lambda only to execute your background tasks.

**Elastic Beanstalk:**

Elastic Beanstalk used to host an application but **Lambda** is not used to host an application

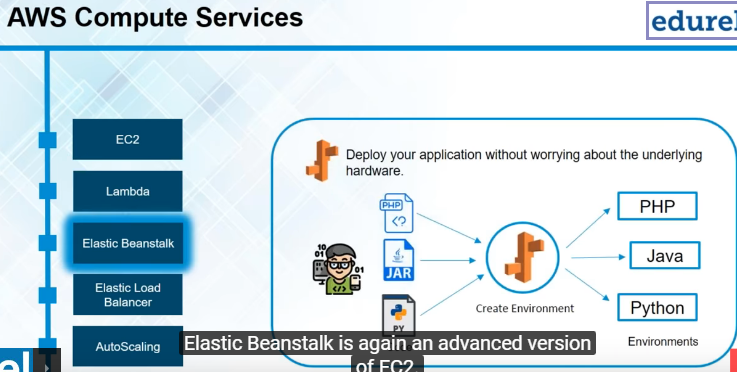
Elastic Beanstalk don’t have to configure in all the details or don’t have to set up environment.

Elastic Beanstalk is having limited number of environments.

**Elastic Load Balacer :** is used to distribut your work load among a number of instances. Suppose have 5 servers are running and all the traffic is directed to the 1st instance and it doesn’t make sense all other 4 servers are idle. Now the the traffic is distributed these 5 instances and the protocol is ELB . ELB distributes the workload equally among the instances.

**AutoScaling :** is a service which is used to scale up and down automatically without your manual intervention. If u are using Autoscaling you have to use Load Balancer.

Autoscaling and ELB have to used together

****

**Migration :**

When you want to transfer your data to the AWS infrastructure or you want to transfer your data back from AWS infrastructure . So if you have petabytes scale of data in your data center and you want to send it to AWS infrastructure , you will be using service as a Migration.

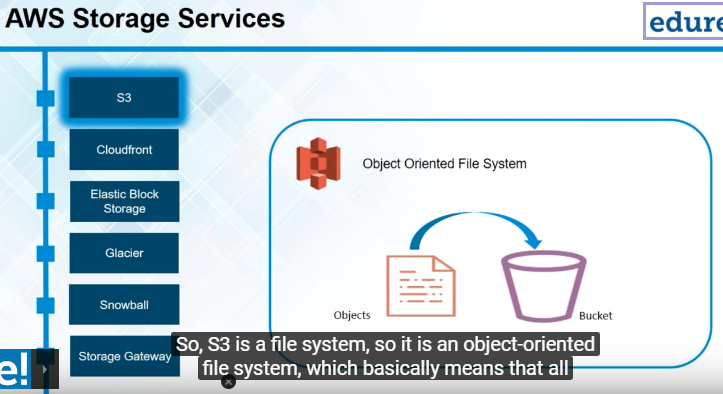
Now there is a service called Snow ball in migration , which is used to physically transfer your data to AWS infrastructure.

AWS sends you a physical device which is just like a hard drive to your premises and transfer your data on to it , then AWS sends it back to the infrastructure.

**Security & Identity Compliance :**

IAM is used to authenticate users and define user rights to them .

**Storage**:



**S3 Simple Storage Service** : It’s a file system , Its an **object based file system** in which can store your files and access them as and when required.

All your files that you upload on S3 are treated as a objects. These objects have to be stored in a bucket. We can consider the bucket is a Folder. The root folder has to be a bucket. We cannot just upload to S3. 1st we have to create bucket and inside this bucket , the secondary folders are called normal lingo, and the 1st folder which we created is called bucket.

Once create the bucket then we can upload files and these files and in the AWS lingo are objects. So we can upload objects and these objects will have certain path , which you can incorporate in our application and that is how you can access your file from your file systems.

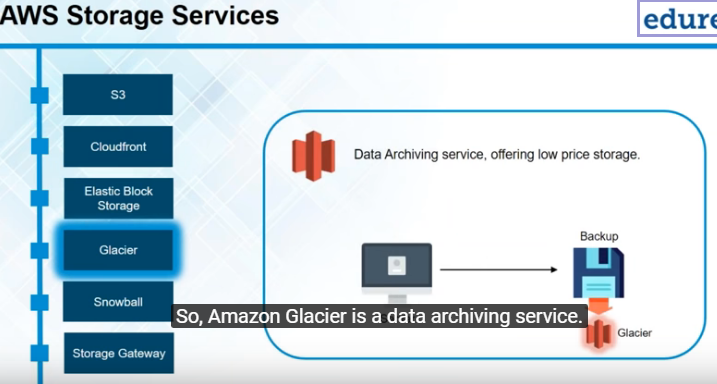
Database cannot include your executable file. So suppose have image file , so this image file would not be stored in a database , its better to store that image file in a file system and hence that image file using a path , which can be stored in the database.

**Elastic Block Storage :** is basically like a hard drive to EC2.When you are using to EC2 instances , obviously your operating system or your software is being stored somewhere. So Ec2 is backed by EBS for that matter , So EBS basically acts as a hard drive for EC2 and it cannot be used independently , it has to be used with EC2 only.

**Amazon Glacier :**

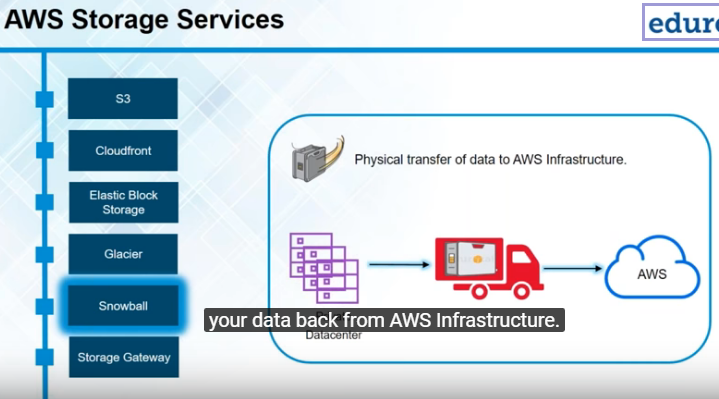
It is a data archiving service, So when you have to backup data when you are say , suppose S3 or E2 instance you back it up on Amazon Glacier. So why would you back it up on Amazon Glacier is because they use magnetic tapes and these magnetic tapes are cheap and hence your data storage on Amazon Glacier becomes cheaper.

Why need to store in Amazon Glacier? So you would store data which is not frequently accessed . For example in Hospital old patients records can store the data in Amazon Glacier.if tomorrow may be in sometime when that patient returns it can always retrieved from Amazon Glacier.

****

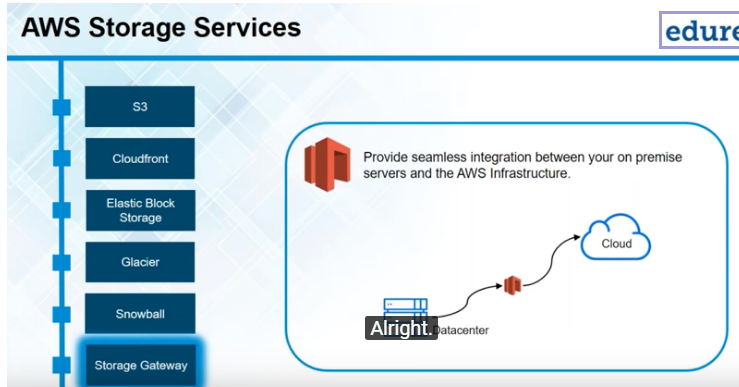
**Snowball:**

Is basically physical device which is used to transfer data from datacenters to the AWS infrastructure . It is a way of transferring your data to the AWS infrastructure or transferring your data from the AWS infrastructure



**Storage gateway :** is a service is used between your data center and your Cloud . It can be used between your data center resource as well.

How it ll be used ? Suppose we are using Application server and database server , so now your storage gateway will sit in between your database server and the application servers and it will keep on taking snapshot of your database and will keep on storing it on S3.

****

**Networking & Content Delivery :**

**Cloudfront:** is that content delivery network . It’s a caging service if user wants to connect website which is very far from the user location , that website can be caged to a location which is near the user and from the location the user can access the website. **Edge Locations**

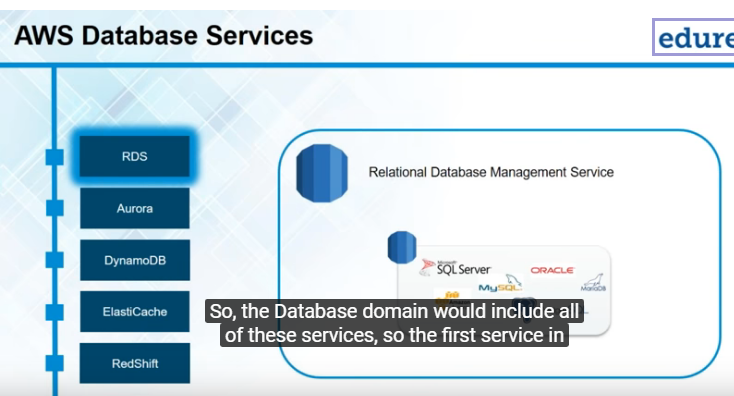
**Route53 :** redirects your traffic from the URL that you purchase, say a domain selling website like GoDaddy and redirects your instances or which are hosting your web application.

It translates in to simple to the IPAddress and redirects your traffic to the IP Address

**Messaging :**

It is used to send email to bulk to your customer base , you can send SES(Simple Email Service) and you can also handles the replies the customer gives

**Database :**

****

**RDS(Relational Database Service)** : Its managing service , which manages databases for you. Manages database like MySQL, PostgreSQL . and when say manage means they can automatically update the DB Engines ot they can automatically commit your changes.

RDS is relational databases and DynamoDB is non relational databases.

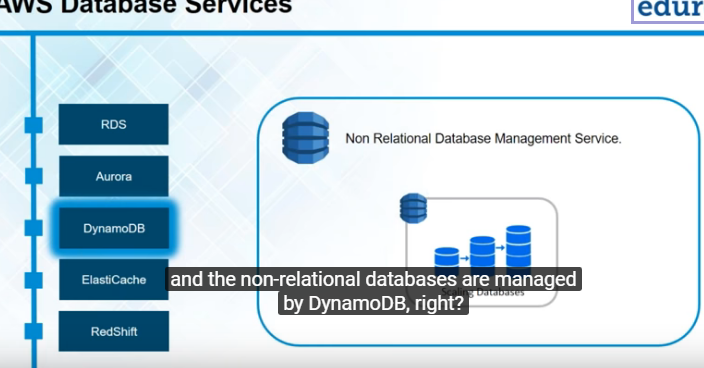
**Amazon Aurora** : is a database developed by Amazon itself. So its included in RDS ie its relational database which is also managed by RDS.

Amazon Aurora is actually based on MySQL , and its 5 times faster than MySQL, If you are using MySQL and replace the MySQL server with an Amazon Aurora server

****

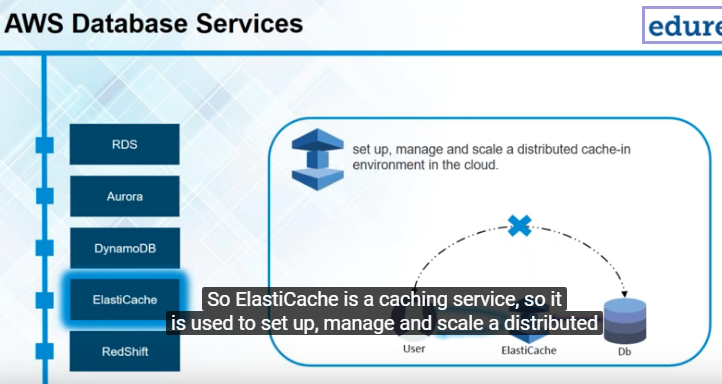
**DynamoDB:**

When you have unstructured data , you stored your unstructured data in non relational database . No need to create any storage limit , it will be increase automatically

****

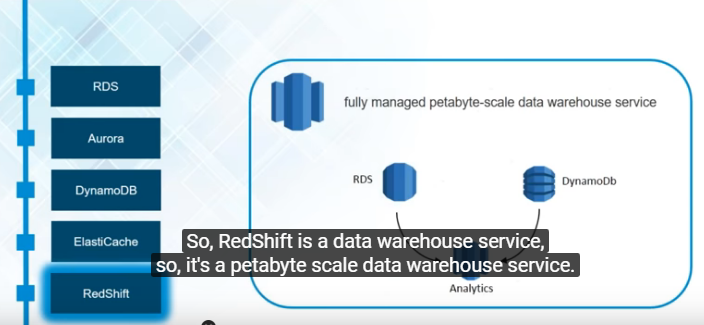
**ElasticCache:**

Database query is used frequently and resultset data is stored in cache

****

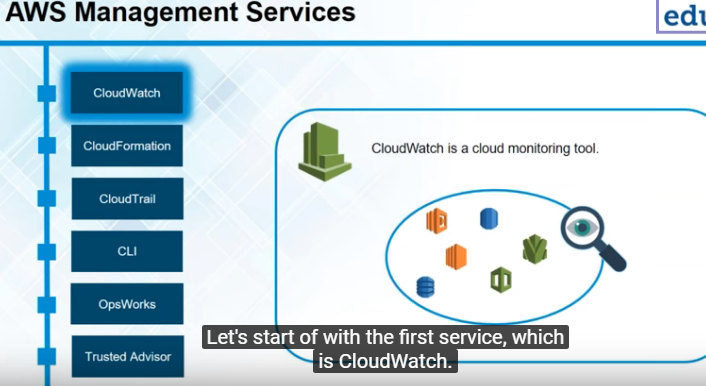
**RedShift:**

This is datawarehouse service and its used for analyse the data from RDS and DynamoDB as well

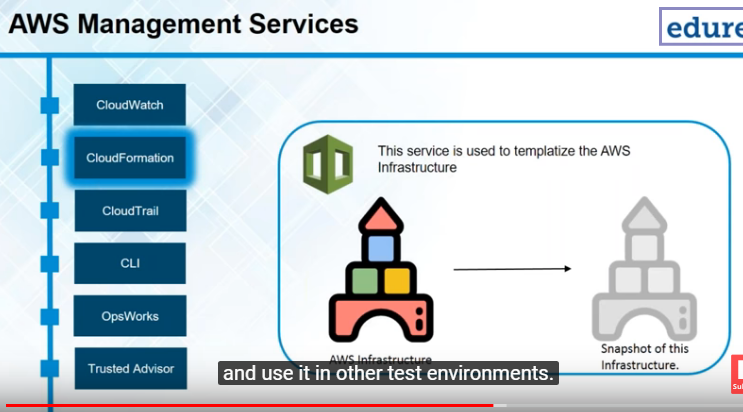
****

**Management Tools :** using for managing AWS resources , example Cloudwatch. This is monitoring tool

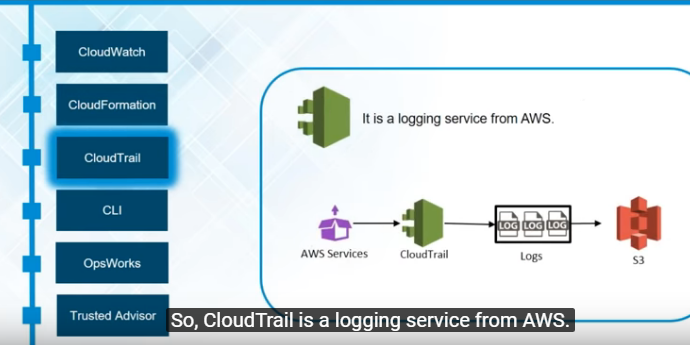
**CloudWatch :** is basically a monitoring tool , which is used to monitor all your AWS resources in your AWS Infrastructure . Create an ALARM in cloud watch and whenever your usage will cross 90% , it will figure an alarm and alarm in turn will send you a notification may be send email or whatever parameter which we set , will receive an alarm in that scene .



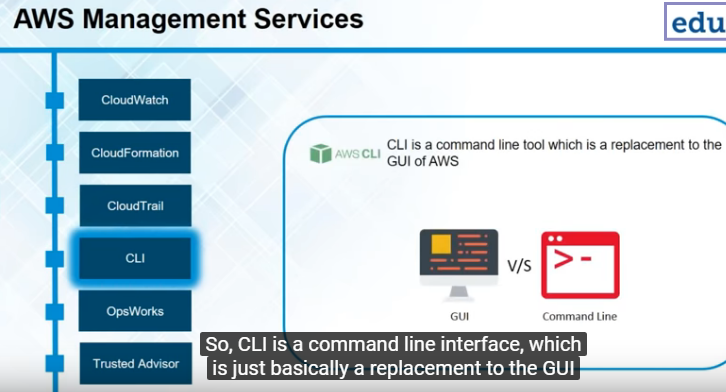
**CloudFormation** : is basically used to templatize your AWS infrastructure , now why would we templatize your AWS infrastructure as when we have different environments and you want to launch the same infrastructure in different environments . So if we create infrastructure and we don’t want to create that again , you can always take snapshot of its using the cloudformation and then you can templatize this infrastructure and use it other test environment



**CloudTrail :** is a logging service from AWS . so we can log all our API requests and API responses in CloudTrail . If we want troubleshoot a problem . CloudTrail will generate logs and will store the logs in to S3, which a file system provided by AWS.

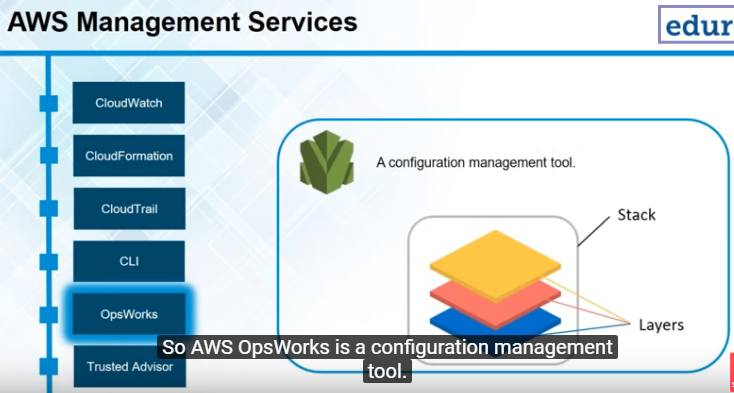
****

**CLI :** is a command line interface , which is just basically a replacement to the GUI interface that you have. AWS dashboard is the GUI . We can also use command line to deploy instances and the way to do that is by using AWS CLI

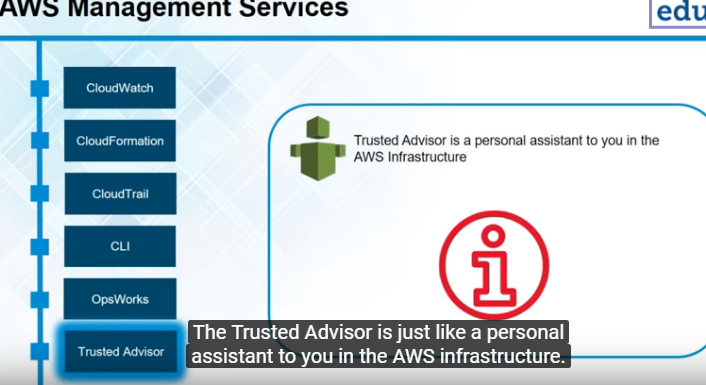
****

**OpsWorks :** is a configuration management tool . It consists two parts Stack & Layer.

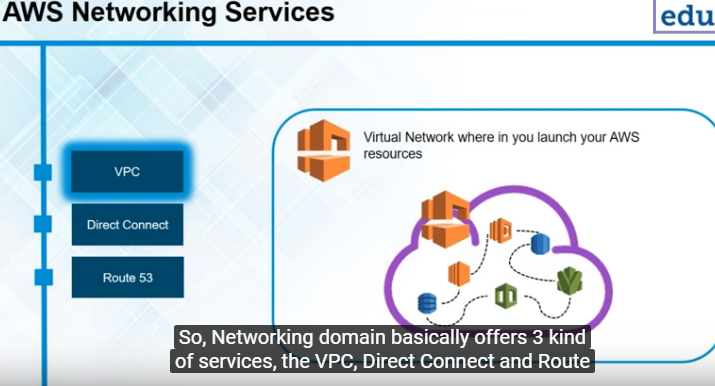
Layers are basically different AWS services that you have combined together and when you combine together the whole system is known as stack . Basically all your resources , all the services that you are using and different layers in that stack , and the combination is known as Stack

****

**Truster Advisor :** is apersonal assistant to you in the AWS infrastructure , It advices you on monthly expenditure and using our IAM policies , if it recognizes that a lot of user , a lot of different peoples are using your AWS account and you have not set up any IAM policies on your AWS account. It will advise you to create those IAM policies and hence this will enable you to create your AWS account.

****

**Networking domain:**

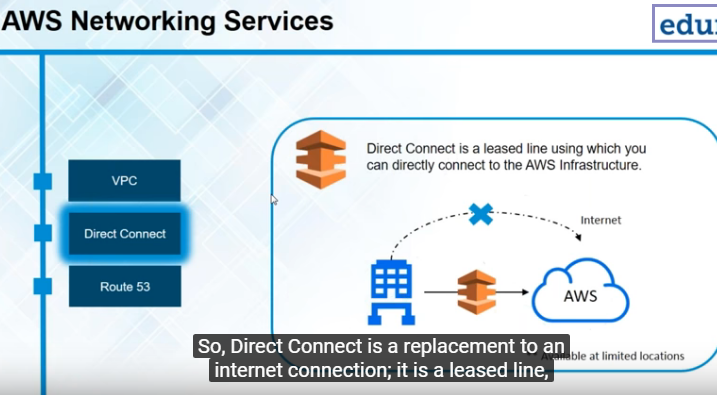
****

**Virtual Private Cloud(VPC) :** it’s a virtual network . If you include all your AWS resources that you have launched inside one VPC , then all these resources become visible to each other or can interact which each other once they are inside the VPC.

The other use for VPC is when you have a private data center , and you are using AWS infrastructure as well, and you want your AWS resources to be used as if they were on your own network , in that case you will establish ‘’Virtual Private Network” that is VPN connection to your virtual private cloud in which you have included all the services that you want on your private network .you will connect your private network to the VPC using the VPN and then you can access all your AWS resources as if they were on your own network and that is what VPC is all about

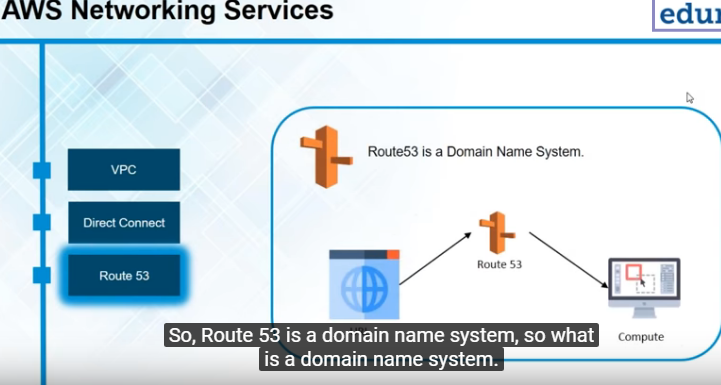
VPC provides security , it makes communication between the AWS services easy and it also connect your private data center to the AWS infrastructure.

**DirectConnect :**

****

Direct Connect is a replacement to an internet connection, it’s a leased line a directline to the AWS Infrastructure , So if you feel that the bandwith of the internet is not enough for your data requirements or networking requirements you can take a leased line to the AWS infrastructure in the form of DirectConnect Service , So instead of using internet , you would now use the Direct Connect service for your data stream to flow between your own data center to the AWS infrastructure

**Route53 :**

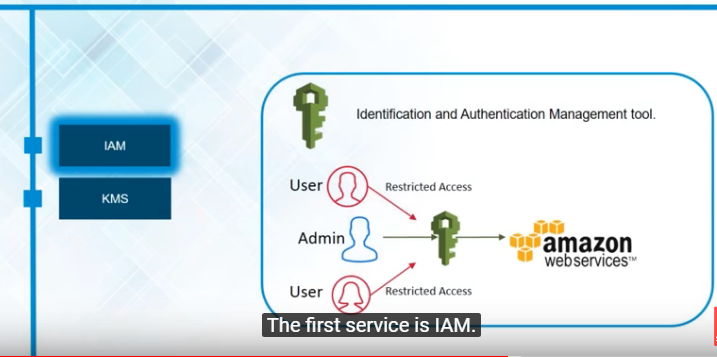
****

Route53 is a domain name system. Domain name system is basically whatever URL you enter has to be directed to a domain name system which converts the URL to IP address , the IP address is of the server in which your website is being posted . The way its function like this , you buy a domain name and the only setting that you can do in that domain name or the setting which is required in the domain name are the name server. Now these name server names are provided you to by Route53 provided are to be entered into the entered in the settings of that domain name.

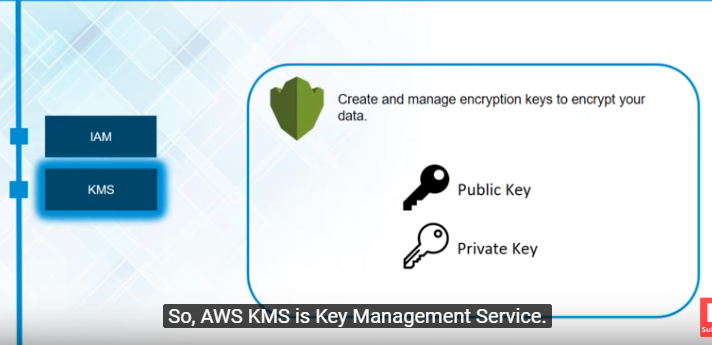
So whenever a user points to that URL , user will be pointed to the Route53. The work and the domain name setting are done. You will have to configure to Route53. Now that your request has reached Route53 , it has to be pointed to the server on which your website is hosted, So on Route53 now you have enter the IP address or the alias of the instance which you want your traffic to be directed to . So you feed in the IP address or you feed in the alias , and its done. The loop is now complete. Your URL now get pointed to Route53 and Route53 in turn will point to the instance on which your application or website is being hosted. So this is the roles which Route53 plays. It’s a domain name system , So it basically redirects your traffic from your URL to the IP address of the server or which you application or website is hosted.

**Security Service domain :**

**Identification & Authentication Management(IAM) :** create user , group and permission , Suppose you want a user to just review all instances are there, so you can give him that access.

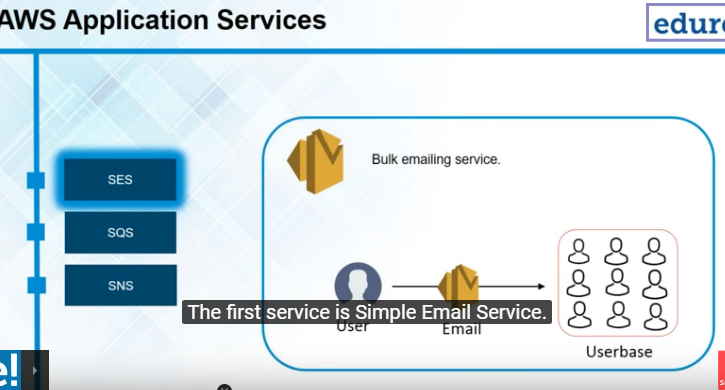
****

**Key Management Service(KMS) :** Basically any instancethat you have launched in an AWS is based on this infrastructure that there will be a public key and we will provided with a private key and the public key is with the AWS. Whenever we want to connect your instance , you have to upload the private key or you have to get the private key and then AWS will master private key with your public key and if it matches it will authenticate you to your AWS instances. KMS assigns you with the private key , we can create new key pair or we can existing one . If we lose our private keys in any case, there is no way you can gain access back to your particular AWS resource , which will be using the private key.

****

**Messaging Services :**

**Simple Email Service (SES) :** If we have large user base and you wantto send emails to them **,** you can do that on a push of a button using the SES , also if you want replies to be automated that can also be done using SES

****

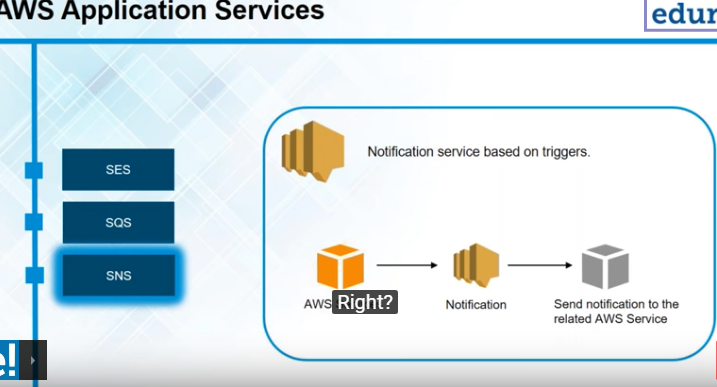
**Simple Queue Service :** acts as a buffer,For Image processing application , Whenever you upload the image , Suppose we have to do 5 tasks , Now these 5 taskswill be listed in your simple queue service or your simple queue and a server will keep reference with this queue and see what all tasks are left to be done on the image . This helps when you have multiple servers running for your processing , and suppose first 2 operations are done by the first server and the next 3 operations are may be done by some other server , So the next server should know what are the operation already done , and this knowing as actually referenced through your SQS .

So whenever a task is done that task is removed from the queue , and the next task is queued and that is what SQS basically does. Its based on priority , FIFO , The 1st task that you have listed will be executed 1st.

****

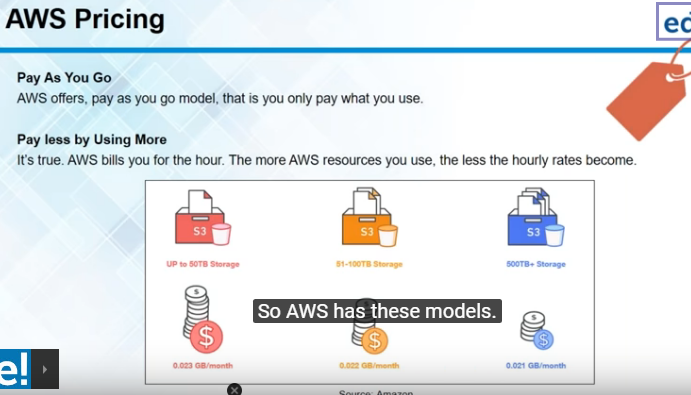
**Simple Notification Service (SNS) :** it’s basically send notification to other AWS service.

For Ex the image processing which we upload and now you also want it emailed whenever an image is uploaded , SNS will send a notification through SQS and SES that an image has been uploaded. Now when the notification is sent to SQS that notification can also include the number of tasks or the tasks that have to be done on that image. So SNS sends a notification to SQS with the details that have to be added in the queue and SNS sends a notification to SES with the details that have to be added , so sent the respective email to the respective person.

****

**AWS Pricing :**

**Pay As you Go :**  it means pay what you use

****

