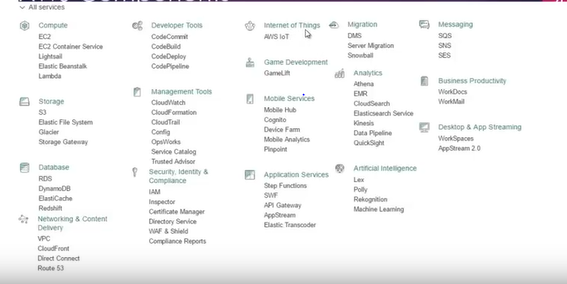


**AWS: Amazon web services:**

**Key Components in AWS:**



1. **Global Infrastructure:**

**Region**

**Availability Zone (AZ)**

**Edge Location**

1. **Content and Network delivery:**

**VPC**

**Direct content**

**Route 53**

1. **Compute:**

**EC2**

**EC2 Container service**

**Elastic Beanstalk**

**Lambda**

**LightSail**

1. **Storage:**

**S3**

**EFS (Elastic File System)**

**Glacier**

**Storage Gateway**

1. **Database:**

**RDS**

**Dynamo DB**

**ElastiCache**

**RedShift**

1. **Analytics:**

**Athena**

**EMR**

**Data Pipeline**

**Cloud Search**

**Elastic Search Service**

**Kinesis**

**Machine Learning**

**Quick Sight**

1. **Migration Services**

**DMS**

**Server Migration**

**Snowball**

1. **Security, Identity & Compliance:**

**IAM**

**Inspector**

**WAF & Shield**

**Certificate Manager**

**Compliance Reports**

**Directory Services**

1. **Management Tools:**

**Cloud Watch**

**Cloud Formation**

**Cloud trail**

**Config**

**Ops Work**

**Service Catalogue**

**Trusted Advisor**

1. **Application Services**

**Step Function**

**API Gateway**

**App Stream**

**Elastic Transcoder**

**SWF**

1. **Developer Tools**

**Code Commit**

**Code Build**

**Code Deploy**

**Code Pipeline**

1. **Messaging**

**SES**

**SQS**

**SNS**

1. **Mobile Service**

**Mobile Hub**

**Cognito**

**Device Farm**

**Mobile Analytics**

**Pinpoint**

1. **IOT:**
2. **Game Development:**

**Game Lift**

1. **Business Productivity:**

**Work Docs**

**Work Mail**

1. **Desktop & App Streaming:**

**Work Spaces**

**App Stream 2.0**

1. **Artificial Intelligence:**

**Lex**

**Poly**

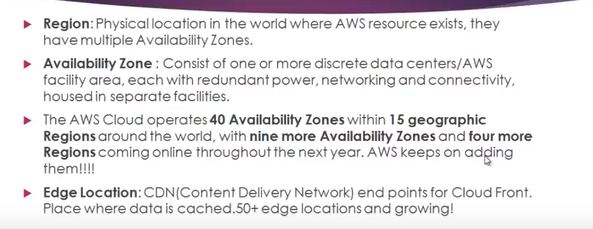
**Rekognition**

**Machine Learning**

1. **LABS:**
2. **AWS Account Creation**
3. **Setting up Billing**
4. **console**
5. **IAM**
6. **EC2**
7. **Connecting to Linux EC2 instance**
8. **Connecting to Windows EC2 instance**
9. **Launch different types of EC2 instance**
10. **EBS volumes and Snapshots**
11. **Security Group**
12. **AMI (Amazon Machine Image)**
13. **Cloud Watch**
14. **Hosting a webpage in EC2**
15. **ELB (Elastic Load Balancer)**
16. **Auto Scaling**
17. **Metadata**
18. **S3 (Simple storage service)**
19. **How to redeem your AWS free credit**
20. **How to Pay bill**
21. **AWS CLI**
22. **CDN**
23. **Route53 session1**
24. **Route53 session2**
25. **Route53 session3**
26. **Launching RDS instance**
27. **RDS features**
28. **Dynamo DB**
29. **VPC – Custom VPC creation**
30. **VPC security**
31. **Default VPC & VPC deletion**
32. **SQS**
33. **SNS**

**Global Infrastructure:**





Explanation:

Dedicated data center or data centers, it can be a single data center or it can be a grouping of more than 1 data centers with its unique networking connectivity and so on.

Region is nothing but it consists of 2 or more than 2 availability zones.

Region is the physical location on earth where the aws services are available.

CDN: Content Delivery Network.

End point of CDN in aws is edge location.

CDN service of amazon is called Cloud Front.



Explanation: we have some website data or a file in US location. Someone from India wants to use that data, then the request from Indian user has to go all the way across to US. As the request has to travel this much distance, there will be network delay (and network latency).

So good way to avoid this is to catch this data somewhere in Mumbai and so that there is no need to reach to US location. So, Indian users can easily reach the Mumbai location and band width will be very nice and there won’t be any network delay.

So instead of travelling the request from the user to all the way to US goes to the nearby location

So, the Cloud front have something like Edge location. So, these edge locations are the places where the data is cached.

**Content and Network delivery:**



**Explanation:**

VPC: Virtual Private Cloud.

VPC means Amazon has an offering which will let us use and create a section of AWS Cloud where we can launch our own AWS resources in a virtually defined network. I.e. we will have full control over the virtual network. We can do certain customizations of a network like we can define our own IP address, and can change the Sub net address and so on.

VPC is an isolated cloud source within AWS. I.e. it is an isolated datacenter.

Direct connect: It is the way to connect AWS without using internet. It is dedicated network connections to AWS.

Benefit is to Increase band width and low latency.

They have 1GBPS port and 10 GBPS port. Speed is awesome here.

Route 53: It is a DNS service from AWS. It is highly available and the scalable domain main system or DNS web service.DNS is the service which will help you to route the end users or network traffic to internet applications or web applications by translating web sites names to numeric IP address.

For ex. When user hits [www.google.com](http://www.google.com) , for this user will start internet and browse by the web site name [www.google.com](http://www.google.com) , then DNS service sits in the back end, it takes the human readable website name and it converts into corresponding numeric IP address. Then only we can see the website content. This is what a normal DNS service does.

Here the service is called as Route 53.

Why amazon gave this name? Port 53 is the actual default port number for DNS. So, they name it as Route 53.

Additional info: Route 66 is the first intersect in US. By referencing to that they named it as Route 53.

Compute:



Explanation:

EC2 is very popular concept of AWS. Its heavily depends on Virtualization concept. These are the virtual server on the cloud. We get to spin up, we get to scale up and down between new virtual servers just in few seconds.

We can launch our own server instance and we can choose our version, we get choose our memory parameters and so on.

EC2 container service: this is EC2 which is configured for Docker.

Docker is a popular service which is using container virtual concept. To manage and run Docker we have Docker containers .this is offered from EC2 container service.

Elastic Beanstalk: read above.

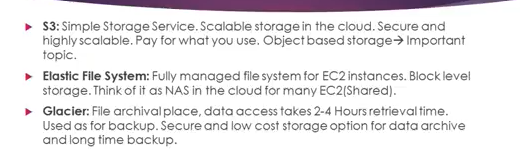
Lambda: Run the code without thinking abt servers, infra structure, underlined servers or anything.

Just Write the code and submit it to the AWS.

Lambda is a server less architecture. Only when the code is running, we will be charged only for the code execution time.

LightSail: Read above.

Storage services:





Explanation:

S3: Read above.

S3 is Object based storage that means we get to store your flat files, that means we cannot install the software’s and we cannot install anything on that. Think about it is like a drop box service facility that is running on S3 model.

We can store unlimited data on S3.

Elastic File System (EFS): It is fully managed File System for EC2 instances.

Under Compute category, EC2 is Virtual servers on the cloud.

EFS is a dedicated storage for EC2 instances. But unlike S3, it is block level storage.

NAS: Network Area Storage which is usually used in Offices, EFS is something like that.

So, this EFS is an alternative of NAS. We can connect one EFS to multiple EC2 instances. Basically this is the storage for EC2 instances.

Glacier: this is file archival.

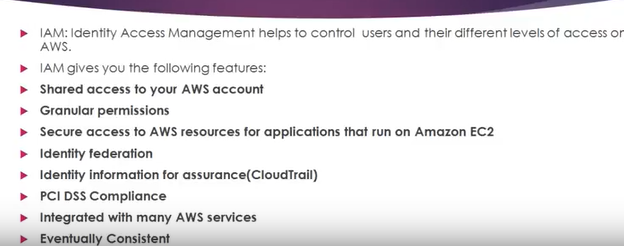
Primary reason for this is for data backup. Once we put our data to Glacier or archive. To retrieve that data back, it takes up to 2 to 4 hours. So, in the retrieval time there is a delay but it is very low cost when compared to S3 or EFS. This is one of the secure and low cost for data archival. If we are planning to take up long time backup this is the best option.

Storage Gateway:

Hybrid Storage integration means we

Lab:

1. Aws account creation
2. Setting up with billing alarm
3. IAM users sign-in link: [**https://satishluckchinnu.signin.aws.amazon.com/console**](https://satishluckchinnu.signin.aws.amazon.com/console)



Explanation: IAM: Identity Access Management.

It is a module that helps us to control users and their different levels of access on AWS.

Shared access to your AWS account: we can create a bunch of users and provide them with AWS console access. I.e. we can share the access.

Granular permission: we can provide Granular level of permission.

For ex: we want to create a user name A with full control over ec2 instance. We can do that.

If we want to create a user name B with database control over instances in AWS with not having control over EC2 instances. We can do that too.

Secure access to AWS resources for applications that run on Amazon EC2: in AWS we can secure

Identity federation:

For ex: we are working in a company having different sorts of accounts or profiles like Facebook, LinkedIn or any other active directory services. We can incorporate the same set of credentials with AWS.

So, by using this Identity federation we can federate those existing active directory or credentials from our corporate or from any 3rd party services and incorporate with AWS.

Identity information for assurance: There is a service in AWS by name Cloud Trail.

So, users who are accessing AWS resources, it gets locked into the Cloud Trail services. That means, by exploring or by going into Cloud Trail we get to see who accessed what services / what resources and what find tough time.

PCI DSS Compliance:

For example, if we want to host Credit card payment system in AWS, we can do that.

This PCI DSS Compliance will help to store and process the Credit card related items in secure manner.

Integrated with many AWS services:

We can integrate IAM feature with other AWS features such as EC2, Elastic Cloud balancer etc.

Eventually consistent:

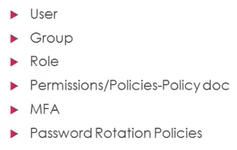
The moment we make any changes to IAM, it gets store in AWS System but it takes some time to reflect.

Reason for taking time is the changes that we make to IAM, it will stored in multiple regions so, it takes some time.

Advantage: Since it is stored in multiple regions it is field proof. There may be a slight difference in time but that is actually fine.

Free to use: This IAM service use it for free.

Creating IAM in lab:



User: Is the person that uses AWS resources or AWS Platform.

We can give provision to n number of users and provide them with certain level of access.

For ex: we can create a user A and give control over AWS Console.

And create another user B and providing just with EC2 access.

Etc.

Group: it is collections of users who has a common permission or common purpose.

For Ex: If we want to group people who has access to EC2 instance.

And people who having admin right to S3 buckets. We can group them under a single group and name it as S3 Admins.

Role:

If we want to create a new role and provide it with S3 Admin access, but it should be only assign

Into AWS resources. For ex. It can be assign to EC2.

MFA: Multi factor authentication.

It is an added level of security feature on top of conventional User name and Password.

Instead of using plain Username and Password, we can add MFA layer – it will ask for RSN tokens and google authenticator keys etc.

Amazon encourages the use of MFA.

Permissions/ policy- Policies doc:

Policy defines one or more permission.

How to define a policy: By using Policy document.

These Policy documents are featured in AWS cloud.

Policies are nothing but policies that are made of JSON.

JSON: Java Script Object Notation. This language is Key valued pair model.

I.e. define attribute and key value or key value pair model and create your custom policy documents or use the existing documents in AWS.

Password rotation policies: In this we can manage policies and we can define the policies on the Password rotation.

For ex: If we want to prompt all the users in AWS instance to change the password with in every 15days, we can define that.

