AWS
Amazon web services

CONTENTS

Chapter	Topics
S3	Understanding S3 S3 Naming Convention Transferring Data to S3 Storage classes in S3 Lifecycle policy in S3 Cost optimization for S3 Versioning in S3 Encryption in S3 Static website and CORS
Cognito	User pool Identity Pool Difference between user pool and Identity Pool

GLOBAL INFRASTRUCTURE

1. **Aws Free tier account and monitoring**

   Billing dashboard
   go to preferences and enable the receive free tier usage alert

2. **Global infrastructure**

   We can access AWS i) AWS console ii) AWS cli iii) SDKs

3. **AWS Region:**
   - *AWS Region* is a separate geographic area where we cluster data centers.
   - Each AWS Region is completely independent.

   Availability zones: Each **AWS Region** consists of multiple, isolated, and physically separate AZ's
   These are required in order to avoid fault tolerance

4. **Compute section:**

   EC2 Elastic compute cloud: Virtual machine
   ECS Elastic Container service: Container as a service.
   EBS Elastic beanstalk: PASS solution manages infrastructure for you.
   AWS Lambda: Serverless computing platform.

5. **Storage overview:**

   Databases: SQL Database - RDS, NOSQL Database - Dynamodb, Elastic cache, Redshift.
   Storage: AWS S3 object storage service file used in application storing file and retrieving files.

(2)
IAM (Identity and Access Management)

- IAM provides access to accounts services where we can manage User, Roles, Groups & Policy password policy.
- It applies globally to all AWS regions.

Users: we create users and assign necessary permissions to them in the form of policies.

Groups: We can create groups for ex. Dev QA etc. and attach policies at the group level.

Policy:

A policy is a set of permission
Always explicit deny overrides explicit allow

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": *,
   "Resource": "*"
  },
```

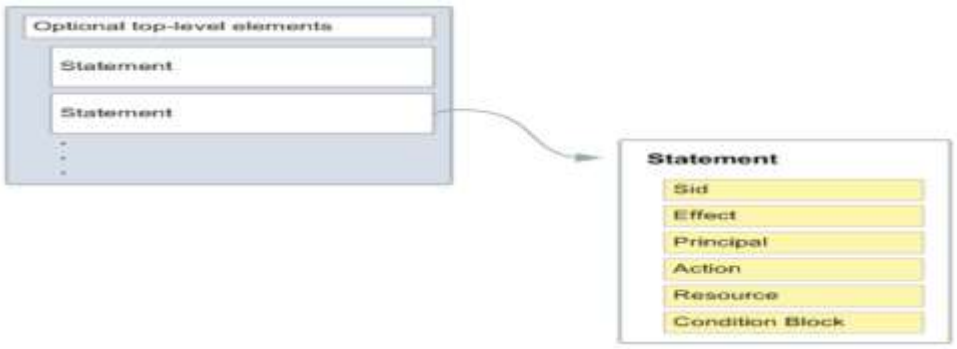
Policy types:

1. **Identity Based Policy:** Applicable on users, groups of users, and roles
   - AWS Managed policy:
   - Custom Managed Policy:
   - Inline Policy
2. **Resource Based policy:** Attach to a resource such as an Amazon S3 bucket
3. **Session based Policy:** create a temporary session for a role or federated user

Imp Notes:

More than one policy can be attached to a user or a group at the same time.
Policies can't be attached directly to resources like EC2 instance, S3 bucket etc.,

Basic Policy structure:



Effect : Can take only two value allow or deny
Principal: who is assuming the policy

Resource: on whom you are assuming the policy

Ex:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FirstStatement",
      "Effect": "Allow",
      "Action": ["iam:ChangePassword"],
      "Resource": "*"
    },
    {
      "Sid": "SecondStatement",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
     ]
}
```

Q: Can we attach multiple policies to user group or a role.

Yes, we attach.

Roles: A role is a set of permissions that grant access to actions and resources in AWS.

- Roles comes between services, like ec2 wants to access S3 or non-AWS user (hybrid account) should access AWS Resources.
- Policies can't be attached to aws resources hence roles come into picture.
- EC2 can be attached one role at a time.
- Can we assign multiple roles to a EC2 instance? No, we can't. we can assign only single role to EC2 instance.

Q: If an ec2 instance is not able to access s3 bucket what could be the reason

A Role needs to be attached with proper policy defined.

Assume Role:

Returns a set of temporary security credentials that you can use to access AWS resources that you might not normally have access to. These temporary credentials consist of an access key ID, a secret access key, and a security token.

Example:

1. create a IAM user
2. Add him ec2 full access policy
3. Try to list the bucket (aws s3api list-buckets) - you can't list the bucket because the user is not having the permission to list bucket (access denied)
4. Grant the user to assume a role
    Create a role
    Attach a policy s3 full access policy
    get inside the role ---- trust Relationship change it to user instead of ec2
    AWS: "ARN OF IAM USER"
5. aws sts assume-role --role-arn <enter the role arn> --role-session-name s3-access-example --duration-seconds 3600
6. copy the accesskey secretkey and seeison token
7 set AWS_ACCESS_KEY_ID=<enter the copied access key>
  set AWS_SECRET_ACCESS_KEY=<enter the copied session key>
  set AWS_SESSION_TOKEN=<sessiontoken>
8. aws s3api list-buckets
9. Remove the env variable
  set AWS_ACCESS_KEY_ID=
  set AWS_SECRET_ACCESS_KEY=
  set AWS_SESSION_TOKEN=

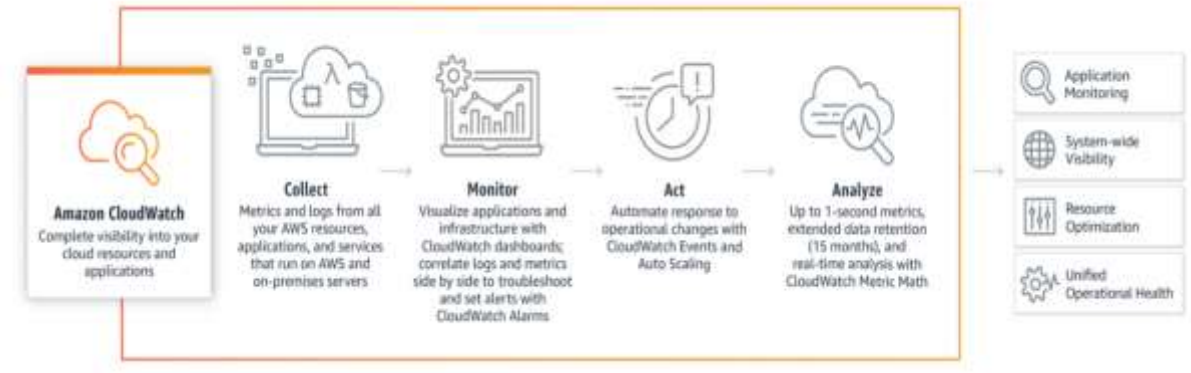
The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users).

How to delegate access across AWS accounts using IAM Roles

Primary account: Create an IAM user with sts policy attached
Secondary Account: Create a Role with cross account functionality enabled by adding the account id of the primary account
Login as IAM user and click on switch role give the details and login.

(3)
CloudWatch

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time



Default metrics of EC2 instance: Network usage CPU Usage

Metrics:
Metrics are data about the performance of your systems
Basic monitoring: which polls for every 5 minutes
Detailed monitoring: which polls for every 1 minute.

Alarm:
CloudWatch Alarms feature allows you to watch CloudWatch metrics and to receive notifications when the metrics fall outside of the levels (high or low thresholds) that you configure
Ex:
If CPU utilization goes beyond the static threshold alarm goes to alarm state
Three states in CW Alarm:
Alarm state
Insufficient
OK state

Events: An Event indicates change in AWS environment
Event Resource: Which resource you want to monitor
Event target: to alert the event change through notifications

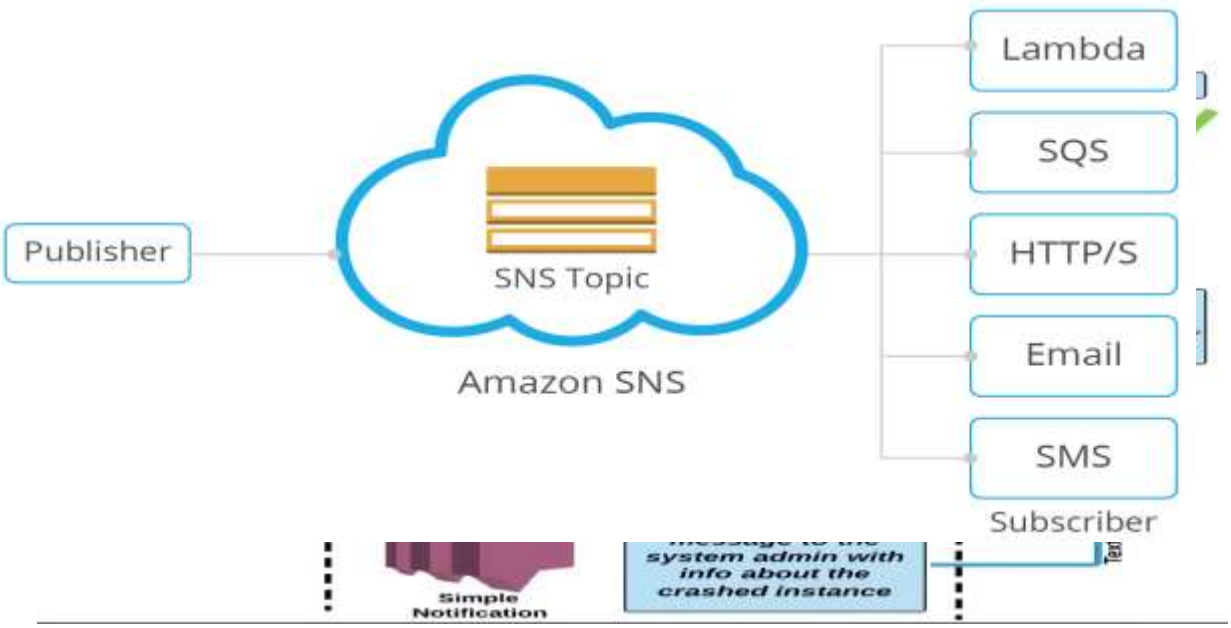
Logs:
CloudWatch Logs enables you to centralize the logs from all your systems, applications, and AWS services

follow the link and try to reproduce the same:
Ex: https://www.youtube.com/watch?v=F4IE69V-iuw

(4)
Simple Notification Service

- Amazon Simple Notification Service is a notification service provided as part of Amazon Web Service.
- It provides a low-cost infrastructure for the mass delivery of messages, predominantly to mobile users

Topic:
An Amazon SNS topic is a logical access point that acts as a communication channel

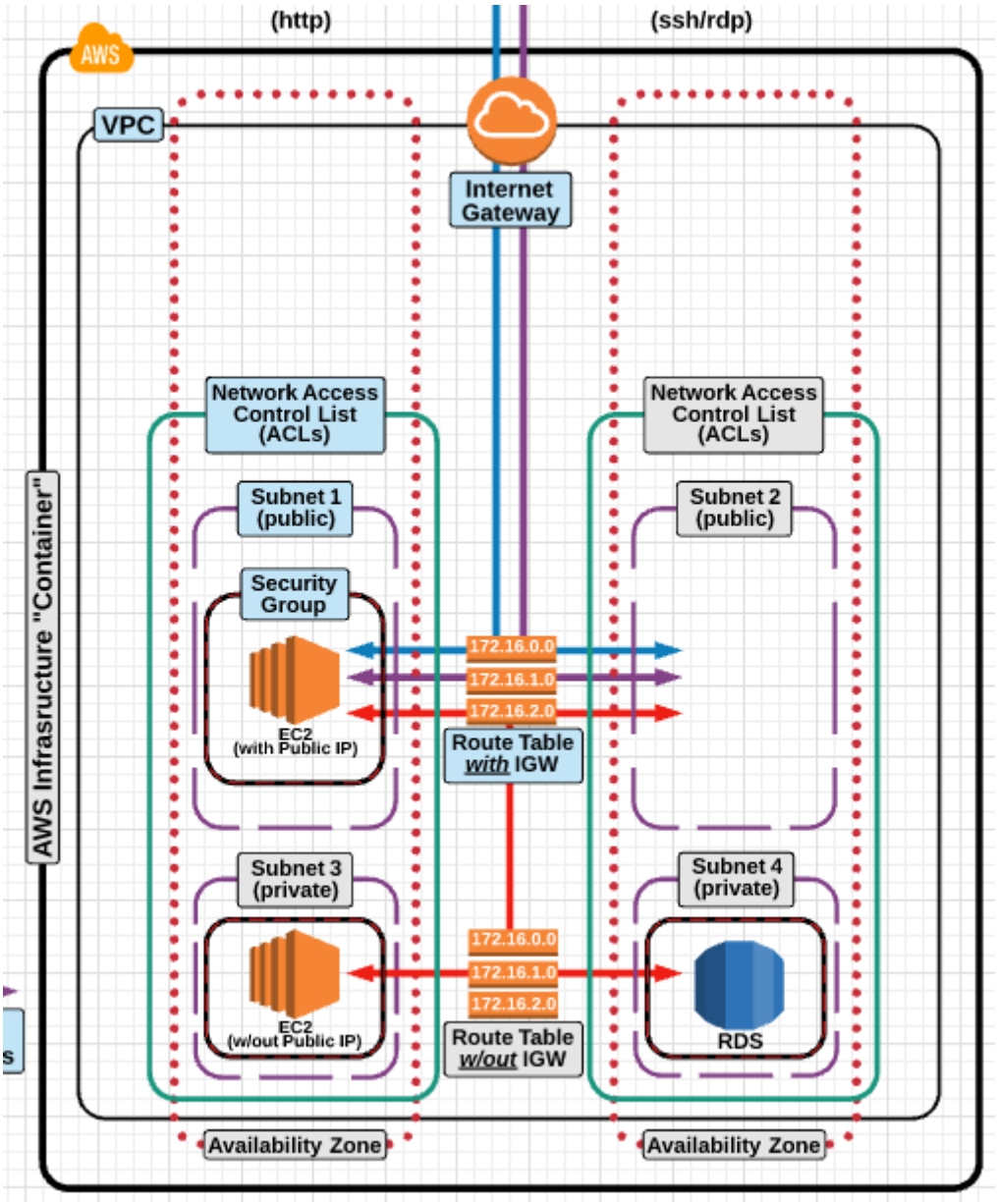


(5)
CloudTrail

- Auditing tool records all AWS account activity.
- Any action taken by users, roles and AWS services are recorded to cloud trial.
- Cloud trial events are kept for 90 days in event history
- You can create a trail of your own store the event history in s3 bucket.
- There are two types of event
  Management events: Management operations performed on AWS
  Data events : currently supported S3 and Lambda: You can now record all API actions on S3 Objects and receive detailed information such as the AWS account of the caller, IAM user role of the caller, time of the API call, IP address of the API, and other details
  Insights events: AWS CloudTrail Insights helps AWS users identify and respond to unusual activity associated with write API calls by continuously analyzing CloudTrail management events.

(6)
Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define like EC2 instance Databases.



CIDR:(Classes interdomain routing)

Classless Inter-Domain Routing is a method for allocating IP addresses and for IP routing.

Ex: The IPv4 block 192.168.100.0/22 represents the 1024 IPv4 addresses from 192.168.100.0 to 192.168.103.255.

I.e. 2^ (32-22) = 2^10 = 1024 IPv4 addresses.

VPC design:

VPC CIDR = 10.180.0.0/16 means we have 65536 IPv4 address			
IPV4 Address range is 10.180.0.0 ---- 10.180.255.255			
Public subnet 1	Public Subnet 2	Private Subnet 1	Private Subnet 2
10.180.0.0/24	10.180.1.0/24	10.180.2.0/24	10.180.3.0/24
256 IPV4 address	256 IPV4 address	256 IPV4 address	256 IPV4 address
10.180.0.0-10.180.0.255	10.180.1.0-10.180.1.255	10.180.2.0– 10.180.2.155	10.180.3.0 – 10.180.3.255

- A **subnet** is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet.
- Some IP addresses are reserved they are
    - 10.180.0.0 Network address
    - 10.180.0.1 VPC Router
    - 10.180.0.2 DNS server (**DNS**. (Domain Name System) The Internet's system for converting alphabetic names into numeric IP addresses)
    - 10.180.0.3 Future use
    - 10.180.0.255 N/W Broadcast address
- VPC spans multiple Availability zones.
- Subnets must be associated with route table
    - A public subnet has a route to internet
    - A private subnet doesn't have route to internet. It creates higher level of security.
- You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

VPC Quota or VPC limitations

- 5 VPC per region
- 5 IGW per region
- Subnet per VPC 200
- IPv4 CIDR blocks per VPC 4
- Elastic IP addresses per Region 5
- Internet gateways per Region 5
- NAT gateways per Availability Zone 5
- Network ACLs per VPC 200
- Rules per network ACL 200

VPC Peering:

- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.
- Instances in either VPC can communicate with each other as if they are within the same network.
- You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account.
- The VPCs can be in different regions (also known as an inter-region VPC peering connection).

Conditions:

- CIDR block shouldn't overlap
- Transitive peering relationships are not supported. i.e here VPC B cannot connect with VPC C.
- If the VPCs are in different regions, inter-region data transfer costs apply.
- You cannot have more than one VPC peering connection between the same two VPCs at the same time.

NACL:

1. A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. (Firewall at subnet level)
    1. Inbound means – incoming (Ingress)
    2. Outbound means – outgoing (egress)
    3. Always explicit deny take precedence over allow

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
80	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
90	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Security Group:

A **security group** acts as a virtual firewall for your instance to control inbound and outbound traffic.

Security group rules:

Inbound | Outbound

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	Admin access.
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	Web traffic.
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0	Secure web traffic.

Add Rule

Natgateway:

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC, but external services cannot initiate a connection with those instances.

Bastionhost:

A **bastion host** is a **server** whose purpose is to provide access to a private network from an external network.

VPN:

Is mainly used to establish a secure and private tunnel from you network or device to aws network

- Aws site-to-site vpn: enables you to securely connect your on-premises network to your vpc.
- AWS client vpn : enables you to securely connect users to AWS or on premises network.

Do we have another way we can connect to the resources in a private subnet?

We can setup a vpn server in the public subnet and configure it to connect to resources residing the private subnet

(6)
Elastic Cloud Compute

An **EC2 instance** is a virtual server in Amazon's Elastic Compute Cloud (**EC2**)

EC2 instance types:

EC2 Instance Type:

- Instances Types describe the "hardware" components that an EC2 instance will run on:
  - Compute power (processor/vCPU)
  - Memory (ram)
  - Storage Options/optimization (hard drive)
  - Network Performance (bandwidth)
- Instance Types are grouped into families and types that you can choose from that have different purposes:
  - General Purpose (T2, M5, and M4):
    - T2 - Burstable performance, good for many general purposes
    - M4/M5 - Small or mid-size databases, data processing, enterprise applications
  - Compute Optimized - (C4 and C5):
    - High performance web servers, science/engineering apps, ad serving
  - Memory Optimized - (X1e, X1, and R4):
    - High performance databases, in-memory databases, large data processing engines
  - Accelerated Computing - (P3, P2, G3, F1):
    - P2/P3 - Machine/Deep learning, high performance databases, server-size GPU compute workloads
    - G3 - 3D visualizations and rendering, application streaming, video encoding, server-side graphics workloads
    - F1 - Genomics research, financial analytics, big data, and security
  - Storage Optimized - (H1, I3, D2):
    - D2/H1 - MapReduce, HDFS, network file systems, or data processing applications
    - I3 - NoSQL databases (Cassandra/MongoDB/Redis), data warehouses, Elasticsearch

Note that the instance families and types are the 'current' generation as of April 2018.

EC2 purchasing options:

EC2 Purchasing Options:

On-Demand:

- On-demand purchasing lets you choose any ***instance type*** and provision/terminate it at any time
- Is the ***most expensive*** purchasing option
- Is the ***most flexible*** purchasing option
- You are only charged when the instance is ***running*** (and billed by the second)

Reserved Instances (RI):

- Reserved purchasing allows you to purchase an instance for a ***set time period*** of one or three years
- This allows for a ***significant price discount*** over using on-demand
- You can select to pay upfront, partial upfront, or none upfront
- Once you buy a reserved instance, you own it for the selected time period and are ***responsible for the entire price*** - regardless of how often you use it
- Purchases of AZ-specific RIs provide capacity reservation in that AZ. Regional RI purchases do not - so it is theoretically possible AWS will run out of capacity

Spot Instances:

- Spot pricing is a way for you to ***"bid"*** on an instance type, and only pay for and use that instance when the spot price is ***equal to or below*** your "bid" price
- This option allows Amazon to sell the use of ***unused instances***, for short amounts of time, at a ***substantial discount***
- ***Spot prices fluctuate*** based on supply and demand in the spot marketplace
- You are ***charged per second (with conditions)***
- When you have an active bid, an instance is ***provisioned for you when the spot price is equal to or less than you bid price***
- A provisioned instances ***automatically terminate when the spot price is greater than your bid price***.
- Bid on unused EC2 instances for "non production applications"

Dedicated Hosts:

- A dedicated physical machine that you have full control over. This can help save money on license fees and meet certain regulatory compliances

EBS (Elastic block storage)

EC2 Elastic Block Store (EBS) Basics:

- EBS volumes are ***persistent***, meaning that they can live beyond the life of the EC2 instance they are attached to
- EBS backed volumes are ***network attached storage***, meaning they can be attached/detached to or from various EC2 instances
- However, they can only be attached to ONE EC2 instance at a time
- EBS volumes have the benefit of being backed up into a ***snapshot*** - which can later be restored into a new EBS volume
- By default, EBS volumes are replicated within the Availability Zone
- EBS volumes are usually mounted to the file system at /dev/sda1 or /dev/xvda

---

EBS Types:

EC2 Elastic Block Store Volumes:

General Purpose SSD:

- Use for dev/test environments and smaller DB instances
- Performance of 3 IOPS/GB of storage size (burstable with baseline performance)
- Volume size of 1GB to 16TB
- Considerations when using T2 instances with SSD root volumes (burstable vs. baseline performance)

Provisioned IOPS SSD:

- Used for mission critical applications that require sustained IOPS performance
- Large database workloads
- Volume size of 4GB to 16TB
- Performs at provisioned level and can provision up to 32,000 IOPS per volume

Throughput Optimized HDD and Cold HDD:

- Cheaper than SSD options, also less performant
- Cold HDD - Designed for less-frequent access
- Volume size of 500GB - 16 TB
- Cannot be a boot volume

EBS Magnetic (Previous Generation):

- Low storage cost
- Used for workloads where performance is not important or data is infrequently accessed
- Volume size of Min 1GB Max 1 TB

EFS:

Amazon Elastic file system is a regional service storing data within and across multiple Availability Zones (AZs) for high availability and durability

Difference between EBS v/s EFS v/s S3

AMAZON EBS	AMAZON EFS	AMAZON S3
Hardly scalable	Scalable	Scalable
Block Storage	Object storage	Object Storage
Faster than S3 and EFS	Faster than S3, slower than EBS	Slower than EBS and EFS
Accessible only via the given EC2 Machine	Accessible via several EC2 machines and AWS services	Can be publicly accessible
Is meant to be EC2 drive	Good for shareable applications and workloads	Good for storing backups
File System interface	Web and file system interface	Web interface

Snapshot EBS

- You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are *incremental* backups

Snapshots are stored in S3

- Launch two ec2 instance in different az's(instance1 & instance2)
- Create EBS volume and attach it to instance1
- The volumes are attached to instance1 you can verify it by logging into instance1 and executing "lsblk" command, but it's not mounted you can verify it through by running command "df -TH"
- Mount the volume to instance1
    - Format the disk with ext4: "mkfs -t ext4 /dev/xvdf"
    - Create a directory in root: 1. "cd /" 2. "mkdir /mnt/mydisk"
    - Mount the disk: "mount /dev/xvdf /mnt/mydisk"
    - you can verify that disk is mounted by running "df -TH" command.
- Create some files
- Take a snapshot
- Unmount the disk
    - umount /mnt/mydisk
    - Detach the volume from ec2 instance.
    - delete the volume
- Create a new volume from snapshot
- Attach the volume to newly created instance2.
- Mount the volume to instance2
    - Create a directory in root: 1. cd / 2. mkdir /mnt/mydisk
    - mount /dev/xvdf /mnt/mydisk

Assignment: Difference between instance level snapshot and volume snapshot

Data life cycle Manager:

You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes

Amazon machine image (AMI):

- An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance.
- You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.

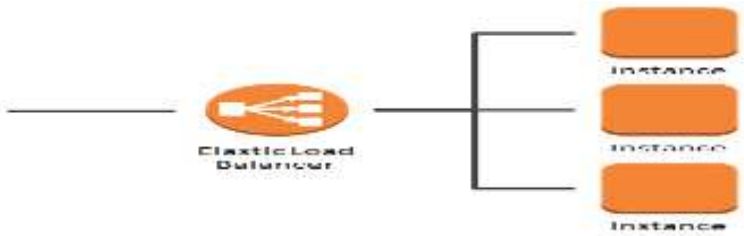
Difference between Snapshot and AMI

An EBS snapshot is a backup of a single EBS volume. The EBS snapshot contains all the data stored on the EBS volume at the time the EBS snapshot was created.

An AMI image is a backup of an entire EC2 instance. Associated with an AMI image is EBS snapshots. Those EBS snapshots are the backups of the individual EBS volumes attached to the EC2 instance at the time the AMI image was created.

Elastic load Balancer (ELB):

Manage and control the flow of inbound request to group of targets by distributing the requests evenly across the targets. The targets may be EC2 instances lambda or containers.



Types of Load balancer:

Application load balancer:

- Used mainly for web application running http and https protocols.
- Operates at request level.

Network Load balancer:

- Ultra-high Performance at very low latency.
- Operates at connection level, routing traffic to targets with in VPC.
- Can handle millions of requests per second.

Classic load Balancer:

- Used for applications that were built in existing EC2 classic env.
- Operates both at connection & request level.

Example: Classic load balancer

- Spin up an EC2 instance1 in another availability zone (az1) with http port open in Security group
- Add the below script and launch the instance.

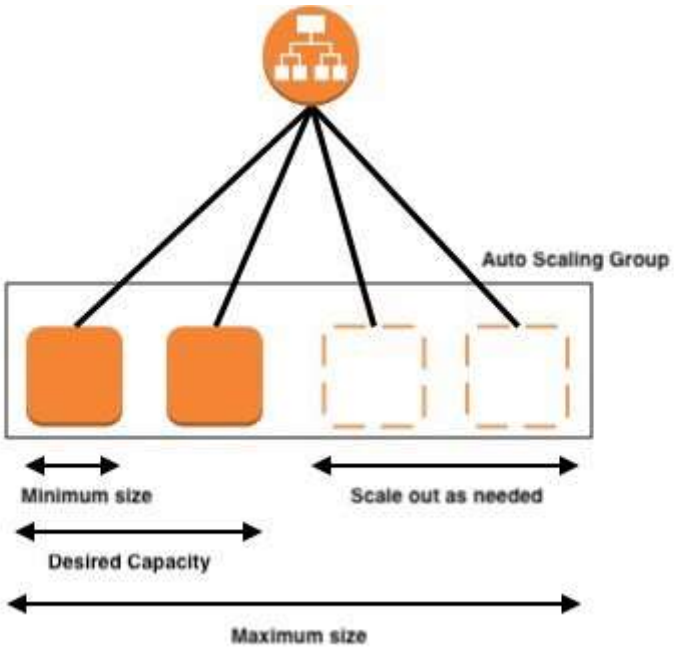
```
#!/bin/bash
yum update -y
yum install httpd -y
echo '<h1> Response from server-1 </h1>' > /var/www/html/index.html
service httpd start
chkconfig httpd on
```

- Spin up one more EC2 instance1 in another availability zone (az2) with http port open in Security group
- Add the below script and launch the instance.

```
#!/bin/bash
yum update -y
yum install httpd -y
echo '<h1> Response from server-2 </h1>' > /var/www/html/index.html
service httpd start
chkconfig httpd on
```

Autoscaling

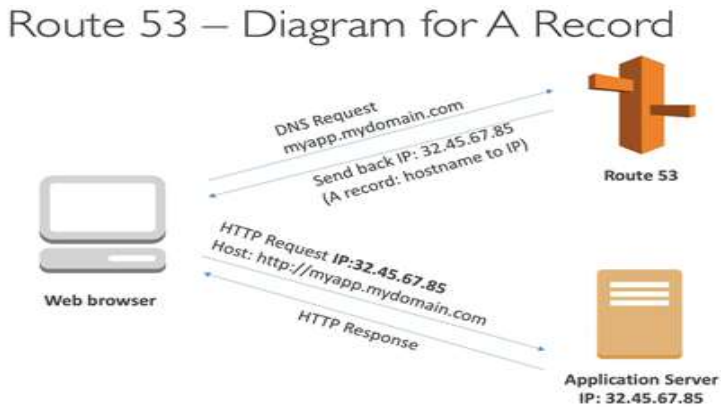
AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.



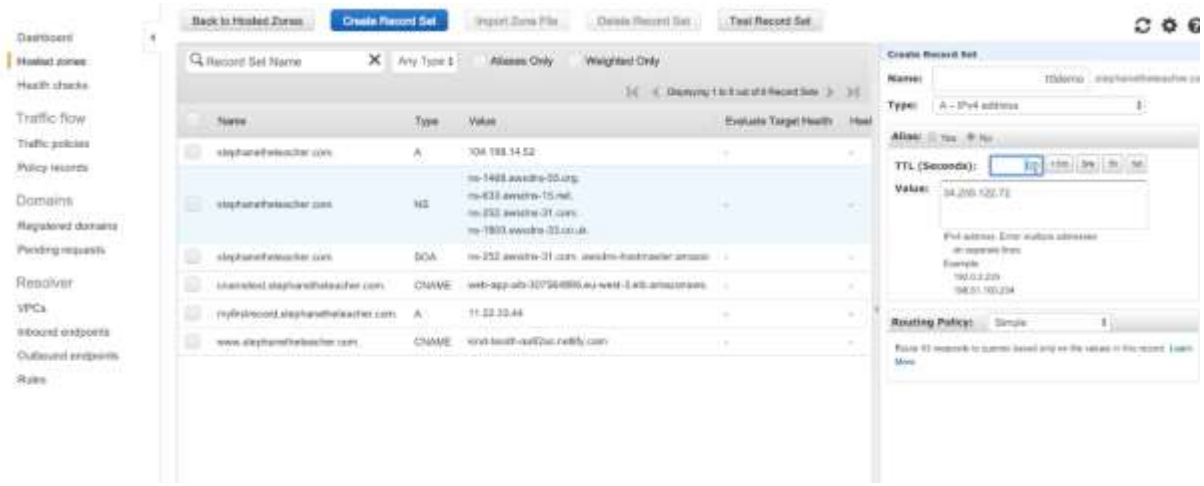
(7)

Route53:

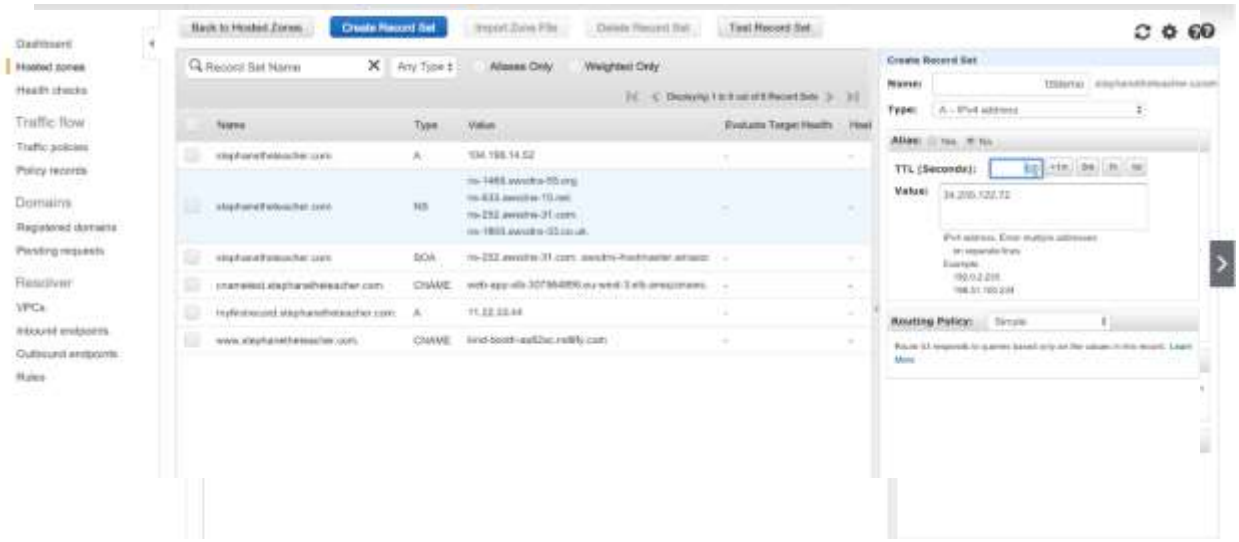
Amazon Route 53 is a highly available and scalable Domain Name Server (DNS) web service, where we can point IP address to domain name or point host name to another host name.



A Record: Maps IP address to domain name ex: 10.180.0.0 to myapp.mydomain.com

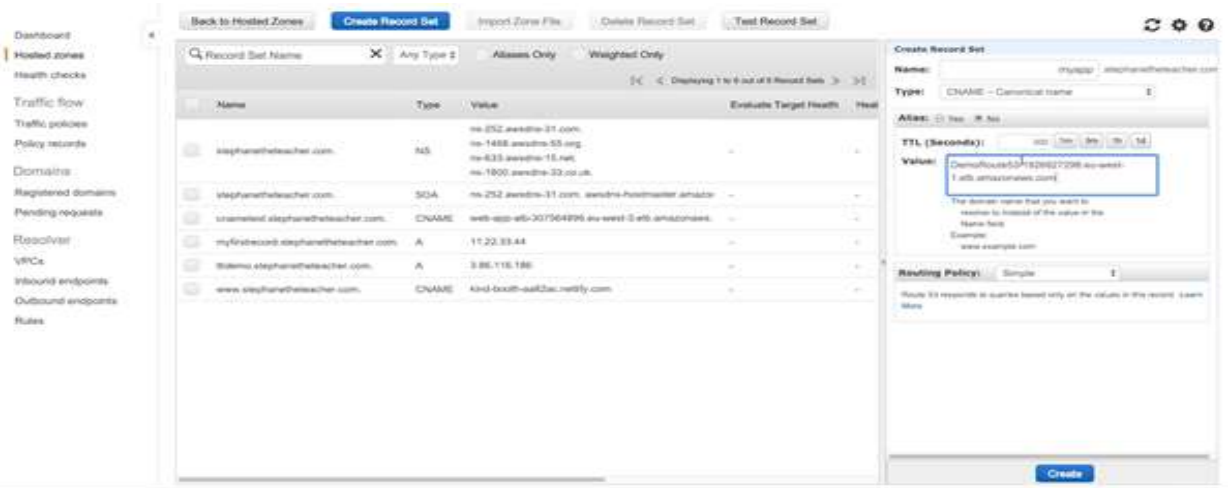


TTL



CNAME Record:

Maps hostname to another host name: us-east.2.elb.amazonaws.com to myapp.mydomain.com



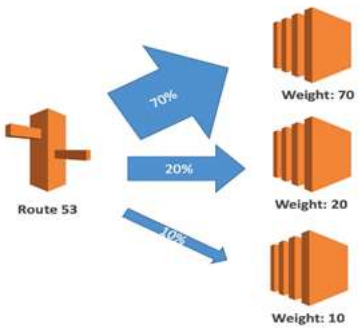
Alias Record: points a host name to AWS Resource ex: myapp.mydomain.com to us-east.2.elb.amazonaws.com

Latency Routing Policy:

Use when you have resources in multiple AWS Regions, and you want to route traffic to the region that provides the best latency.

<u>Weighted Routing Policy:</u>

Use to route traffic to multiple resources in proportions that you specify.



Simple storage service (S3)

Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.

Single operation upload:

- It's a traditional upload where you will upload the object in one part
- A single operation upload can upload the file up to 5GB in size.

Upload object in parts:

- Using multipart upload, you can upload the large objects up to 5TB.
- You can use multipart upload for the objects from 5MB to 5TB in size.

Rules for bucket naming:

- Bucket names must be between 3 and 63 characters long.
- Bucket names can consist only of lowercase letters, numbers, dots . and hyphens -.
- Bucket names must begin and end with a letter or number.
- Bucket names must not be formatted as an IP address (for example, 192.168.5.4).
- Bucket names can't begin with xn-- (for buckets created after February 2020).

Limitation of S3 bucket:

- Only 100 buckets can be created per account.
- Can hold unlimited objects

S3 Storage classes:

- Standard:
  - Designed for general- and all-purpose storage
  - Default storage option
  - 99.999999999% object durability
  - 99.99% object availability
  - Most expensive storage class.
- Reduced Redundancy storage
  - Designed for non-critical objects
  - 99.99% object durability
  - 99.99% object availability
  - Less expensive than standard
- Infrequent access
  - Designed for less frequently accessed objects.
  - 99.999999999% object durability
  - 99.99% object availability

  Less expensive than reduced redundancy storage

- Glacier

- Designed for long term archival storage
- May take several hours to retrieve the objects from this storage
- Cheapest s3 storage class

S3 Life cycle policy:

An object lifecycle policy is a set of rules that automate the migration of the object storage class to different storage class

By default, lifecycle policies are disabled for a bucket

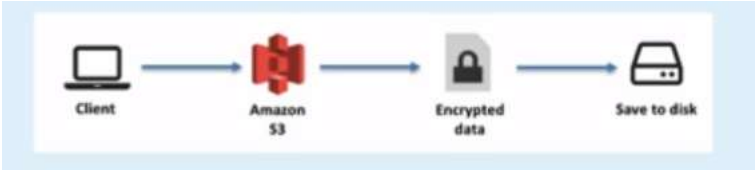
Example:

- I have a work file that I am going to access every day for the next 30 days
- After 30 days, I may only need to access that file once a week for the 60 next days
- After which (90 days total) I will probably never access the file again but want to keep it just in case

S3 Encryption:

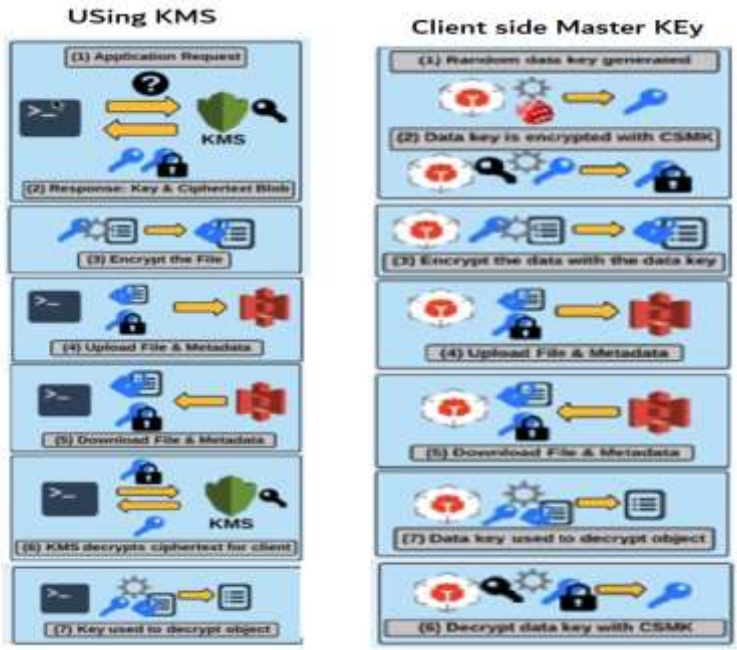
Two ways of protecting information with S3

1. **Server side/$_{\text{At rest}}$:**



2. **In-transit/Client-side encryption:**

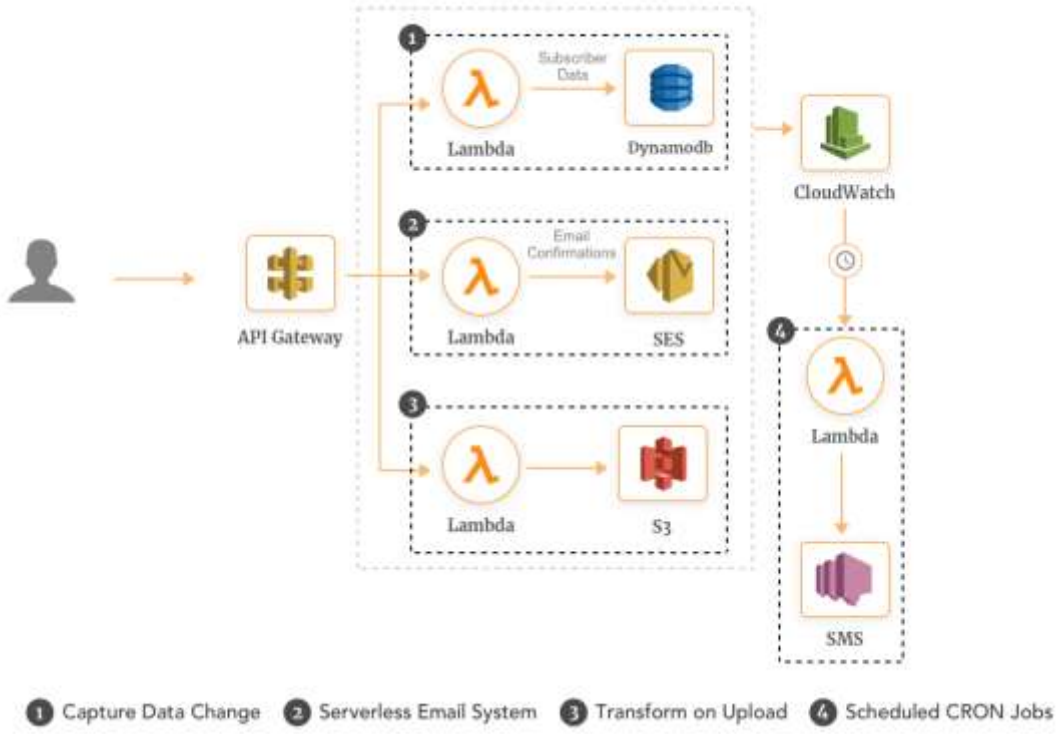
In client side/ in-transit we have two types



Lambda

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

Just upload your code and Lambda takes care of everything required to run and scale your code with high availability



- Manage your virtual functions not really caring about the servers
- Run on demand
- Scaling is automated

Billing:
Pay per request first one million requests is free $0.20 per one million request.
compute time 0.00001667 for every GB-seconds used.

AWS Lambda Languages:
NodeJS, Python, Python3, Gr00vy, java, csharp, Scala and GO

AWS Lambda Integration

Kinesis, API Gateway, DynamoDB, AWS S3, CloudWatch Events, CloudWatch logs, SNS and Incognito

Key Management Service (KMS)

AWS Key Management Service (**AWS KMS**) is a managed service that makes it easy for you to create and control customer master keys (CMKs), the encryption keys used to encrypt your data.

(11)
AWS secrets

AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

(12)
Relational Database Service (RDS)

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Which relational database engines does Amazon RDS support?
Amazon RDS database engines:

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle
- Microsoft SQL Server

Encryption in RDS:
Encryption at rest is supported for

- Amazon Aurora
- PostgreSQL
- MySQL
- MariaDB
- Oracle
- Microsoft SQL Server

Q: Can we enable encryption on exciting DB

Encrypting existing DBs is not supported. To do this, you'll need to create a new encrypted instance and migrate data to it. The encryption key can be stored in KMS.

Q: **Which is the non-relational database supported in AWS**

Amazon DynamoDB is the NoSQL database supported by AWS

The Classic Load Balancer is a connection-based balancer where requests are forwarded by the load balancer without “looking into” any of these requests. They just get forwarded to the backend section

The Application Load Balancer operates at the request level only. If you're dealing with HTTP requests, which you are for your web application, we can use this It also supports advanced features like host and path-based routing

We can create target groups in order to route to traffic to the respective paths

Cognito

- It Mainly provides authentication authorization and user management for your application
- It provides a managed user pool to mange identity for the application

Cognito provides user flows:

- Signup
- Signin
- Forgot or change password
- Multifactor authentication
- Email and phone verification

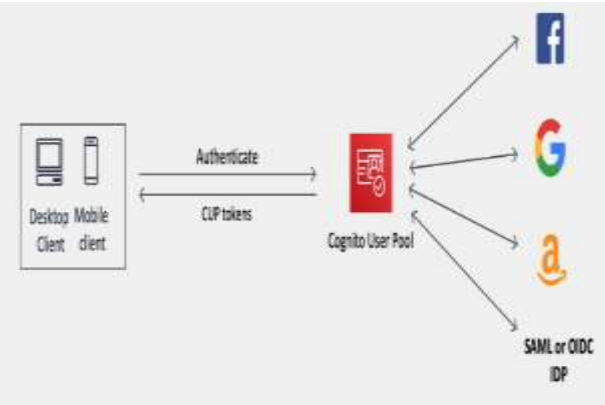
It also provides software development kit to your mobile or web application, and also provide lambda triggers in order to customize any of these user flows with you own business logic

It also provides a built-in hosted UI for these user flows

Social identity can be integrated
Facebook
Google
Amazon
SAML

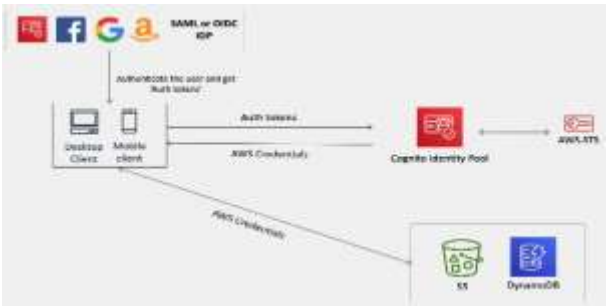
After authentication the user the Cognito provides the best practice way of accessing the AWS resources securely from the app by providing temporary credentials

User Pool:



- User pools acts as mediator between your app and external
  social identity providers
- you can add multiple identity providers as you need.
- The user pool manages the token exchange with each of the
  providers and gives your app standard user pool tokens of
  same format

Identity poll:



- Where you exchange the authentication token to get temporary aws credentials which you can use to access the resources directly from the app
- These can be used independently of each other or used together

Difference between user pool and identity pool

AWS Cognito User Pools is there to authenticate users for your applications

Say you were creating a new web or mobile app and you were thinking about how to handle user registration, authentication, and account recovery, you don't need to implement user authentication inside your application, rather you can integrate AWS Cognito User Pools, which will manage user sign-up, sign-in, password policies.

AWS Cognito Identity pool:

- This is a service which was designed to authorize your users to use the various AWS services. The source of these users could be a Cognito User Pool or even Facebook or Google.

In other words, Identity Pools are used to assign IAM roles to users (who had been authenticated through a separate Identity Provider which could be Cognito User Pools or Social logins (e.g; Gmail, Facebook & etc.)). Because these users are assigned an IAM role, they each have their own set of IAM permissions, allowing them to access AWS resources directly.

So, the difference is

- AWS Cognito User Pools: Granting access to a application
- AWS Cognito Identity Pools: Granting access to amazon service

Difference between IAM and Cognito

AWS IAM gives securely and control access to AWS services and resources for your users
AWS Cognito It Mainly provides authentication authorization and user management for your application

Link for Cognito user pool creation: https://www.youtube.com/watch?v=jTu--LpjA18

Basis Of	IAAS	PAAS	SAAS
Stands for	Infrastructure as a services.	Platform as a services.	Software as a services.
Uses	IAAS is used by network architects.	PAAS is used by developer.	SAAS is used by end user.
Access	IAAS give access to the resources like virtual machines and virtual storage.	PAAS give access to run time environment to deployment and development tools for application.	SAAS give access to the end user.

amitrooge@gmail.com
amit@2586