Install java :- yum install java-1.8.0-openjdk

Install javac:- yum install java-devel

**Intro to Cloud computing**

Use of remote computer to perform certain task.

If we have an application ready and we want it to be available to all in the world, then we can buy a server and put our application on the server, need to connect that server on internet so that everyone can use it.

->To do this, you have to buy servers, which will cost you huge money.

-> configure these servers manually, if these servers has to connect to internet, then we need to get a static IP address for the server.

-> Need to buy a domain, so that domain will get converted to IP address by DNS server.

If you have an application which requires lots of processing, then cloud computing is the best as we don’t need to buy a heavy server.

1. **AWS Global Infrastructure**

19 Regions 57 AZ🡪2018

5 more Regions in Regions and 19 more AZ

AZ is a Data Centre with switches, firewalls, load balancers, storage and all.

Region🡪 A distinct geographical area. With 2 or more AZ. Like Tokiyo, N-Virginia,Singapore,Mumbai, N-America.

And so on.

Edge Locations are Endpoints for AWS which are used for caching content. This consist of cloudfront which is Amazons content delivery network.

There are more edge locations than regions. There are over 150 edge locations.

Edge locations are small data centers that hosts the web (static & dynamic) contents.

1. **IAM(Identity Access Management)**

It allows you to manage users and their level of access to the AWS Console.

It offers following features

1. Centralized Control of AWS account
2. Shared access to AWS account
3. Granular permission
4. Identity federation
5. Multifactor authentication
6. Provide temp. Access for users.
7. Allows to set up ur own password rotation policy
8. Integrated with many AWS services
9. Supports PCI DSS compliance

Key terms for IAM

1. Users🡪End users such as pple, emp of organization
2. Groups🡪 A collection of users, each user in the group will inherit the permission of the grp
3. Policies🡪 made up of documents, called policy documents. They are in JSON Format. They give permission as to what a user/Group/Role is able to do.
4. Roles🡪 You create roles and then assign them to AWS Resources.

**IAM Lab**

1. Select a Region
2. Click on services
3. Security Identity and compliance
4. You can customize the link by Giving your name.
5. AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS Management Console, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources.
6. **You can enable MFA for your AWS account** and for individual IAM users you have created under your account. MFA can be also be used to control access to AWS service APIs.
7. You can download multiple authenticator apps like Google Aunthenticator

|  |  |
| --- | --- |
| Android | [Authy](https://play.google.com/store/apps/details?id=com.authy.authy), [Duo Mobile](https://play.google.com/store/apps/details?id=com.duosecurity.duomobile), [LastPass Authenticator](https://play.google.com/store/apps/details?id=com.lastpass.authenticator), [Microsoft Authenticator](https://play.google.com/store/apps/details?id=com.azure.authenticator), [Google Authenticator](https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2) |
| iPhone | [Authy](https://apps.apple.com/us/app/authy/id494168017), [Duo Mobile](https://apps.apple.com/us/app/duo-mobile/id422663827), [LastPass Authenticator](https://apps.apple.com/us/app/lastpass-authenticator/id1079110004), [Microsoft Authenticator](https://apps.apple.com/us/app/microsoft-authenticator/id983156458), [Google Authenticator](https://apps.apple.com/us/app/google-authenticator/id388497605) |

1. Second option, Activate MFA on your root account
2. Click on Manage MFA
3. Manage MFA account dialogue box will appear
4. Continue with Virtual MFA device
5. Scan the QR Code from Google Authenticator
6. Add the 2 consecutive numbers appearing on the screen
7. Click in Assign MFA
8. Close the dialog box
9. Go back to DashBoard(click on dashboard, top left)
10. You can see two green ticks
11. Always remember IAM is on Global Region, you cannot select a specific region
12. **Create an individual IAM user**
13. Manage users🡪Add Users
14. Give username (rincy.babu)
15. Select AWS access Type
16. Specify the access type (you can specify both programmatic and AWS management control access🡪 Auto generate password🡪 user has to create a password in nxt login
17. Click on **Permission Button**
18. We can create a group🡪click on Create Group🡪Give name as Developers\_Group🡪 Create a Policy(tick on the policy)you want to give like Administrative access or so
19. Click on **Create group**
20. An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users. For example, you could have a group called Admins and give that group the types of permissions that administrators typically need. Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and needs administrator privileges, you can assign the appropriate permissions by adding the user to that group. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old groups and add him or her to the appropriate new groups.
21. Now we can add our user to the created group
22. Tick on group🡪 click on next🡪you can add a Tag for identification
23. Click on create user
24. It will start showing you

**User Access key ID Secret access key Password Email login instructions**

rincy.babu AKIAQR6Z6U3JE6PWAVRB \*\*\*\*\*\*\*\*\* Show \*\*\*\*\*\*\*\*\* Show [Send email](mailto:?subject=Welcome%20to%20Amazon%20Web%20Services&body=Hello,%0A%0AYou%20now%20have%20access%20to%20the%20AWS%20Management%20Console%20for%20the%20account%20ending%20in%205522.%20------%0A%0ASign-in%20URL:%20https://rincycheriyan.signin.aws.amazon.com/console%0AUser%20name:%20rincy.babu%0A%0AYour%20password%20will%20be%20provided%20separately%20by%20your%20AWS%20account%20administrator.%20During%20your%20first%20sign-in,%20you%20must%20change%20your%20password.%0A%0A------%0A%0AStay%20connected%20with%20AWS%20by%20creating%20a%20profile:%20https://pages.awscloud.com/IAM-communication-preferences.html%0A%0ASincerely,%0AYour%20AWS%20Account%20Administrator)

1. You can also download the CSV file and provide it to your user
2. If you close it, and you can see the user with the group
3. If you go back to user dash board, you will see two more Green Ticks added
4. Apply an IAM password policy
5. Click on it🡪 click on button Manage Password🡪click on set password policy🡪you can click on checkboxes
6. If you go back to dashboard, you will see all buttons enabled.
7. **Assigning Roles**
8. Click on Roles on the Left hand side
9. **What is a Role**
10. An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.
11. Click on Create Role
12. Select type of trusted entity as AWS Service
13. Choose a use case as EC2
14. Click on Next Permission
15. **Filter policy🡪S3**
16. AmazonS3full access
17. Click on Next Tags; Give a Key value pair
18. Click on Review
19. Give a proper role name like S3\_Admin\_access
20. **Click on Create Role**

**Summary IAM**

* **IAM is universal, it does not apply to any region**
* **The root account is simply the account created when we first setup our AWS account. It has complete admin access**
* **New users have no permissions when first created**
* **New users are assigned AccessKeyId and Secret access Key when first created.**
* **These are not same as passwords. You cannot use access key id and secret key id to login to the console. You can use this to access AWS via API’s and Command Line.**
* **You get to view them only once, if you lose them you have to regenerate them, so save that CSV file safely**
* **Always set MFA in your root account**
* **You can create and customize your own password rotation policy.**

**SERVICE-4 BILLING ALARM**

Most of the time, we will be using free tier services. As we would also we learning about services that need license and cost, we need to create a Billing Alarm so as to ensure we don’t cross certain limit over a period of month.

1. Select the Region as N-Virginia
2. Select the service Cloud Watch (it comes under Management and Governance)
3. Click on Billing at the right side corner
4. Click on Create Alarm
5. You must see Metric and Conditions Screen
6. Select the conditions
7. Threshold type->static
8. Whenever estimated charge is Greater Than Threshhold
9. 10 USD
10. Click on Next
11. Notification Page🡪 select create a topic
12. Create TOPIC
13. Enter your mail id for notification
14. Click on next
15. Give name to the alarm like Billing Alam
16. Give alarm Description
17. You will move to Create and Preview Page
18. Click on create Alarm button
19. You will see alarm being created with all details, which you wanted.
20. Initially it will show Insufficient, as 1hr is not over.
21. Over your mail, you will receive notification
22. On the dashboard you will be able to see Alarm Details
23. **S3 (Simple Storage Service)**

It’s a safe place to store your objects. Like files.

* Files can be from 0 Bytes to 5TB
* There is unlimited storage
* Files are stored in Buckets (it’s a folder)
* S3 is a universal name space, it means name must be unique as it creates a WebAddress.
* When you upload a file successfully to S3 bucket, you will receive a status code of 200
* Objects consist of following

Key (name of the Object)

Value (This is simply the data, made up of sequence of bytes)

Version Id (important for Versioning)

Metadata (Data about data like Tagging)

**Data Consistency for Files**

1. If you put a data, data will be available immediately
2. If you update the same file, there can be a possibility that data will be available after some time (It may give you version1 or Version2) but if you wait for some time, it will give u version2.

**S3 has following guarantees**

1. Built for 99.999% availability and durability

It is unlikely that it will be lost

S3 has following features

1. It has Tiered storage
2. Life cycle management ( we can move a file from one tier to other)
3. We can version a file
4. We can encrypt a file
5. MultiFactor Authentication deletion
6. You can also secure your data using Access Control List and bucket policies.
7. Its not suitable to install an OS

**S3 classes**

1. S3 IA(infrequent Access)- for the data that is accessed less frequently, but requires rapid access when needed. Lower fess as compared to S3 but has retrieval charges.
2. S3 One Zone –IA🡪 for where you want a lower cost option for infrequently accessed data, but do not need multiple AZ
3. S# Intelligent Tiering (introduced in 2018)🡪 it is designed to optimize cost by automatically moving the data to the most cost effective access tier without effecting the performance or operational overheads. This is using ML, it checks how frequently u are accessing your data. Accordingly it moves your data
4. S3 Glacier comes in 2 flavors🡪 S3 glacier and S3 glacier deep Archive.

**S3 charges you in following ways**

1. Based on storage
2. Requests
3. On the storage Management pricing based on tiering
4. Data Transfer pricing
5. Transfer acceleration
6. Cross region replication pricing

**What is cross region replication?**

If we have a bucket in N-America, and we want it to be replicated in Sydney region, so as to have a proper backup in case of any disaster, we can do that for security. Asa you upload an object in your bucket, it will get replicated in other region as well.

**S3 Transfer Acceleration**- it enables fast easy and secure transfer of files over long distance between end user and S3 Bucket. It takes advantage of Edge Location, asa data arrives at edge location, data is routed to S3 over an optimized network path.

**Lab S3**

Remember S3 is stored in global region.

Services>>Storage>>S3>>Create Bucket>>Enter Bucket name which has to be unique globally>>Select a Region>>Clcik on Next>> No versioning as of now>>Click on next >>by default all public access is denied, we will keep it blocked>> click on Create bucket after viewing preview.

Click on the bucket to see its Objects,>> Select two images>>Upload>> you will see Progress of upload>>if you click on the object, will be able to see its details.

You cannot make it public, until the bucket itself is public.

So move to bucket, click its checkbox.>> Edit public access>> uncheck Block public access>>Save>>confirm

Now go inside your bucket>> select the file you want to make public>> click on Actions>> Make public>> now click on object URL and you will be able to see the image

**S3 pricing Tier**

What makes up cost for S3🡪 Storage, request and data retrival, data transfer, management and data transfer.

**What are the different Tiers**

1. S3 Standard
2. S3 IA(infrequent Access)
3. S3 OneZone- IA
4. S3 Intelligent Tiering
5. S3 Glacier
6. S3 glacier Deep archive

Cost wise in descending Order

S3 standard>IA>Intelligent Tiering>one zone>Glacier>Glacier Deep archive

**S3 Security and Encryption-> by default all buckets are private, you can implement**

Bucket policies(For the entire bucket) and Access Control Lists(for the individual objects) on buckets.

Encryption🡪so that no one would be able to eavesdrop your data will be able to understand it. It is achieved by

🡪SSL/TLS

🡪Encryption at REST (Data at Server side is encrypted and stored) It could be Server Side Encryption or client side encryption. For server side encryption (AWS will encrypt the data for you) for client side(you have to do it using your own algo)

Server Side Encryption using S3 Managed Keys (SSE-S3)

AWS Key Management System (SSE-KMS)

Server side encryption with Customer provided Keys (SSE-C)

Lab🡪 Change Encryption to AES256

**S3 Version control**

Used to store all versions of an object. It’s a great Backup tool.

Versioning cannot be disabled it can only be suspended(you have to delete the old bucket and create a new one).

It could be integrated with life cycle rules(moving objects from one storage to other like shifting to S3 glacier)

Versioning’s MFA Delete capability, provides an extra layer of security.

**Versioning Lab**

1. Create a new bucket
2. Add a text file to it.
3. Make it public and access it, upload new version and make it public again.
4. You can add three different versions and check, in the bucket Storage capacity and File details.
5. Check the total storage will be exponentially increased.
6. All newly added objects will be private.

**S3 Life Cycle Management and Glacier**

1. Click on bucket>> Management>>Add Lifecycle Rule>>Enter rule Name>> Enter Tag
2. Click Next>>storage class Transition>>Current+previous>>
3. Add transition for current version>>Standard IA after 30 days
4. Add transition for current version>>S3 Glacier after 60 days acknowledge the same
5. Similarly do it for previous version
6. Click on next>> configure expiration
7. Expire current version after 425 days and delete previous version after 425 days
8. Click on Save
9. Which will create our own life cycle rule

**Summary Life cycle Management**

1. **Automates moving of objects between different storage tiers**
2. **It can be used in conjunction with versioning**
3. **It could be applied to current and previous version**

**What is S3 prefix 🡪 so prefix is like a folder within our bucket to store our objects**

**Example:**

**Mybucket/folder1/subfolder1/image.jpg-🡪 here folder1/subfolder1 is prefix**

**S3 has extremely low latency, you can get first byte out of bucket within 100-200msThis prefix increases. You can achieve a high number of request/secnd/prefix**

**You can get better performance by using different prefixes**

**S3-Cross Account Access**

There are 3 ways through which we can share a bucket across Accounts.

1. Using bucket policies and IAM; its programmatic access only
2. Bucket Access Control, Programmatic access
3. Using IAM roles, which allows programmatic and console access.

**Cross Region Replication**

1. Create a Bucket with a unique name.
2. Management>>Replication>>create a rule>> enable versioning>>cross region replication>>get started>>create a new bucket>> choose a different region>> replicate
3. Check if ur replicated bucket contains your file?? NO
4. Add new version in original bucket and check in your replicated bucket now
5. Delete from original and check status?? It wont get deleted from replicated bucket
6. **Summary**

**Replication requires versioning**

**Files in the bucket wont get replicated**

**Subsequent updated files will be replicated**

**Delete markers are not replicated**

**Delete individual version will not be working**

**S3 Transfer Acceleration**

S3 transfer acceleration utilizes the CloudFront edge network to accelerate your uploads to S3. Insead of loading directly to your S3 bucket; you can use a distinct URL to upload directly to an edge location which will then transfer that file to S3. You will get a distinct URL to upload to

1. **AWS DataSync**

Used to move large amount of data from on-premise to AWS.

**CloudFront**- is a CDN(content delivery system) is a system of distributed servers that delivers webpages and other web content to a user bed on the geographical locations of the users, the origin of the web page and a connect delivery server.

If a web data is available at London region, we all used to get it from there using our network

london

**Key Terminologies**

Edge location🡪location where data is cached

Origin🡪 origin of all files that the CDN will distribute. This can be S3 bucket, an EC3 Instance, an Elastic Load Balancer or Route53

Distribution-

Edge location

Edge location

Edge location

london

Edge location

Edge location

Edge location

**CloudFront Lab**

1. Cloud front is a Content Delivery netwok.
2. Improves read performance, as conetent is cached at the edge.
3. There are total 216 edge locations
4. DDos (distributed denial service)protection
5. Can expose external HTTPS

**CloudFront Origins**

1. S3
2. Custom Origin(HTTP) App LB, EC2 instance, s3 website

S3 or HTTPs

How Cloud front works at high level

Cloud front url

Client

sends a request

Edge Location

**CloudFront Geo Restriction**

1. You can restrict who can access your content

**Cloud front signed URL/signed cookies**

Distribute paid shared content to premium users.

We can use Cloudfront signed URL/Cookies and we can attach a policy which include

URL expiration

IP ranges to access data

Trusted signers

Signed URL= access individual file

Signed Cookies= access multiple files

Cloud front>>create distributions>>its a global service>>Web distribution>> getstarted>>

Origin domain name: name of bucket

Origin path: you have any folder inside S3, we can specify that path. (leave it blank)

Keep everything default >> click create distribution>> it will take time to create>> once done it will show>> status:deployed and enabled>> we will get domain name and ID>>

Copy domain name>> go to browser>> copy that domain name and name of S3 file name.. you will see the content on browser.

Its much more faster.

To invalidate it🡪click on cloud front>>Distribution setting>>Invalidations>>Create invalidations>> you can invalidate a file or entire directory>>

You can specify /\* to invalidate entire cloud front>> click on invalidate

Then go to cloud front and disable the entire cloud front>> it will take around 15mins>>

Once disabled you can delete it.

Summary: Edge location, Origin, Distribution(collection of edge locations) it can WebDistribution(for websites) or RTMP (for media streaming).

Edge locations are not jst read only, you can also write something.

**CloudFront Signed URL’s and Cookies**

CloudFront🡪WebDistribution for web application

RTMP for Media streaming

1 file=1signed url

Multiple files= 1cookie

Whn we create signed url/cookies, we can attach a policy which can include

-> url expiration

-> IP ranges

-> Trusted users

Summary

Use signed url/cookies when you want to secure content so that only the people you authorize are able to access it.

Signed url is for individual file, 1 file=1 URL

1 signed cookie is for multiple files. 1 cookie= multiple files.

If your origin policy is EC2, the use cloudfront.

SNOWBALL

Big disks (Petabyte scale data transport solution that uses secure applications to transfer large amount of data in and out of AWS. Using snowball addresses common challenges with large scale data transfer includes high netweok cost, lond transfer times and security concerns. Transferring data with snowball is simple,fast,secure and can be as little as 1/5th cost of high speed internet.

Snow ball comes in 2 flavours 50TB and 80TB. Snowball edge is a 100TB data transfer device with on-board storage. It has compute and storage facility.

1. **EC2🡪 elastic compute cloud**

Web service, which provides resizable compute capacity in the cloud. EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capcity, both up and down, as your computing requirement changes.

EC2 Pricing Models🡪

1. On Demand (allow you to pay a fixed rate by the hour) with no commitments. It is useful for

🡪 users that want the low cost and flexibility of Amazon EC2 without any up-front payment or long term commitment.

🡪Applications with short term, spiky or unpredictable workloads

🡪Applications being developed or tested on Amazon EC2 for the first time.

1. Reserved- provides you with a capacity reservation and offer a significant discount on the hourly charge for an instance. Contract term can be 1yr/ 3yr term

🡪Applications with steady state or predictable usage

🡪Applications that require reserved capacity

🡪users able to make upfront payment to reduce their total computing cost even further.

1. Spot- Enables you to bid whatever price you want for the instance capacity, providing for even greater saving if your applications have flexible start and end times.

🡪applications that have flexible start and end times

🡪applications that are only feasible at very low compute prices

🡪users with urgent computing needs for large amounts of additional capacity.

1. Dedicated Hosts- physical EC2 server dedicated for your use. Dedicated hosts can help you reduce cost by allowing you to use your existing server-bound software license.

🡪Useful for regulatory requirement that may not support multi-tenant virtualization

🡪Great for licensing which does not support multi-tenant or cloud deployments

🡪Can be purchased On-Demand (hourly)

**LAB EC-2**

Services>>EC2>>lauch instance>> choose an AMI(amazon machine Image)>> Linux>>t2 micro(Free tier)>>

no. of instance:1

Network: default VPC

You can select subnets🡪 us-east1A (AZ)

Default>>

Add Storage(RAM)>>

Ad Tags>>

Configure security group(to enable traffic over network)

SSH TCP .0.0.0

HTTP TCP 0.0.0.0

>> Review and launch>>

Create public and private key>> give a name and download to a safe location

>> it will launch an instance in some time

To connect to this instance we can

Terminal for mac

plugin for chrome

easiest way is Connect button

windows users>> go to chrome browser and add a plugin Secure shell App>> launch it

click on secure shell App>>

get public IP address of your instance>>

go to chrome browser SSH plugin

enter your public IP address in hostname, username is ec2-user

now for Identity you need to convert your pem file

go to command prompt, go to the directory where you have stored your pem file

type the command

**ssh-keygen -y -f linuxkeypair.pem>linuxkeypair.pub**

**generating public key, this is asymmetric encryption.**

**Rename your pem file to without extension file**

**Ren linuxkeypair.pem linuxkeypair**

Go back to your chrome extension

Import both pub and the other file without extension

New Connection and click on ENTER

It will start your instance on windows OS

**Type Sudo su**

**Type clear**

**Yum update –y**

***NO Updates, as everything is upto date***

**Yum install httpd –y (to install Apache, it will convert our EC2 instance to a web server)**

**Once done**

**Type Cd /var/www/html**

**You are going to your web directory, what all you type there will be converted to a web application**

**Type nano index.html to create a new web page**

**Write some html code there**

**<html><head><title>first App</title></head>**

**<body> <h1> Hello World Application</h1></body>**

**</html>**

**Press control+X**

**Press yes>> Enter**

**Using ls command you will be able to see the same file**

**Type service httpd start (to start httpd service)**

**Type chkconfig on (to start service if your ec2 instance accidently reboots)**

**Go to your browser and enter your IP address**

**Summary**

**After launching an EC2 instance>> for windows OS connect>> chrome browser>> SSH chrome extension plugin>> add that plugin to browser>> launch>> add usrename (ec2-user)>> host name IP address of your launched instance>>Import public and private key>> click on new connection and start your instance**

**Once started>>yum update>> yum install httpd –y >> cd /var/www/html>>nano index.html**

**>>ls(to check your index.html) >> service httpd start>> chkconfig on**

**>> go to browser and add your IP address and check if your html page is available or not.**

**LAB-2 EC-2**

Check description of instance

Then check status of instance

**Summary**

* Termination protection is turned off by default, you can turn it on.
* On an EBS backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated
* EBS root volume of your default AMI’s can be encrypted. You can also use some third party tool to encrypt the root volume

**Security groups**

Go to EC2 instance>> Description>> security groups>>click on view inbound rules>> to see which all ports are allowed.

**On the left pane>> click on security groups>>edit inbound rules>> delete http IPV4 and IPV6>> check if previous o/p is available or not**

Go to default security group>> edit inbound traffic>>Add rule>>add MySQL and Save rule>> you can add multiple security groups to an EC2 instance

Go to EC2 Dashboard>>click on running instance>> Actions>>Networking>>change security rules>> you can assign a security group.

**Summary**

**All inbound traffic is blocked by default**

**All outbound traffic is allowed.**

**Changes to security gp. Take effect immediately**

**You can have any number of EC2 instance within a security group**

**You can have multiple security groups attached to EC2 instances**

**Security groups are stateful (which means if you create an inbound rule allowing traffic in, that traffic is automatically allowed back again.)**

**You cannot block specific IP addresses using Security Groups, instead use Network Access Control Lists.**

**You can specify allow rules, but not deny rules**

**EBS 101**

Amazon Elasic block store(EBS) provides persistent blockRDS storage volumes for the use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its AZ to protect you from component failure, offering high availability and durability

5 types

1. General type(SSD)🡪 for most workloads
2. Provisioned IOPS(SSD)🡪Databases
3. Throughput IOPS SSD



**IAM Roles with EC2**

IAM>> create roles>>Ec2>>administrator access>>give a name like admin Access>>Done

Go to EC2>>go to ssh>>connect your instance>>post connecting

Type: Cd ~

Ls (u will not be able to see anything) but there is a hidden directory

Aws s3 ls (unable to access credentials)

Go to EC2 instance>>Actions>>instance setting>> IAM roles>> click on the role u have created>>click on apply>> you can also see this IAM role in properties

Now go back to SSH and click the command “ aws s3 ls” you have full admin access and u will be able to see bucket list details

**Once done Delete EC-2 instance**

**Summary**

**Roles are more secure than storing ur access key and secret access key on individual EC2 instances.**

**Roles are easier to manage**

**Roles can be assigned to an EC2 instance after it is created.**

**Roles are universal**

**Creating EC2 instance using scripts**

**EC2 using boot strap scripts**

#!/bin/bash

yum update -y

yum install httpd -y

service httpd start

chkconfig httpd on

cd /var/www/html

echo "<html><h1>Hello Cloud Gurus Welcome To My Webpage</h1></html>" > index.html

aws s3 mb s3://YOURBUCKETNAMEHERE

aws s3 cp index.html s3://YOURBUCKETNAMEHERE

**EC2 Meta Data**

Go to ssh

Sudo su

Clear

curl <http://169.254.169.254/latest/user-data>

you will get full data you have fired previously

curl [http://169.254.169.254/latest/user-data>bootstrap.txt](http://169.254.169.254/latest/user-data%3ebootstrap.txt)

(to shift the data to bootstrap.txt)

Cat bootstrap.txt --🡪 to see the data in the txt file

For meta data

curl <http://169.254.169.254/latest/meta-data/>

**to see data**

curl <http://169.254.169.254/latest/meta-data/local-ipv4>

curl <http://169.254.169.254/latest/meta-data/public-ipv4>

**EFS🡪Elastic File system**

It’s a file storage service for amazon elastic compute cloud instance. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

**Benefits and Features**

### **Dynamic elasticity**

Amazon EFS automatically and instantly scales your file system storage capacity up or down as you add or remove files without disrupting your applications, giving you the storage you need – when you need it. You simply create your file system and start adding files with no need to provision storage in advance.

### **Fully managed**

Amazon EFS is a fully managed service providing shared file system storage for general purpose workloads. It provides a simple interface allowing you to create and configure file systems quickly and manages the file storage infrastructure for you, removing the complexity of deploying, patching, and maintaining the underpinnings of a file system.

### **Scalable performance**

Amazon EFS is designed to provide the throughput, IOPS, and low latency needed for general purpose workloads. Throughput and IOPS scale as a file system grows and can burst to higher throughput levels for short periods of time to support the unpredictable performance needs of file workloads. For the most demanding workloads, Amazon EFS can support performance over 10 GB/sec and up to 500,000 IOPS.

### **Shared file storage**

Amazon EFS provides secure access for thousands of connections. Amazon EC2 instances and on-premises servers can simultaneously access a shared Amazon EFS file system using a traditional file permissions model, file locking capabilities, and hierarchical directory structure via the NFSv4 protocol. Amazon EC2 instances can access your file system across AZs and regions while on-premises servers can access using AWS Direct Connect or AWS VPN.

**Lab:**

Create 2 EC2 instances, create a file load it on EFS, modify it and check if one single copy is existing or not.

Boot strap script

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

service httpd start

chkconfig httpd on

yum install -y amazon-efs-utils

**LOAD BALANCER**

Distributes traffics,

EC2 Instance-2

EC2 Instance-1

Checks the health of instance

High availability of application

Fault tolerance

Load balancer

Elasticity

Security, robust networking and

Security

**Types of Load balancer**

1. Classic- used to load balance for highly available
2. Application Load balancer- provides application level routing options, it routes the traffic based on application content, spread on multiple EC2 instances, services, microservices
3. Network Load balancer🡪 it handles TCP , have high capacity to handle millions of requests/second
4. Left hand pane>> Load balancer>>classic load balancer>>give name>>rest default>>click on assign security group(with HTTP port)>>configure health check>>add EC2 instances(attach both EC2 instances to Load balancer)>>add tag>> create
5. Put load balancer DNS name in browser and check if its working fine or not

**AutoScaling groups**

1. Create Load balancer, don’t select any instances
2. Launch config on left pane i. AMI

Storage

Security

Scripts/commands

1. Auto scaling groups (responsible to create instaces,trigger alarm)

Create instances in multiple AZ(subnet)

Click on Advance options and select your load balancer

Health check of LB

Click on USE SCALING POLICY

Add Increase and decrese instance details and set up alarm

* 1. #min instance
  2. #max instance
  3. Alarm
  4. Scaling policy

1. Check the instances are created in instance tab
2. Check LB in LB tab, and it will display instances attached to LB
3. Manually terminate an instance and check, what will happen
4. You can also Test autoscaling policies using stress

**VPC**

Virtual private cloud it’s a separate dedicated isolated network to AWS account to host instances. We can specify our own routers, our own rules and all. In every region there is a default VPC.

Services>>VPC>>cerateVPC>>give name>>CIDR 10.0.0.0/16>>No IPV6>>default>>create and close

Inside VPC we can create subnets, so that we can divide our VPC into some more logical instance.

Public subnet private subnet

Microservice instances

Public IP

Database instance

Public subnet 10.0.1.0/24

Private subnet 10.0.2.0/24

All subnets are by default private.

Inside public Subnet🡪 select it>>actions>>modify auto assign IP setting>>Auto assign IPV-4>>enable it

VPC

Internet gateway

Internet Gateway

Leftpane>>internetGateway>>create internetgateway

Attach IG to appropriate VPC

Action>>inside VPC>>select your VPC

**To provide internet only for public subnet we need to create a Route Table**.

Associate it with VPC

Click on actions>> associate it with public subnet>>

Associate Route table with Internet gateway

Click on routes>>add route>>destination 0.0.0.0/0>>target Internetgateway

**SSH and HTTP ports enabled on WebServer instance**

VPC

Web Server

Database Server

Internet gateway

Bastion server

Nat gateway

DB team

**Security group**

1. Create an EC2 instance
2. Select ur created VPC
3. Select ur public subnet
4. Go to security group
5. SSH -🡪Source MY IP
6. HTTP🡪 anywhere

**Bastion server**

SSH-->My IP

**DBServer**

SSH of Bastion server private IP address 10.0.1.163/32

MySQL/Aurora -🡪10.0.1.0/24

Check if bastion server is connected with DB instance or not

Start ssh on baston server & from Bastion server SSH to DBServer

To connect with db server we can use the command **ssh -i "newLinuxKey.pem" ec2-user@10.0.2.230**

Wherein [ec2-user@10.0.2.230](mailto:ec2-user@10.0.2.230) is the ip address of dbserver

But u have to transfer **newLinuxKey.pem to linux instance**

**For that use WinSCP tool**

**Enter username: ec2-user**

**Enter hostname: baston server IP address**

**Click on Advanced>> Authenticate>> Browse the location of pem file>>**

**Jst drag the pem file from windows to linux system.**

**And check in putty, using ls –la id pem file is available or not. Using the command**

**ssh -i "newLinuxKey.pem"** [**ec2-user@10.0.2.230**](mailto:ec2-user@10.0.2.230)

**you can check the connection again**

but db server doesn’t have any internet connection. Even if you try to do yum httpd update –y, it wont work.

To establish internet connectivity, we have to create **NAT Gateway**

**VPC>> left pane>>Nat gateway>>cerate it in public subnet>>allocate elastic IP address>>create a Natgateway.**

**Create a RouteTable>>RT-2>> in our VPC>>create>>**

**Route table>>click>> subnet associations>>edit subnet associations>>associate it with private subnet>>**

**Click on Routes>>Edit routes>>in Target add NAT GATEWAY ID>>save routes**

**Now check in SSH using command sudo su**

**Yum update httpd –y**

**It will start updating.**

**NACL-🡪network access control Lists**

**It is at subnet level.**

**By default everything is denied(inbound and outbound is denied).**

**NACL have rules#**

**Each rule no. can enable a particular type of communication.**

**Separate rules to allow or deny**

**Storage (S3, Glacier, EFS,StorageGateway,SnowBall)**

1. Block Storage🡪(EBS) to install OS, DBS, platforms, runtime, hosting apps, web servers and so on
2. Object Storage🡪 static files, Objects, S3 , Glacier
3. File Storage🡪 (EFS)when we need a directory structure

We can also create a volume and store our data.

(EBS) Console>>EC2>>Left pane>> Elastic Block store>> Volume>>create Volume

Once volume is created>> Actions>>attach volume to the instance (you an only attach this with one instance)

>> Create a windows instance>>copy instance ID from properties>>go to Volumes>>Actions>>attach volume>>paste instance ID>>Device(any name)

🡪You can format it

**Volume Types**

1. General purpose SSD🡪 support 3000 iops, ideal for all workload types, splly for root Volume, used for development/ testing wokloads
2. Provisioned IOPS SolidStateDrive🡪used for large DB workloads, SQL/Oracle critical ; used for IO intensive applications; which needs consistent throughput

Max 64000iops

1. Throughput optimized HDD🡪 data ware housing, for throughput optimize, 16tB
2. Cold HDD🡪 cost effective storage, when we need to access infrequently.
3. Magnetic🡪 low cost

EFS🡪 network attached file system.

🡪Several EC2 instances can use single file system

🡪Only used to store files and folders

Object Storage🡪 file with some metadata (google drive/onedrive)

Offers Versioning

**Object Storage S3 (Secure, scalable, high available)**

Buckets (folders)

Enables Versioning

Store and retrieve any amount of your data

Tiered storage (different classes)

-standard

-intelligent tiering

- standard IA

- one zone IA

-Glacier

- Glacier Deep archive

Life cycle Management (Example of News Paper)

Satic Website hosting

**LAB**

Service>> S3>>create bucket>> unique name>> Region>>

1. **S3 (Simple Storage Service)**

It’s a safe place to store your objects. Like files.

* Files can be from 0 Bytes to 5TB
* There is unlimited storage
* Files are stored in Buckets (it’s a folder)
* S3 is a universal name space, it means name must be unique as it creates a WebAddress.
* When you upload a file successfully to S3 bucket, you will receive a status code of 200
* Objects consist of following

Key (name of the Object)

Value (This is simply the data, made up of sequence of bytes)

Version Id (important for Versioning)

Metadata (Data about data like Tagging)

**Data Consistency for Files**

1. If you put a data, data will be available immediately
2. If you update the same file, there can be a possibility that data will be available after some time (It may give you version1 or Version2) but if you wait for some time, it will give u version2.

**S3 has following guarantees**

1. Built for 99.999% availability and durability

It is unlikely that it will be lost

S3 has following features

1. It has Tiered storage
2. Life cycle management ( we can move a file from one tier to other)
3. We can version a file
4. We can encrypt a file
5. MultiFactor Authentication deletion
6. You can also secure your data using Access Control List and bucket policies.
7. Its not suitable to install an OS

**S3 classes**

1. S3 IA(infrequent Access)- for the data that is accessed less frequently, but requires rapid access when needed. Lower fess as compared to S3 but has retrieval charges.
2. S3 One Zone –IA🡪 for where you want a lower cost option for infrequently accessed data, but do not need multiple AZ
3. S3 Intelligent Tiering (introduced in 2018)🡪 it is designed to optimize cost by automatically moving the data to the most cost effective access tier without effecting the performance or operational overheads. This is using ML, it checks how frequently u are accessing your data. Accordingly it moves your data
4. S3 Glacier comes in 2 flavors🡪 S3 glacier and S3 glacier deep Archive.

**S3 charges you in following ways**

1. Based on storage
2. Requests
3. On the storage Management pricing based on tiering
4. Data Transfer pricing
5. Transfer acceleration
6. Cross region replication pricing

**What is cross region replication?**

If we have a bucket in N-America, and we want it to be replicated in Sydney region, so as to have a proper backup in case of any disaster, we can do that for security. Asa you upload an object in your bucket, it will get replicated in other region as well.

**S3 Transfer Acceleration**- it enables fast easy and secure transfer of files over long distance between end user and S3 Bucket. It takes advantage of Edge Location, asa data arrives at edge location, data is routed to S3 over an optimized network path.

**Lab S3**

Remember S3 is stored in global region.

Services>>Storage>>S3>>Create Bucket>>Enter Bucket name which has to be unique globally>>Select a Region>>Clcik on Next>> No versioning as of now>>Click on next >>by default all public access is denied, we will keep it blocked>> click on Create bucket after viewing preview.

Click on the bucket to see its Objects,>> Select two images>>Upload>> you will see Progress of upload>>if you click on the object, will be able to see its details.

You cannot make it public, until the bucket itself is public.

So move to bucket, click its checkbox.>> Edit public access>> uncheck Block public access>>Save>>confirm

Now go inside your bucket>> select the file you want to make public>> click on Actions>> Make public>> now click on object URL and you will be able to see the image

**S3 pricing Tier**

What makes up cost for S3🡪 Storage, request and data retrival, data transfer, management and data transfer.

**What are the different Tiers**

1. S3 Standard
2. S3 IA(infrequent Access)
3. S3 OneZone- IA
4. S3 Intelligent Tiering
5. S3 Glacier
6. S3 glacier Deep archive

Cost wise in descending Order

S3 standard>IA>Intelligent Tiering>one zone>Glacier>Glacier Deep archive

**S3 Security and Encryption-> by default all buckets are private, you can implement**

Bucket policies(For the entire bucket) and Access Control Lists(for the individual objects) on buckets.

Encryption🡪so that no one would be able to eavesdrop your data will be able to understand it. It is achieved by

🡪SSL/TLS

🡪Encryption at REST (Data at Server side is encrypted and stored) It could be Server Side Encryption or client side encryption. For server side encryption (AWS will encrypt the data for you) for client side(you have to do it using your own algo)

Server Side Encryption using S3 Managed Keys (SSE-S3)

AWS Key Management System (SSE-KMS)

Server side encryption with Customer provided Keys (SSE-C)

Lab🡪 Change Encryption to AES256

**S3 Version control**

Used to store all versions of an object. It’s a great Backup tool.

Versioning cannot be disabled it can only be suspended(you have to delete the old bucket and create a new one).

It could be integrated with life cycle rules(moving objects from one storage to other like shifting to S3 glacier)

Versioning’s MFA Delete capability, provides an extra layer of security.

**Versioning Lab**

1. Create a new bucket
2. Add a text file to it.
3. Make it public and access it, upload new version and make it public again.
4. You can add three different versions and check, in the bucket Storage capacity and File details.
5. Check the total storage will be exponentially increased.
6. All newly added objects will be private.

**S3 Life Cycle Management and Glacier**

1. Click on bucket>> Management>>Add Lifecycle Rule>>Enter rule Name>> Enter Tag
2. Click Next>>storage class Transition>>Current+previous>>
3. Add transition for current version>>Standard IA after 30 days
4. Add transition for current version>>S3 Glacier after 60 days acknowledge the same
5. Similarly do it for previous version
6. Click on next>> configure expiration
7. Expire current version after 425 days and delete previous version after 425 days
8. Click on Save
9. Which will create our own life cycle rule

AWS CLI

Go to IAM>>Go to user>> create a user with programmatic access(access key will be generated)>> give admin access

Go to EC2 dashboard>> create a new instance>>ssh into it

Aws s3 ls>> it will ask for credentials>>aws configure>> enter access key>> enter region>> now try to use the same command>> aws s3 ls , u will find all buckets in ur region

You can also make a bucket using command aws s3 mb s3://bucket name

But if you use command cd ~, you will move on to ur root

Type ls, you won’t find anything

But use cd .aws-🡪 you will move to hidden directory .aws>> do ls and you will find config file

Use nano credentials to see your credentials

Rm –rf .aws to remove config file

Go to IAm>> create a new role and give it admin access>>

Go to EC2 and give it IAM access

Now you will be able to access your S3 buckets through CLI

CLI commands to create a bucket **aws s3 mb s3://***bucket-name*

To list bucket content **aws s3 ls** *s3://bucket-name*

To delete a bucket **aws s3 rb** *s3://bucket-name/*

**To force delete a bucket aws s3 rb** *s3://bucket-name* **–force**

**Delete all objects of s3 bucket** aws s3 rm s3://<bucketname> –recursive

Copy content of S3 file into EC2 instance

aws s3 cp s3://rincynewbucket12345/API.txt API.txt

[**https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3-commands.html#using-s3-commands-managing-buckets-creating**](https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3-commands.html#using-s3-commands-managing-buckets-creating)

**Summary Life cycle Management**

1. **Automates moving of objects between different storage tiers**
2. **It can be used in conjunction with versioning**
3. **It could be applied to current and previous version**

**What is S3 prefix 🡪 so prefix is like a folder within our bucket to store our objects**

**Example:**

**Mybucket/folder1/subfolder1/image.jpg-🡪 here folder1/subfolder1 is prefix**

**S3 has extremely low latency, you can get first byte out of bucket within 100-200msThis prefix increases. You can achieve a high number of request/secnd/prefix**

**You can get better performance by using different prefixes.**

**S3 CLI Access**

1. Install latest aws cli for windows from  <https://awscli.amazonaws.com/AWSCLIV2.msi>
2. Run msi
3. Open cmd and type aws --version you must receive aws-cli/2.1.6 Python/3.7.9 Windows/10 exe/AMD64 prompt/off as O/P
4. Open IAM on aws management console
5. Open ur user
6. Click on security credentials
7. Generate access key and secret access key
8. Come back to cmd
9. Type **aws configure**
10. Copy and paste access key and secret access key; default region and default o/p as none
11. While u run this command, it creates a small file internally
12. If u want to check that file, type aws configure list
13. In linux ls ~/.aws you will see 2 files config and credentials
14. To check config file type cat ~/.aws/config
15. You will see aws region
16. To check credentials type cat ~/.aws/credentials

**CLI on EC2 instance**

Its not a best practice to set your credentials on .aws file, so

IAM roles can be assigned to an aws EC2 instance

**S3-Cross Account Access**

There are 3 ways through which we can share a bucket across Accounts.

1. Using bucket policies and IAM; its programmatic access only
2. Bucket Access Control, Programmatic access
3. Using IAM roles, which allows programmatic and console access.

**Cross Region Replication**

1. Create a Bucket with a unique name.
2. Management>>Replication>>create a rule>> enable versioning>>cross region replication>>get started>>create a new bucket>> choose a different region>> replicate
3. Check if ur replicated bucket contains your file?? NO
4. Add new version in original bucket and check in your replicated bucket now
5. Delete from original and check status?? It wont get deleted from replicated bucket
6. **Summary**

**Replication requires versioning**

**Files in the bucket wont get replicated**

**Subsequent updated files will be replicated**

**Delete markers are not replicated**

**Delete individual version will not be working**

**S3 Transfer Acceleration**

S3 transfer acceleration utilizes the CloudFront edge network to accelerate your uploads to S3. Insead of loading directly to your S3 bucket; you can use a distinct URL to upload directly to an edge location which will then transfer that file to S3. You will get a distinct URL to upload to

**RDS-Relational Database service**

-fully managed service, high availability of Data

-cost effective

-resizable

-supports mysql,oracle,postgresql,mariadb, sql server

-AWS aurora

- automatically manages backups, software/OS patching,failover

- Scale CPU, Storage, Memory

-we can create a DB instance

- choose instance type

- storage capacity

- db engine

-enable autobackup

-Create read replicas

**LAB**

RDS>> Create Database>>Database Engine>>MySQL>>Version>>

Create a database instance and check..

Check out difference between Amazonaurorra,MySQL and POSTgreSQL

**AWS Lambda Service**

-> Its an event driven server less computing platform.

-> the code we run on lambda is called lambda function

-> the lambda function runs whenever it is triggered by pre configured event sources like SDK,dynamodb, S3, cloudwatch

Container contains Lambda.

LAB-🡪

Services>>Lambda>>create function>>

**Persistence IN AWS**

**RDS**

1. What is RDS

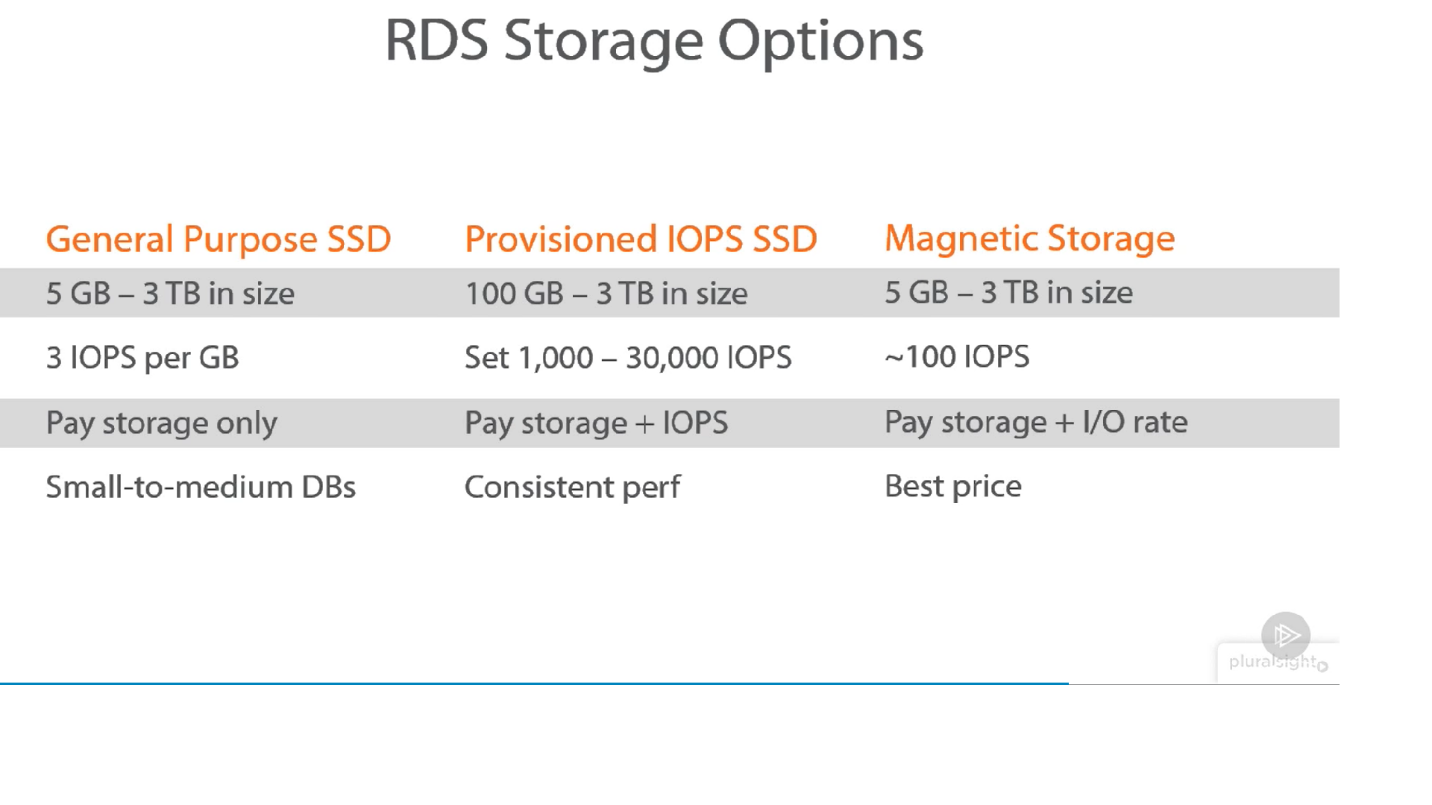
Core Concept \_.DB instance, replication, Data Backups, Multiple DB engine

Native capabilities of any DB is

DB set up, auto backup, manual Backup, restore from snapshot, point I time recovery.

What all DB are allowed

1. Mysql
2. Postgres SQL
3. Oracle EE|Std Edition|std edition1
4. Microsoft SQL Server Express
5. Amazon Aurora(MySQL compatible)



How to choose a DB

1. How much storage is needed
2. What type of storage makes sense
3. Where should this db run
4. What level of resilience is necessary?
5. Which db engine should I Use?

LAB

1. Service>>RDS>>MYSQL>>check box in the bottom(only enable options eligible for RDS free Usage tier)>>Next>>Specify DB Details>>keep everything default>> Go to Settings>> enter a unique DB instance identifier>> give a Master un>> and Master password and confirm password>> configure advanced settings
2. Once DB has created, we need DNS name/EndPoint copy that end point
3. Open Mysql work bench
4. Enter host as the copied endpoint
5. Enter un and pw and log i

Also we can connect through JDBC code

**public** **class** DemoJDBC {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

Class.*forName*("com.mysql.cj.jdbc.Driver");

System.***out***.println("Driver loaded");

String hostname="rincyinstance.cv457ufx06u8.us-east-1.rds.amazonaws.com";

String port="3306";

String dbName="adminUser";

String userName="root";

String password="emmunuel";

String jdbcUrl = "jdbc:mysql://" + hostname + ":" +

port + "/" + dbName + "?user=" + userName + "&password=" + password;

Connection con=DriverManager.*getConnection*(jdbcUrl);

System.***out***.println("Got connection");

Statement smt=con.createStatement();

ResultSet rs=smt.executeQuery("Select \* from Employee");

**while**(rs.next())

{

System.***out***.print(rs.getObject(1)+"\t"+rs.getString(2)+"\t"+rs.getFloat(3));

System.***out***.println();

}

}

}

**DynamoDB**

1. Fully managed service by AWS. No need to think about OS, pf
2. NO SQL DB service
3. Provides fast, consistent and scalable db(Vertical and horizontal).
4. It supports key value and Doc Data Model.
5. We store data in unstructured manner.
6. It comes under SERVERLESS service of deployment.
7. Cost effective, highly available.
8. It internally replicates the data, to ensure high availability
9. It has a partition Key

LAB

Services>>DynamoDB>>Create a Table>>Create Items from Item Tab

Show TTL and how to add different attributes

**CloudFormation**

Create and customize infra using code(JSON/YAML)

Create a template to create some resources for you, like EC2 instance, s3 bucket, or lambda

A template is a declaration of the AWS resources that make up a stack. The template is stored as a text file whose format complies with the JavaScript Object Notation (JSON) or YAML standard. Because they are just text files, you can create and edit them in any text editor and manage them in your source control system with the rest of your source code.

1. Resources
2. Parameters
3. Mappings
4. o/p
5. MetaData
6. Conditions

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-anatomy.html>

**Lab**

Create an EC2 instance through cloud formation template.

Try to replace it with other template

Show How to avoid termination of instances.

**ElasticBean**

For Sample Application

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/tutorials.html>

Services>>ElasticBeanStalk>>Create instance>> name>> nodejs>>give tag name>>upload your app from local>> it will take sometime >> go to configuration to see the application

You can also create a VPC, Load balancer and so on

Create Environment>> WebServer envt>>

**SNS (Simple Notification service)**

Fully managed service.

Similar to a whats app group.

It’s a push based messaging system, which means there is a producer, who produces a message through this SNS.

This can be a text, info about transaction, also offers us to create a topics.

Subscriber

Email,lambda,sms…. Who will receive and manipulate the message

Producer

EC2, LB,

Publish/send msg

SNS (Topics)

Msgs are immediately delivered not stored

**LAB**

Services>>SNS>>Topic>>Create Topic>>send msg

Using S3 publish a message

Through SDK

### **Install the Toolkit**

1. Open Help → Install New Software….

2. Enter https://aws.amazon.com/eclipse in the text box labeled “Work with” at the top of the dialog.

3. Select the required "AWS Core Management Tools" and other optional items from the list below.

4. Click “Next.” Eclipse guides you through the remaining installation steps.

Restart Eclipse

It will ask for Access and Secret Key

1. To generate that, go to ROOT account
2. My Security Credentials
3. Generate Access Key
4. It will generate, download the csv file and keep it safe
5. Create A maven project
6. Add aws sns dependency

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-sns</artifactId>

<version>1.11.847</version>

</dependency>

1. Create a new Java class

Get topic code through

<https://docs.aws.amazon.com/sdk-for-java/v2/developer-guide/examples-simple-notification-service.html>

publish message

https://github.com/awsdocs/aws-doc-sdk-examples/blob/master/javav2/example\_code/sns/src/main/java/com/example/sns/PublishTopic.java

Program

**package** com.cg.sns;

**import** com.amazonaws.services.sns.model.SubscribeRequest;

**import** software.amazon.awssdk.regions.Region;

**import** software.amazon.awssdk.services.sns.SnsClient;

**import** software.amazon.awssdk.services.sns.model.PublishRequest;

**import** software.amazon.awssdk.services.sns.model.PublishResponse;

**import** software.amazon.awssdk.services.sns.model.SnsException;

**public** **class** TestSNS {

/\*public static void pubTopic(SnsClient snsClient, String message, String topicArn) {

try {

PublishRequest request = PublishRequest.builder()

.message(message)

.topicArn(topicArn)

.build();

PublishResult result = snsClient.publish(request);

System.out.println(result.getMessageId() + " Message sent. Status was " + result.sdkHttpResponse().statusCode());

} catch (SnsException e) {

System.err.println(e.awsErrorDetails().errorMessage());

System.exit(1);

}\*/

**public** **static** **void** main(String[] args) {

**final** String USAGE = "\n" +

"PublishTopic - publish an Amazon SNS topic\n" +

"Usage: PublishTopic <message> <topicArn>\n\n" +

"Where:\n" +

" message - message text to send.\n\n" +

" topicArn - the ARN of the topic to look up.\n\n";

String message = "Hello AWS";

String topicArn="arn:aws:sns:us-east-1:038574925522:Topic-1";

SnsClient snsClient = SnsClient.*builder*()

.region(Region.***US\_EAST\_1***)

.build();

*pubTopic*(snsClient, message, topicArn);

}

//snippet-start:[sns.java2.PublishTopic.main]

**public** **static** **void** pubTopic(SnsClient snsClient, String message, String topicArn) {

**try** {

PublishRequest request = PublishRequest.*builder*()

.message(message)

.topicArn(topicArn)

.build();

PublishResponse result = snsClient.publish(request);

System.***out***.println(result.messageId() + " Message sent. Status was " + result.sdkHttpResponse().statusCode());

} **catch** (Exception e) {

e.printStackTrace();

System.*exit*(1);

}

//snippet-end:[sns.java2.PublishTopic.main]

}

}

**SQS(Simple Queue Service)**

A message queue is a queue of messages exchanged between applications. Queue contains messgaes

Messages are data objects that are inserted in the queue by sender applications and

Received by receiving application*s*. Receiving applications get the data objects from

the queue and process the data received from the queue based on the application

requirements.

Messgaes stay in queue for min 4 days to max 14 days.

Messages size must be less than 256KB

SQS-Message Visibility TimeOut🡪 once a message is polled by a consumer, it becomes invisible to others.

SDK Code

<https://github.com/awsdocs/aws-doc-sdk-examples/blob/master/javav2/example_code/sqs/src/main/java/com/example/sqs/SQSExample.java>

**package** com.cg.sqs;

**import** java.util.List;

**import** software.amazon.awssdk.regions.Region;

**import** software.amazon.awssdk.services.sqs.SqsClient;

**import** software.amazon.awssdk.services.sqs.model.ChangeMessageVisibilityRequest;

**import** software.amazon.awssdk.services.sqs.model.CreateQueueRequest;

**import** software.amazon.awssdk.services.sqs.model.GetQueueUrlRequest;

**import** software.amazon.awssdk.services.sqs.model.GetQueueUrlResponse;

**import** software.amazon.awssdk.services.sqs.model.Message;

**import** software.amazon.awssdk.services.sqs.model.ReceiveMessageRequest;

**import** software.amazon.awssdk.services.sqs.model.SendMessageBatchRequest;

**import** software.amazon.awssdk.services.sqs.model.SendMessageBatchRequestEntry;

**public** **class** SQSDemo {

**public** **static** **void** main(String[] args) {

String queueName = "queue" + System.*currentTimeMillis*();

SqsClient sqsClient = SqsClient.*builder*()

.region(Region.***US\_EAST\_1***)

.build();

// Create a queue

String queueUrl= *createQueue*(sqsClient, queueName );

List<Message> messages=*receiveMessages*(sqsClient, queueUrl);

*changeMessages*(sqsClient, queueUrl, messages);

}

**public** **static** **void** sendBatchMessages(SqsClient sqsClient, String queueUrl) {

System.***out***.println("\nSend multiple messages");

// snippet-start:[sqs.java2.sqs\_example.send\_\_multiple\_messages]

SendMessageBatchRequest sendMessageBatchRequest = SendMessageBatchRequest.*builder*()

.queueUrl(queueUrl)

.entries(SendMessageBatchRequestEntry.*builder*().id("id1").messageBody("Hello from msg 1").build(),

SendMessageBatchRequestEntry.*builder*().id("id2").messageBody("msg 2").delaySeconds(10).build())

.build();

sqsClient.sendMessageBatch(sendMessageBatchRequest);

// snippet-end:[sqs.java2.sqs\_example.send\_\_multiple\_messages]

}

**public** **static** String createQueue(SqsClient sqsClient,String queueName ) {

System.***out***.println("\nCreate queue");

// snippet-start:[sqs.java2.sqs\_example.create\_queue]

CreateQueueRequest createQueueRequest = CreateQueueRequest.*builder*()

.queueName(queueName)

.build();

sqsClient.createQueue(createQueueRequest);

// snippet-end:[sqs.java2.sqs\_example.create\_queue]

System.***out***.println("\nGet queue URL");

// snippet-start:[sqs.java2.sqs\_example.get\_queue]

GetQueueUrlResponse getQueueUrlResponse =

sqsClient.getQueueUrl(GetQueueUrlRequest.*builder*().queueName(queueName).build());

String queueUrl = getQueueUrlResponse.queueUrl();

**return** queueUrl;

// snippet-end:[sqs.java2.sqs\_example.get\_queue]

}

**public** **static** List<Message> receiveMessages(SqsClient sqsClient, String queueUrl) {

System.***out***.println("\nReceive messages");

// snippet-start:[sqs.java2.sqs\_example.retrieve\_messages]

ReceiveMessageRequest receiveMessageRequest = ReceiveMessageRequest.*builder*()

.queueUrl(queueUrl)

.maxNumberOfMessages(5)

.build();

List<Message> messages = sqsClient.receiveMessage(receiveMessageRequest).messages();

**return** messages;

// snippet-end:[sqs.java2.sqs\_example.retrieve\_messages]

}

**public** **static** **void** changeMessages(SqsClient sqsClient, String queueUrl, List<Message> messages) {

System.***out***.println("\nChange message visibility");

**for** (Message message : messages) {

ChangeMessageVisibilityRequest req = ChangeMessageVisibilityRequest.*builder*()

.queueUrl(queueUrl)

.receiptHandle(message.receiptHandle())

.visibilityTimeout(100)

.build();

sqsClient.changeMessageVisibility(req);

}

}

}

**Kinesis**

In today's rapidly developing world, data is an asset. For large enterprises in this

digital era, data is constantly being generated from various digital sites or sources,

such as sensors, telemetry data, audio files, videos, application or server logs, gaming

apps, and website click streams. It is, therefore, a challenge to collect, process, and

analyze large amounts of data in real time. Usually, the data of any enterprise will

exist in various formats, come from different sources, and generate data according to

individual velocity rates.

There are three major components in the preceding diagram. Let's understand them

in the following points:

Producer: Any real world source like a camera that feeds data

**Code Commit Service**

1. Create a Java project
2. Create a local repo
3. Service>>Code commit>>create a repo
4. To enable Git credentials
5. Go to IAM>>select user>>security credentials>>Git>> Generate Credentials>>store the csv file properly
6. Come to eclipse>>Create a local repo and commit the code
7. Commit to repote using the url of the remoterepo
8. Enter un and pw to confirm push
9. Check the cod ein remote repo
10. Make some changes in remote repo and commit again
11. Create a branch
12. Ur repo>>commits>>left pane Branch>> create branch>> branch name>>branch from master
13. Make conflicts and resolve it

Create Notifications

1. Go to repository>> Settings in left pane>> notifications>>Create a notification rule

**Code Build service (Similar to Jenkins)**

Fully managed service, to compile the source code

Run unit test and the artifacts will be ready to deploy

Create an S3 bucket and enable versioning

We can create a buildspec.yml file for executing the commands

<https://docs.aws.amazon.com/codebuild/latest/userguide/sample-codedeploy.html>

**version:** 0.2

**phases:**

**install:**

**runtime-versions:**

**java:** corretto8

**build:**

**commands:**

- echo Build started on `date`

- cd AWSCodeCommit && mvn install

**post\_build:**

**commands:**

- echo Build completed on `date`

- mvn package

**artifacts:**

**files:**

- target/my-javaapp-1.0-SNAPSHOT.jar

- appspec.yml

**discard-paths:** yes

**Code Deploy**

Fully managed service

Automate code deployment

Deploy the code on any instance

Deploy code to lambda functions, or ECS containerized architecture

{

"Version": "2008-10-17",

"Statement": [

{

"Sid": "eb-ad78f54a-f239-4c90-adda-49e5f56cb51e",

"Effect": "Allow",

"Principal": {

"AWS": "AROAQR6Z6U3JHGFT34B7Z"

},

"Action": "s3:PutObject",

"Resource": "arn:aws:s3:::elasticbeanstalk-us-east-1-038574925522/resources/environments/logs/\*"

},

{

"Sid": "eb-af163bf3-d27b-4712-b795-d1e33e331ca4",

"Effect": "Allow",

"Principal": {

"AWS": "AROAQR6Z6U3JHGFT34B7Z"

},

"Action": [

"s3:ListBucket",

"s3:ListBucketVersions",

"s3:GetObject",

"s3:GetObjectVersion"

],

"Resource": [

"arn:aws:s3:::elasticbeanstalk-us-east-1-038574925522",

"arn:aws:s3:::elasticbeanstalk-us-east-1-038574925522/resources/environments/\*"

]

},

{

"Sid": "eb-58950a8c-feb6-11e2-89e0-0800277d041b",

"Effect": "Allow",

"Principal": {

"AWS": "\*"

},

"Action": "s3:DeleteBucket",

"Resource": "arn:aws:s3:::elasticbeanstalk-us-east-1-038574925522"

}

]

}