19.19Apr. Cloud Basics and RDS

19 April 2023 07:17 PM

Cloud Providers(IaaS, PaaS)

1. Amazon = AWS(Amazon Web Service)
2. Microsoft = Azure
3. Google = GCP(Google Cloud Platform)

Cloud Providers(SaaS)

1. Salesforce

AWS:

- Set of services
  - **Management Console = Understanding**
  - CLI(Command) = Automation
  - SDK (Programming = Python[boto3]) = Cloud App.
  - REST API = Integration(Architecture)
  - IaaC Tool = DevOps + CloudOps
    - AWS CloudFormation
    - Terraform

Services in AWS

- Storage = S3(Bucket Features), EBS= Elastic Block Storage

S3

- Bucket & Its features
  - Versioning
  - Storage Class
  - Permission
  - IAM Policy --> S3 Bucket
  - Object Lifecycle
- PaaS

EC2

- Instance Purchasing Options
- EBS --> Volume, Snapshot, AMI
- Security Group
- Keypair
- IaaS

RDS

- PaaS
- In Background hidden mode
  - Create EC2 Instance
  - Install Database Server(MySQL)
  - Configure according to input given
- Use RDS

Task: S3 & EC2 Integration

1. S3 Bucket
2. EC2 Instance(Storage Server-SS)
3. Create Abc.txt File in SS
4. Install AWS CLI
5. Configure AWS CLI
6. Upload Abc.txt file in S3

GDrive
> PaaS
> AppScript Engine(google script)
> Create an Application

Mother Food = IaaS
PaaS = Hotel
SaaS = Swigi

IaaS = Cloud Virtual Server (Techi. Gig) = Minikube/Kubeadmn, Kuberenetes = 1 Day

PaaS = (Professional) = EKS, ECS = 1 Hr
OS : Linux
PL : Python Developer
IDE
Debugger
Compiler
Editor

SaaS = (Basics) =
WhatsApp
Don't Know OS
Don't Know PL
WordPress
UIPath
BP
Wix

1. S3
2. IAM
3. EC2
4. RDS

Infra
IaaS = On Prem --> Virtual(OnPrem --> Cloud)
PaaS = Database(RDS)
Deploy
Server Base:
On Prem./Dedicated Machine(Laptop) 40GB ---> Virtualization(VirtualBox) 20GB/WSL --> EC2(Cloud Virtual Server)
Containerization(Docker) 100MB--> Handle(K8S) 150MB
Serverless :
Containerization(Docker) 100MB--> Handle(K8S) 150MB --> Microservice(Lambda) 10MB to 200MB

Module 2: SQL
Database Client = MySQL Client ---> SQL(Structured Query Language) Command
CRUD Operations to Database Objects(**Table**, Index, View)
Create
Retrieve(Select)
Update

Delete

Module 5: Cloud(AWS)

RDS(Database Server) = MySQL Server

1. Order RDS MySQL Server
2. **EC2 Instance(Install MySQL Client)**
   a. **Order EC2**
   b. **Install MySQL Client**
   c. **Connect MySQL Server using MySQL Client**
3. *Perform CRUD Operations on Database Table*
   a. *Create Table EMPLOYEE(Name, Salary)*
   b. *Insertion AAA, 123*
   c. *Update AAA, 456*
   d. *Delete where Name=AAA*
4. S3 Integration(AWS CLI) to EC2

1. Order RDS MySQL Server
   a. Security Group
   b. Database Name: nubeera-db
   c. User Name: admin
   d. Password: y04RpyGmmtVUI5lY7xxs
   e. End Point: URL
2. EC2 Instance (MySQL Client)
   a. Order EC2 Instance (Ubuntu)
   b. Install MySQL Client
   c. Connect using Command
      i. $ mysql --version
      ii. $ mysql -h URL -u admin -p
      Enter Password:

```
$ sudo su
# apt update
# apt install mysql-client
# mysql --version
# mysql -h nubeera-db.ctdu80bhn4xh.us-east-1.rds.amazonaws.com -u admin -p
mysql> show databases;
mysql> create database nubeeradb;
mysql> use nubeeradb;
mysql> show tables;
mysql> create table emp(Name varchar(200), Salary int);
mysql> insert emp(name,salary) values('AAA',123);
mysql> select * from emp;
mysql> update emp set name='BBB';
mysql> delete from emp where name='BBB';
mysql> \q
# apt install awscli
# aws --version
# Create AK & SAK to configure AWS CLI
```

- Top Right Your Name --> Security Credentials --> Create Access Key
- AK: AKIARKD35RHMQEYPMNOM
- SAK: I8U7fJRaW+7zTAHpcECncPf6tO6PN1b/8Js4Q1hq
- Region Code: us-east-1
- Output Format: table

- # aws configure
- AK: …
- SAK: ,,,
- Region Code: us-east-1
- Output Format: Table

aws s3 ls

cat > abc.txt
welcome
Ctrl+z

cat abc.txt

aws s3 cp test.txt s3://mybucket/test2.txt

aws s3 cp abc.txt s3://nubeera-study-material/abc.txt