**AWS**

I know how to migrate an existing application to AWS and to have the application hosted on the AWS cloud.

Hosting the ~~N-tier~~ enterprise java web application entirely in AWS. By hosting the Java application on AWS the application on AWS infrastructure looks identical expect the hardware and servers will be in the cloud.

**How do you host on the AWS cloud?**

I used EC2 instances to host the application in web service

In amazon we have **EC2(Elastic compute cloud)** service🡺 it has two different types of harddrives like storage

1. Instance storage 2. Elastic block Service (EBS)🡺 It can be detached from one server and mounted on another server

**RDS instance to host our database**

AWS has Java SDK

Java SDK has API client as a JAR file

**SNS:** Simple notification service is hosted on AWS, it is similar to the JMS in Java.

Clients can be notified via HTTP, SMTP or SMS- When the cloud alarm triggered by the problem with our infrastructure/ application.

Using SNS and SQS in the Java application can help building large web scale applications.~~give many~~ ~~application characteristics on a large business scale.~~

**SQS(Simple Queue service):** It provides a queue model that allows anyone to publish a message and anyone can consume a message. It does the video encoding🡺 Like FIFO(First in first out)

In project:

There is an AWS plugin that will allow us to access/manage some AWS services from the eclipse environment.

We can do Maven integration for AWS project integration.

**Services in AWS**

Simple Storage Service (**S3**)- We can use S3 bucket to store images uploaded by the users.

There is an API in Java SDK that helps to do CRUD operations

**Amazon platform:**

**SERver**

**Ec2**

**Ec2 container service**

**Elastic Beanstalk**

**Lambda**

**Lightsail**

**Storage:**

**S3--🡪**Simple storage service

storing objects like word, files, images.

Not used for application saving.

**Glacier:**

**EFS: elastic file service**

Storing files

**Storage Gateway**

**Database**

**RDS**

Relational db

**DynamoDB**

no relation

**RedShift**

Amazon dataware service solutions

**Elasticache**

**Migration**

**Snowball**

**DMS**

**Service Migration Service(SMS)**

**ATHENA**

**EMR**

**Cloud Search**

**Elastic Search**

**Kinesis**

**Data pipeline**

**Quick sight**

**Security and identity—(IAM)**

**Inspector**

**IAM-🡪**

AWS **Identity and Access Management (IAM**) enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

**S3:**

S3 is object based and allows to upload files in cloud.

Unlimited storage

Files can be 0 bytes to 5tb

Files are stored in buckets.

S3 is universal namespace that is names must be unique globally.

If the upload was successful then we will receive http 200 code.

S3 is 99.999% 11.9’s durable

Tiered storage available.

Lifecycle management

Versioning

encryption

**Data consistency for s3**

Read after write consistency for PUTS of new objects.

Eventually consistency for overwrite PUTS and DELETES (can take some time to propagate).

**EC2**

Amazon elastic compute cloud is a web service that provides resizable compute capacity in cloud. Amazon Ec2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon Ec2 changes economics of computing by allowing you to pay only for capacity that you actually use. Amazon Ec2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

Prices:

**On demand** – allow you to pay a fixed rate by hour with no commitments.

**EC2 instance types**

D2- dense storage

R4-Memory

M4-general purpose

C4- compute optimized

G2-Graphics

I2- High speed storage

F1- field programmable gate array

T2- lowest cost, general purpose

P2-graphics/general purpose GPU

X1- memory optimized