

Lecture 2 - 12th Sept 2025

Joining AWS & Creating an EC2 instance

- pick up private key from "IAM".
- Selecting config. for the instance
which is called as "AMI".
- choose machine size (CPU, RAM, Storage, cores,).
- Security Group are RULES which help to
OPEN protocols → similar to a FIREWALL
in Linux.
- Add "Tag" to a VM for organizing VMs.
- Get a "key-pair".
 - .pem 2 formats.

- .ppk J1
↓

Once we download it : the private key is deleted by AWS → but they still keep the public key.

∴ User has - PRIVATE key

AWS has - PUBLIC key.

Backend working of AWS Console

User req on console → cloud manager → physical server
↓
create instance
user is given login credentials for that instance

Create EC2 programmatically.

→ "Basically we want to create a

REQUEST Object".

Assignment 1 System Flow.

1. Static site is hosted in **AWS S3**.
2. **API Gateway** is an AWS service which allows us to create & publish APIs.
3. Once created, we can **hook the APIs to the static site**, in such a way that on clicking an "X" button an API 1 gets called.
4. But API Gateway only contains code for redirection → it is not inherently responsible for execution of features.
5. These functions are executed on **AWS Lambda** which is serverless system.

"Serverless Backend" → is a backend where user does not need to create a server → you just specify what software you want to run → when request comes, the machine is instantiated automatically.

6. In Lambda we specify:

- what function?
- What params?
- what condition do we need to bring it on.

7. **LEX** is an AWS service to create chatbot.

It is a "Conversational AI SDK".

8. **AWS SQS** is a message queue. which is required for scheduling messages & execute things **Asynchronously**.

9. **SES** is a notification service.

10. Databases: key-value | docs | — | ...



Required when you need to have
Natural language queries on a data/documents

∴ we will use **ElasticSearch**.

Diving Deep in Architecture

LF0

- captures user input
- can chat with Lex.

LF1

- Lex tells "this is what user wants"
- receives input as raw data.
- Creates a request object based on that data
- pushes that request object to SQS queue.

SQS

- Publisher - Listener arch.
- LF1 = publisher | LF2 = listener.

LF2

