

AWS Services

Purpose built, scalable, fully managed, secure databases

RDS provides relational databases for MySQL and other systems

Reliable, fast, and highly available service

A large variety of encryption and storage technologies

Tools for cryptographic operations

Offers secure user sign up and sign in process

Have the ability to control data access in your applications

Authentication methods available through Cognito user pool

<https://docs.aws.amazon.com/whitepapers/latest/navigating-gdpr-compliance/control-access-to-web-applications-and-mobile-apps.html>

<https://docs.aws.amazon.com/crypto/latest/userguide/awscryp-service-tolevel.html>

https://aws.amazon.com/free/database/?trk=83add82a-8e52-4837-bc73-c323da62d78c&sc_channel=ps&s_kwid=AL!4422!3!610000101516!e!!g!!aws%20database%20services&ef_id=CjwKCAjwhNWZBhB_EiwAPzlhNvLrx9XXFGPsRB0Ub5MIXop4V90WoNNVkJ7jY778J_5kjjJneUF6CzxoCc58QAvD_BwE:G:s&s_kwid=AL!4422!3!610000101516!e!!g!!aws%20database%20services

Security Protocols

Confidentiality: Data is only available to people who are supposed to access it

Integrity: Only authorized individuals can change data values

Availability: Authorized users are always able to access the data

Encryption, firewalls, cryptography, 2FA or MFA are important data security protocols

Regularly check for good security and vulnerabilities

Protection against threats such as phishing, unauthorized access, or data theft

<https://nightfall.ai/5-basic-data-security-protocols-developers-must-know>

Password Encryption

Preventing unauthorized access to user passwords

One way encryption formats: All users are restricted from access, including the administrators

Two way encryption formats: Passwords can be decrypted and can also have authentication methods

<https://www.ibm.com/docs/en/i/7.3?topic=security-password-encryption>

OpenAPI 3.0 Spec

Open API Specification Version 3.0

Interface to RESTful APIs

Doesn't need access to source code, documentation, or traffic inspection to understand a service

Consumers don't need a lot of implementation knowledge

Can be used by documentation tools

Uses template expressions

Media types spread across resources

HTTP status codes provide information about processes

Format names are case sensitive

Can be a single document or in multiple parts

Primitive data types based on ones supported by JSON

Info objects providing information about API

Title and version fields are required

<https://swagger.io/specification/>

APIs

Application Programming Interface

Mechanisms letting software components communicate based on specific protocols and definitions

REST APIs are the most flexible and popular category of APIs

Integration: Integrate new applications with existing ones

Innovation: Respond to new developments and deploy things quickly

Expansion: Meet a large variety of client needs

Ease of maintenance: System changes do not affect API or other components

<https://aws.amazon.com/what-is/api/>

Connecting Front End to Back End

Important for deployment and good user experience

Make a good structure for a client-side rendered application

Implement model view controller architecture

Make a front end and back end and then use CORS

Cross origin resource sharing enabled

Set the correct environmental variables

Make sure everything is working on your local machine

Use a server to deploy front end

Add backend URL to configuration

<https://www.codecademy.com/learn/connecting-front-end-to-back-end>

<https://frontend.turing.edu/lessons/module-4/front-end-back-end-connection.html>

Back End Web App Development

Server side development

Focused on how an application works behind the scenes

Logic and functionality that users don't have to view

Combines servers, applications, and databases

Can be functional or object oriented programming

Statically or dynamically typed languages

Java, PHP, Python are some examples

<https://learntocodewith.me/posts/backend-development/>

Front End Web App Development

Client side development

Establishes the appearance and interactions available

Use HTML/CSS and JavaScript for a dynamic application

Techniques in the field are constantly evolving

Make something that clearly displays relevant information to users

Accommodate for different devices and displays

Web browser retrieves and presents information

<https://frontendmasters.com/guides/front-end-handbook/2018/what-is-a-FD.html>