Software Engineering
Cloud Computing and Cloud Software Architecture

Min-Yuh Day
Associate Professor
Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2020-11-03
課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)
1 2020/09/15 軟體工程概論 (Introduction to Software Engineering)
2 2020/09/22 軟體產品與專案管理：軟體產品管理，原型設計
   (Software Products and Project Management: Software product management and prototyping)
3 2020/09/29 敏捷軟體工程：敏捷方法，Scrum，極限程式設計
   (Agile Software Engineering: Agile methods, Scrum, and Extreme Programming)
4 2020/10/06 功能、場景和故事 (Features, Scenarios, and Stories)
5 2020/10/13 軟體架構：架構設計，系統分解，分散式架構
   (Software Architecture: Architectural design, System decomposition, and Distribution architecture)
6 2020/10/20 軟體工程個案研究Ⅰ
   (Case Study on Software Engineering I)
課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)
7 2020/10/27 基於雲的軟體：虛擬化和容器、軟體即服務
(Cloud-Based Software: Virtualization and containers,
   Everything as a service, Software as a service)
8 2020/11/03 雲端運算與雲軟體架構
(Cloud Computing and Cloud Software Architecture)
9 2020/11/10 期中報告 (Midterm Project Report)
10 2020/11/17 微服務架構：RESTful服務、服務部署
(Microservices Architecture, RESTful services,
   Service deployment)
11 2020/11/24 軟體工程產業實務
(Industry Practices of Software Engineering)
12 2020/12/01 安全和隱私 (Security and Privacy)
課程大綱 (Syllabus)

週次 (Week)  日期 (Date)  內容 (Subject/Topics)
13 2020/12/08  軟體工程個案研究 II  
      (Case Study on Software Engineering II)
14 2020/12/15  可靠的程式設計 (Reliable Programming)
15 2020/12/22  測試：功能測試、測試自動化、測試驅動的開發、程式碼審查  
      (Testing: Functional testing, Test automation, Test-driven development, and Code reviews)
16 2020/12/29  DevOps和程式碼管理：程式碼管理和DevOps自動化  
      (DevOps and Code Management: Code management and DevOps automation)
17 2021/01/05  期末報告 I (Final Project Report I)
18 2021/01/12  期末報告 II (Final Project Report I)
Software Engineering and Project Management

- **Analyze**: Requirements definition
- **Design**: System and Software design
- **Build**: Implementation and unit testing
- **Test**: Integration and system testing
- **Deliver**: Operation and maintenance

Project Management
Product management concerns

Product manager

Business needs

Customer experience

Technology constraints

Technical interactions of product managers

Product manager

Product vision management

Product backlog management

User stories and scenarios

Acceptance testing

Customer testing

User interface design

Software Development Life Cycle (SDLC)

The waterfall model

Requirements definition

System and Software design

Implementation and unit testing

Integration and system testing

Operation and maintenance

Plan-based and Agile development

**Plan-based development**

- Requirements engineering
- Requirements specification
- Design and implementation

**Agile development**

- Requirements engineering
- Design and implementation

The Continuum of Life Cycles

Degree of Change

Low

High

Frequency of Delivery

Low

High

Predictive

Incremental

Agile

Iterative

Predictive Life Cycle

Analyze → Design → Build → Test → Deliver

Iterative Life Cycle

- Analyze
- Analyze Design
- Build Test
- Deliver

Prototype → Refine

A Life Cycle of Varying-Sized Increments

### Iteration-Based Agile

<table>
<thead>
<tr>
<th>Requirements Analysis</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Analysis</td>
<td>Design</td>
<td>Build</td>
<td>Test</td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>Design</td>
<td>Build</td>
<td>Test</td>
</tr>
<tr>
<td>Repeat as needed</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Flow-Based Agile

<table>
<thead>
<tr>
<th>Requirements Analysis</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Analysis</td>
<td>Design</td>
<td>Build</td>
<td>Test</td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>Design</td>
<td>Build</td>
<td>Test</td>
</tr>
<tr>
<td>Repeat as needed</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>Design</td>
<td>Build</td>
<td>Test</td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>Design</td>
<td>Build</td>
<td>Test</td>
</tr>
<tr>
<td>Repeat as needed</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From personas to features

1. Personas
   A way of representing users

2. Scenarios
   Natural language descriptions of a user interacting with a software product
   - Are-developed-into
   - Inspire

3. Stories
   Natural language descriptions of something that is needed or wanted by users
   - Define
   - Inspire

4. Features
   Fragments of product functionality

Multi-tier client-server architecture

Client 1 → Web Server → Application Server → Database Server

Client 2 → Web Server

Client 3 → Web Server

Client ... → Web Server

Service-oriented Architecture

Everything as a service

1. **Infrastructure as a service (IaaS)**
   - Cloud data center
   - Computing
   - Virtualization
   - Network
   - Storage
   - Monitoring
   - Cloud management

2. **Platform as a service (PaaS)**
   - Software as a service (SaaS)
   - Software development
   - Database
   - Logistics management

3. **Software as a service (SaaS)**
   - Photo editing

Software as a service

Outline

• Cloud Computing and Cloud Software Architecture
• AWS Certified Cloud Practitioner (CLF-C01)
• AWS Certified Solutions Architect – Associate (SAA-C02)
• Web Application with AWS Core Services
• AWS Serverless Architecture
• Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito
AWS Certifications

Available AWS Certifications

Professional
Two years of comprehensive experience designing, operating, and troubleshooting solutions using the AWS Cloud

Associate
One year of experience solving problems and implementing solutions using the AWS Cloud

Foundational
Six months of fundamental AWS Cloud and industry knowledge

https://aws.amazon.com/certification/
AWS Certifications

Accredited Educator

Cloud Practitioner

Solutions Architect Associate
AWS Certified Cloud Practitioner

- This certification provides individuals in a larger variety of cloud and technology roles with a way to validate their AWS Cloud knowledge and enhance their professional credibility.
- This exam covers four domains, including cloud concepts, security, technology, and billing and pricing.

https://aws.amazon.com/certification/certified-cloud-practitioner/
AWS Certified Solutions Architect – Associate

• This certification validates your ability to effectively demonstrate knowledge of how to architect and deploy secure and robust applications on AWS technologies.

• This exam is for anyone with at least one year of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable and distributed systems on AWS.

AWS Academy and Certifications

• **AWS Academy Cloud Foundations (ACF)**
  – **AWS Certified Cloud Practitioner (CLF-C01)**

• **AWS Academy Cloud Architecting (ACA)**
  – **AWS Certified Solutions Architect – Associate (SAA-C02)**

[https://aws.amazon.com/training/awsacademy/](https://aws.amazon.com/training/awsacademy/)
# AWS Certified Cloud Practitioner (CLF-C01)

<table>
<thead>
<tr>
<th>Domain</th>
<th>% of Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain 1: Cloud Concepts</td>
<td>26%</td>
</tr>
<tr>
<td>Domain 2: Security and Compliance</td>
<td>25%</td>
</tr>
<tr>
<td>Domain 3: Technology</td>
<td>33%</td>
</tr>
<tr>
<td>Domain 4: Billing and Pricing</td>
<td>16%</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

## AWS Certified Solutions Architect – Associate (SAA-C02)

<table>
<thead>
<tr>
<th>Domain</th>
<th>% of Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain 1: Design <strong>Resilient</strong> Architectures</td>
<td>30%</td>
</tr>
<tr>
<td>Domain 2: Design <strong>High-Performing</strong> Architectures</td>
<td>28%</td>
</tr>
<tr>
<td>Domain 3: Specify <strong>Secure</strong> Applications and Architectures</td>
<td>24%</td>
</tr>
<tr>
<td>Domain 4: Design <strong>Cost-Optimized</strong> Architectures</td>
<td>18%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100%</td>
</tr>
</tbody>
</table>

AWS Certified Cloud Practitioner (CLF-C01)

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/
• **Domain 1: Cloud Concepts**
  – 1.1 Define the AWS Cloud and its value proposition
  – 1.2 Identify aspects of AWS Cloud economics
  – 1.3 List the different cloud architecture design principles

Domain 2: **Security and Compliance**

- 2.1 Define the AWS shared responsibility model
- 2.2 Define AWS Cloud security and compliance concepts
- 2.3 Identify AWS access management capabilities
- 2.4 Identify resources for security support

AWS Certified Cloud Practitioner (CLF-C01)

• Domain 3: **Technology**
  – 3.1 Define methods of deploying and operating in the AWS Cloud
  – 3.2 Define the AWS global infrastructure
  – 3.3 Identify the core AWS services
  – 3.4 Identify resources for technology support

AWS Certified Cloud Practitioner (CLF-C01)

• Domain 4: **Billing and Pricing**
  – 4.1 Compare and contrast the various pricing models for AWS
  – 4.2 Recognize the various account structures in relation to AWS billing and pricing
  – 4.3 Identify resources available for billing support

AWS Certified Solutions Architect – Associate (SAA-C02)

Source: https://aws.amazon.com/certification/certified-solutions-architect-associate
AWS Certified Solutions Architect – Associate (SAA-C02)

• **Domain 1: Design Resilient Architectures**
  
  – 1.1 Design a multi-tier architecture solution
  
  – 1.2 Design highly available and/or fault-tolerant architectures
  
  – 1.3 Design decoupling mechanisms using AWS services
  
  – 1.4 Choose appropriate resilient storage

• Domain 2: Design **High-Performing Architectures**
  
  – 2.1 Identify elastic and scalable compute solutions for a workload
  
  – 2.2 Select high-performing and scalable storage solutions for a workload
  
  – 2.3 Select high-performing networking solutions for a workload
  
  – 2.4 Choose high-performing database solutions for a workload
• Domain 3: Design **Secure** Applications and Architectures
  – 3.1 Design secure access to AWS resources
  – 3.2 Design secure application tiers
  – 3.3 Select appropriate data security options
• Domain 4: Design **Cost-Optimized Architectures**
  – 4.1 Identify cost-effective storage solutions
  – 4.2 Identify cost-effective compute and database services
  – 4.3 Design cost-optimized network architectures

AWS Products and Services

- Analytics
- Application Integration
- Compute
- Business Applications
- Game Tech
- Migration & Transfer
- Media Services
- Satellite
- Robotics
- AR & VR
- Customer Engagement
- Internet of Things
- Mobile
- Migration & Transfer
- Security, Identity & Compliance
- Networking & Content Delivery
- Machine Learning
- Management & Governance
- Quantum Technologies
- Storage
- Blockchain
- Developer Tools

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
# AWS Compute

## AWS EC2
Virtual servers in the cloud

## Amazon Elastic Container Service
Run and manage docker containers

## AWS Batch
Run batch jobs at any scale

## AWS Lambda
Run code without thinking about servers

## AWS Wavelength
Deliver ultra-low latency applications for 5G devices

## Amazon EC2 Auto Scaling
Scale compute capacity to meet demand

## Amazon Elastic Kubernetes Service
Run managed Kubernetes on AWS

## AWS Elastic Beanstalk
Run and manage web apps

## AWS Outposts
Run AWS infrastructure on-premises

## VMware Cloud on AWS
Build a hybrid cloud without custom hardware

## Amazon Elastic Container Registry
Store and retrieve docker images

## Amazon Lightsail
Launch and manage virtual private servers

## AWS Fargate
Run containers without managing servers or clusters

## AWS Serverless Application Repository
Discover, deploy, and publish serverless applications

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
AWS Database

- **Amazon Aurora**
  High Performance Managed Relational Database

- **Amazon ElastiCache**
  In-memory Caching System

- **Amazon Quantum Ledger Database (QLDB)**
  Fully managed ledger database

- **Amazon Redshift**
  Fast, Simple, Cost-effective Data Warehousing

- **Amazon DynamoDB**
  Managed NoSQL Database

- **Amazon Managed Apache Cassandra Service**
  Managed Cassandra-compatible database

- **Amazon RDS**
  Managed Relational Database Service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB

- **Amazon DocumentDB (with MongoDB compatibility)**
  Fully managed document database

- **Amazon Neptune**
  Fully Managed Graph Database Service

- **Amazon RDS on VMware**
  Automate on-premises database management

- **Amazon Timestream**
  Fully managed time series database

- **AWS Database Migration Service**
  Migrate Databases with Minimal Downtime

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
AWS Storage

Amazon Simple Storage Service (S3)
Scalable Storage in the Cloud

Amazon FSx for Lustre
High-performance file system integrated with S3

AWS Backup
Centralized backup across AWS services

CloudEndure Disaster Recovery
Highly automated disaster recovery

Amazon Elastic Block Store (EBS)
EC2 block storage volumes

Amazon FSx for Windows File Server
Fully managed Windows native file system

AWS Snow Family
Physical devices to migrate data into and out of AWS

Amazon Elastic File System (EFS)
Fully managed file system for EC2

Amazon S3 Glacier
Low-cost Archive Storage in the Cloud

AWS Storage Gateway
Hybrid Storage Integration

Source: https://aws.amazon.com/
# AWS Networking & Content Delivery

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon VPC</td>
<td>Isolated Cloud Resources</td>
</tr>
<tr>
<td>Amazon Route 53</td>
<td>Scalable Domain Name System</td>
</tr>
<tr>
<td>AWS Cloud Map</td>
<td>Application resource registry for microservices</td>
</tr>
<tr>
<td>AWS Transit Gateway</td>
<td>Easily scale VPC and account connections</td>
</tr>
<tr>
<td>Amazon API Gateway</td>
<td>Build, Deploy, and Manage APIs</td>
</tr>
<tr>
<td>AWS PrivateLink</td>
<td>Securely Access Services Hosted on AWS</td>
</tr>
<tr>
<td>AWS Direct Connect</td>
<td>Dedicated Network Connection to AWS</td>
</tr>
<tr>
<td>Elastic Load Balancing</td>
<td>Distribute incoming traffic across multiple targets</td>
</tr>
<tr>
<td>Amazon CloudFront</td>
<td>Global Content Delivery Network</td>
</tr>
<tr>
<td>AWS App Mesh</td>
<td>Monitor and control microservices</td>
</tr>
<tr>
<td>AWS Global Accelerator</td>
<td>Improve application availability and performance</td>
</tr>
</tbody>
</table>

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
### AWS Security, Identity & Compliance

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS Identity &amp; Access Management</strong></td>
<td>Manage User Access and Encryption Keys</td>
</tr>
<tr>
<td><strong>Amazon GuardDuty</strong></td>
<td>Managed Threat Detection Service</td>
</tr>
<tr>
<td><strong>AWS Artifact</strong></td>
<td>On-demand access to AWS compliance reports</td>
</tr>
<tr>
<td><strong>AWS Directory Service</strong></td>
<td>Host and Manage Active Directory</td>
</tr>
<tr>
<td><strong>AWS Resource Access Manager</strong></td>
<td>Simple, secure service to share AWS resources</td>
</tr>
<tr>
<td><strong>AWS Shield</strong></td>
<td>DDoS Protection</td>
</tr>
<tr>
<td><strong>Amazon Cognito</strong></td>
<td>Identity Management for your Apps</td>
</tr>
<tr>
<td><strong>Amazon Inspector</strong></td>
<td>Analyze Application Security</td>
</tr>
<tr>
<td><strong>AWS Certificate Manager</strong></td>
<td>Provision, Manage, and Deploy SSL/TLS Certificates</td>
</tr>
<tr>
<td><strong>AWS Firewall Manager</strong></td>
<td>Central Management of Firewall Rules</td>
</tr>
<tr>
<td><strong>AWS Secrets Manager</strong></td>
<td>Rotate, Manage, and Retrieve Secrets</td>
</tr>
<tr>
<td><strong>AWS Single Sign-On</strong></td>
<td>Cloud Single Sign-On (SSO) Service</td>
</tr>
<tr>
<td><strong>Amazon Detective</strong></td>
<td>Investigate potential security issues</td>
</tr>
<tr>
<td><strong>Amazon Macie</strong></td>
<td>Discover, Classify, and Protect your Data</td>
</tr>
<tr>
<td><strong>AWS CloudHSM</strong></td>
<td>Hardware-based Key Storage for Regulatory Compliance</td>
</tr>
<tr>
<td><strong>AWS Key Management Service</strong></td>
<td>Managed Creation and Control of Encryption Keys</td>
</tr>
<tr>
<td><strong>AWS Security Hub</strong></td>
<td>Unified security and compliance center</td>
</tr>
<tr>
<td><strong>AWS WAF</strong></td>
<td>Filter Malicious Web Traffic</td>
</tr>
</tbody>
</table>

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
AWS Cost Management

- **AWS Cost Explorer**: Analyze Your AWS Cost and Usage
- **AWS Budgets**: Set Custom Cost and Usage Budgets
- **AWS Cost and Usage Report**: Access Comprehensive Cost and Usage Information
- **Reserved Instance Reporting**: Dive Deeper into Your Reserved Instances (RIs)
- **Savings Plans**: Save up to 72% on compute usage with flexible pricing

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
AWS Services

• Amazon **EC2**
  – Virtual servers in the cloud

• Amazon **Simple Storage Service (S3)**
  – Scalable storage in the cloud

• Amazon **Aurora**
  – High performance managed relational database

• Amazon **DynamoDB**
  – Managed NoSQL database

• Amazon **RDS**
  – Managed relational database service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB

Source: [https://aws.amazon.com/](https://aws.amazon.com/)
AWS Services

• AWS Lambda
  – Run code without thinking about servers

• AWS Elastic Beanstalk
  – Run and manage web apps

• Amazon VPC
  – Isolated cloud resources

• Amazon Lightsail
  – Launch and manage virtual private servers

• Amazon SageMaker
  – Build, train, and deploy machine learning models at scale

Source: https://aws.amazon.com/
Web Application with AWS Core Services
Cloud Software Architecture

fb.com on AWS

Cloud Software Architecture

Cloud Software Architecture

Cloud Software Architecture

AWS Application Services

Cloud Software Architecture

AWS Security Services

Cloud Software Architecture

AWS Development and DevOps Services

AWS
Serverless
Architecture
AWS Serverless Airline Booking

Source: https://github.com/aws-samples/aws-serverless-airline-booking
AWS Serverless Airline Booking Stack

UI/UX
- Quasar framework
- Vue.js
- AWS Amplify
- stripe Elements

Data/Lang
- Amazon DynamoDB
- Python
- Typescript
- JavaScript

API/Auth
- AWS AppSync
- Amazon API Gateway
- Amazon Cognito

Messaging
- Amazon SNS
- AWS Step Functions

Source: https://github.com/aws-samples/aws-serverless-airline-booking
AWS Serverless Airline Booking

High level infrastructure architecture

Source: https://github.com/aws-samples/aws-serverless-airline-booking
AWS Serverless Architecture
AWS Operational Responsibility Models

Source: Heitor Lessa (2019), How to build a full stack serverless airline ticketing web app. [https://www.youtube.com/watch?v=MyoOeHTp2pg](https://www.youtube.com/watch?v=MyoOeHTp2pg)
Build a Serverless Web Application
Build a Serverless Web Application

Overview

In this tutorial, you'll create a simple serverless web application that enables users to request unicorn rides from the Wild Rydes fleet. The application will present users with an HTML based user interface for indicating the location where they would like to be picked up and will interface on the backend with a RESTful web service to submit the request and dispatch a nearby unicorn. The application will also provide facilities for users to register with the service and log in before requesting rides.

Application Architecture

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application
with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application
with Amazon S3, AWS Lambda, Amazon API Gateway,
Amazon DynamoDB, and Amazon Cognito

1. HTML, CSS, JavaScript, etc.

2. Authentication

3. User Pool

4. Dynamic API calls over HTTP

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

1. Static Web Hosting
Amazon S3 hosts static web resources including HTML, CSS, JavaScript, and image files which are loaded in the user's browser.

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application
with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

User Management

Amazon Cognito provides user management and authentication functions to secure the backend API.

Dynamic API calls over HTTP

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application
with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

**Serverless Backend**

Amazon DynamoDB provides a persistence layer where data can be stored by the API's Lambda function.

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application
with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

HTML, CSS, JavaScript, etc.

Amazon S3

Amazon Cognito

User Pool

Dynamic API calls over HTTP

AWS Lambda

Amazon DynamoDB

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

**RESTful API**

JavaScript executed in the browser sends and receives data from a public backend API built using **Lambda** and **API Gateway**.
Build a Serverless Web Application
with Amazon S3, AWS Lambda, Amazon API Gateway,
Amazon DynamoDB, and Amazon Cognito

5 Terminate resources
Resource Cleanup
You will terminate an Amazon S3 bucket, an Amazon Cognito User Pool, an AWS Lambda function, an IAM role, a DynamoDB table, a REST API, and a CloudWatch Log.
It is a best practice to delete resources you are no longer using to avoid unwanted charges.

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
Summary

• Cloud Computing and Cloud Software Architecture
• AWS Certified Cloud Practitioner (CLF-C01)
• AWS Certified Solutions Architect – Associate (SAA-C02)
• Web Application with AWS Core Services
• AWS Serverless Architecture
• Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito
References

• Titus Winters, Tom Manshreck, and Hyrum Wright (2020), Software Engineering at Google: Lessons Learned from Programming Over Time, O'Reilly Media.
References


• **AWS Cloud Practitioner Essentials (Second Edition)**
  – [https://aws.amazon.com/training/course-descriptions/cloud-practitioner-essentials/](https://aws.amazon.com/training/course-descriptions/cloud-practitioner-essentials/)

• **AWS Certified Cloud Practitioner**

• **AWS Certified Solutions Architect – Associate**

• **AWS Academy Cloud Foundations (AWS ACF)**, AWS Academy
• **AWS Academy Cloud Architecting (AWS ACA)**, AWS Academy