

CSCI 5411 Adv. Cloud Architecting – 2025 Summer
Final Term Project
Architecting and Implementing Applications on AWS

Table of Contents

Overview	2
Scope	2
Project Distribution Overview (component weights in the project).....	3
Requirements of AWS services and well-architected framework	4
Evaluation	5
Submission	6
FAQ	7
Appendix: AWS Service Categories	8

Overview

The objective of this project is to test your understanding of cloud architecting concepts and technologies by having you design and implement an application on Amazon Web Services (AWS). You will be required to choose from various AWS service categories, justify your choices, and demonstrate your understanding of the AWS Well-Architected Framework principles and best practices.

The project encompasses **two components**:

- A compulsory application design and implementation.
- A compulsory one-on-one meeting to demonstrating your projects.

Scope

1. **Application Development:** You will have to choose a domain to design the cloud architecture and implement the application satisfying the requirements of the project. The requirements are described in later sections.

The domain of the application is open, and I suggest you do not choose AI/GenAI for this project as you have practiced that in the mid-term project. However, it is fine that you still choose AI/GenAI applications. This must be a new project, and you are not allowed to use your previous projects.

Example Domains:

- Healthcare - Patient record management, remote monitoring.
 - Logistics & Supply Chain - Shipment tracking system.
 - Smart City/IoT - Real-time traffic monitoring, waste collection management.
 - Finance - Expense tracking, fraud detection platform.
 - Education - Cloud-based learning management system.
 - E-Commerce - Scalable online storefront with cart and payment systems.
 - Energy - Renewable energy monitoring dashboard.
 - Agriculture - Farm sensor data platform (IoT + data processing).
 - Port management - Operation automation, traffic prediction.
2. **Infrastructure as Code (IaC):** Writing Infrastructure as Code (IaC) is mandatory for this assignment. You must use your preferred IaC tools to deploy your infrastructure and application. Aim to deploy everything in a single attempt. If you encounter issues and cannot deploy everything in one go, provide a detailed explanation.
 3. **AWS Academy Learner Lab:** You will use the AWS Academy Learner Lab, which has been specifically set up for this course, to implement your architecture. You have been allocated \$50 in AWS Academy credit for this project. Manage this credit carefully to avoid exceeding your budget.

If you opt to employ your personal AWS account, that's perfectly acceptable; however, I urge you to exercise utmost caution regarding your incurred costs. Please note that AWS Academy Learner Lab have a lot of restrictions on their services. You will have to understand the restrictions before architecting your application. You can find all supported services on the right side of the interface after you launch the learner lab.

Project Distribution Overview (component weights in the project)

Time management is essential in a professional environment. However, many students tend to start their projects at the last minute. To address this, we've implemented milestones, like industry sprints, to help pace the project. Students are encouraged to start early, especially given the compressed term.

Below is the overall weight distribution of all the components:

Weight Distribution		
Project Report		One-on-One Meeting
70%		30%
Milestone 1	Milestone 2	
30%	70%	

Milestone 1: Exploration, Architecture and Design (30% of the project report)

- **Due Date:** 11:59pm, June 8, 2025
- **Deliverables:**
 - Application domain selection and justification. Problem statement and business case for selected application.
 - Detailed functional requirements specification.
 - Non-functional requirements analysis. Analysis of availability, reliability, performance, disaster recovery, cost, security and scalability must be included. Note that the analysis of these non-functional requirements will help you make your project meet the six pillars of the AWS well-architected framework.
 - Initial architecture design with diagram. You must provide a diagram for your architecture. The design and architecture can be slightly improved and modified as you develop your project in the second milestone.
 - Besides architecture design, you also need to provide a data sequence diagram (also called sequence diagram or UML data sequence diagram) to show how data and messages flow between different components, objects, or systems over time.
Note: You can use draw.io to draw the application architecture and data sequence diagram. But you are free to use other tools. You must explain your architecture and data sequence diagram. You are welcome to include any additional diagrams that help clarify and explain your project.
 - AWS services and tech stack. Briefly describe the programming languages, AWS services and tools, as well as any additional technologies you plan to use for implementing your project. You don't need to explain everything in detail at this stage, as you will provide a more comprehensive explanation in the second milestone.
 - Identification of potential architectural challenges.

Milestone 2: Implementation and Documentation (70% of project report)

- **Due Date:** 11:59pm, June 22

- **Deliverables:**
 - Final report on top of the report of milestone 1.
 - Improved architectural diagram and data sequence diagram. There shouldn't be huge changes to your architecture design and data sequence diagram. You keep your original diagrams if you do not need to change them.
 - Detailed explanation of the implementation of the designed project:
 - Service selection and configuration details and screenshots of proper AWS resource configuration and management.
 - Implementation of appropriate monitoring and logging solutions
 - Demonstration and explanation with screenshots showing how the functional and non-functional requirements are met.
 - Security measures at all layers.
 - Cost analysis and optimization strategies.
 - Lessons learned and future improvements.

Requirements of AWS services and well-architected framework

1. Your cloud architecture should use a minimum of **6 categories** from the following AWS service categories. The services contained in each category is listed in the Appendix. If you will be using services that are not listed in the Appendix, feel free to confirm with your instructor.
 - Compute
 - Storage
 - Database
 - Networking & Content Delivery
 - Application Integration
 - Management and Governance
 - Analytics
 - Machine Learning
 - Internet of Things (IoT)
 - Developer Tools
 - Security, Identity, and Compliance
 - End User Computing
 - Media
2. When designing the architecture for the chosen application, it is imperative to adhere to the principles and best practices of all the **six pillars** within the AWS Well-Architected Framework. Due to the strict restrictions on IAM in the AWS Academy Learner Lab, you can skip IAM in your architecture implementation. However, you are expected to explain how you would handle this pillar in your architecture. You need to explain how your design and implementation meet the principles and best practices of the six pillars of the well-architected framework.

Evaluation

The total marks of the project is **45**. Your project will be evaluated based on the following criteria.

- **Project Report (30 marks)**

1. Hosting of the application on AWS (**5 marks**).
 - a. Application runs without any problem on the implemented architecture. (5 marks)
 - b. Application has partial problems running on the implemented architecture. (1-4 marks)
 - c. Application doesn't run. (0 marks)
2. Justification of the chosen AWS services (**10 marks**).
 - a. Provide clear and detailed explanations of the AWS services used, ensuring that each selected service aligns with the requirements specified in point 1 of the 'Requirements of AWS Services and Well-Architected Framework' section. (10 marks)
 - b. The explanations of the services may lack clarity or contain partial inaccuracies, yet they remain comprehensible. And/or the explanations of the services do not adhere to the point 1 of the section of 'Requirements of AWS services and well-architected framework'. (1-9 marks)
 - c. The explanations provided for the services are neither accurate nor clear, and they do not align with the stipulated service categories' requirements. (0 marks)

Note: If your explanation of the AWS services used appears to be copied from AI tools or is too generic and not clearly linked to your specific application design and development, you may lose most of the marks for this evaluation criterion.
3. Implementation of the AWS Well-Architected Framework principles and best practices (**10 marks**).
 - a. Demonstrate accurate application of the AWS Well-Architected Framework by incorporating principles and best practices for the six pillars into the architectural design. The execution of the architecture on AWS aligns with the established design. (10 marks)
 - b. Partially demonstrate the application of the AWS Well-Architected Framework principles and best practices into the architectural design. And/or the execution of the architecture on AWS partially align with the established design. (1-9 marks)
 - c. No application of the AWS Well-Architected Framework principles and best practices into the architectural design. The execution of the architecture on AWS mismatches the established design. (0 marks)

Note: If your explanation of how your solution aligns with the AWS Well-Architected Framework appears to be copied from AI tools or AWS's official documentation, or is too generic and not clearly tailored to your specific application design and development, you may lose most of the marks for this evaluation criterion.

4. Infrastructure of Code (**5 marks**)

- a. The IaC implementation is thoroughly explained with screenshots, and it successfully launches all resources and the application. If the deployment cannot be completed in a single attempt, clear and reasonable explanations are provided. (5 marks)
- b. The IaC implementation lacks clarity, making it difficult for the marker to fully understand. If the deployment cannot be completed in a single attempt, the explanations are incomplete, or the reasoning is unconvincing. (0-4 marks)

- **One-on-One Meeting (15 marks)**

Note 1: At the beginning of the video, please ensure that your student ID card is visible in front of the camera and keep the camera active throughout the entire meeting. Failure to comply with this will result in a score of **ZERO** of the meeting.

Note 2: The meeting is approximately 15 minutes long. Students are expected to use around 7-8 minutes to present their design and discuss the architecture's implementation. The markers will ask questions during or after the presentation. Evaluation is based on the quality of the presentation and the accuracy and thoroughness of the responses to the questions. If a student struggles to answer questions adequately or their responses fail to clarify their design and implementation, the markers will report this to the instructor for further investigation.

Suspected cheating can lead to severe consequences, including a grade of **ZERO**.

1. **Presentation (5 marks)**

- a. The presentation is thoughtfully planned, follows a logical structure, and communicates the information clearly. The presentation seamlessly corresponds with the content of the report. (5 marks)
- b. The presentation is not well prepared such that it is hard for the marker to completely understand the project. The presentation partially matches the content of the report. (0-4 marks)

If the presentation is unrelated to the report, the instructor will contact you to reassess your project.

2. **Question Answering (10 marks)**

- a. All responses are clear, accurate, and consistent with the presentation and the report. (10 marks)
- b. The responses lack clarity, contain inaccuracies, or contradict the content of the presentation and report. (0-9 marks)

If the student is unable to answer all questions; most responses are incorrect, or they are unrelated to the content of the presentation and report, the instructor will contact you to reassess your project.

Please note, this project is intended to test your cloud architecting skills, so focus on the architecture, not the application's codebase. What we would like to see are good architecture design and working application, but not the quality of your code.

Submission

You are only required to submit the project report on Brightspace; there is no need to submit

the source code. The evaluation will be based on both your project report and the one-on-one meeting.

FAQ

1. How long should be the report?

There isn't a required length of the report. But if you provide an explanation of your architecture and service choice selections, your report cannot be just two pages. The length of your report will be reasonable if you try your best to provide a clear explanation.

2. Can I submit my report multiple times?

Yes, you can. If you have submitted your report, but later found that you need to change it, you can just upload a new file. Only the most recent upload is kept on Brightspace.

3. Will I lose points if I do not include architecture and data interaction sequence diagrams?

Architecture diagrams and data interaction sequence diagrams are expected deliverables of this assignment, and you will lose points if you do not submit these deliverables.

Appendix: AWS Service Categories

Compute

- Amazon EC2
- Amazon EC2 Auto Scaling
- Amazon EC2 Image Builder
- Amazon Lightsail
- AWS App Runner
- AWS Batch
- AWS Elastic Beanstalk
- AWS Fargate
- AWS Lambda
- AWS Serverless Application Repository
- AWS Outposts
- AWS Wavelength
- Amazon Elastic Container Registry
- Amazon Elastic Container Service
- Amazon Elastic Kubernetes Service

Storage

- AWS Backup
- Amazon Elastic Block Store
- AWS Elastic Disaster Recovery
- Amazon Elastic File System
- Amazon File Cache
- Amazon FSx for Lustre
- Amazon FSx for NetApp ONTAP
- Amazon FSx for OpenZFS
- Amazon FSx for Windows File Server
- Amazon Simple Storage Service (S3)
- AWS Storage Gateway

Networking and Content Delivery

- Amazon API Gateway
- Amazon CloudFront
- Amazon Route 53
- AWS Verified Access
- Amazon VPC
- Amazon VPC Lattice
- AWS App Mesh
- AWS Cloud Map
- AWS Direct Connect
- AWS Global Accelerator
- AWS PrivateLink

- AWS Private 5G
- AWS Transit Gateway
- AWS VPN
- Elastic Load Balancing

Database

- Compare AWS database services
- Amazon Aurora
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Keyspaces (for Apache Cassandra)
- Amazon MemoryDB for Redis
- Amazon Neptune
- Amazon Relational Database Service
- Amazon RDS for Db2
- Amazon RDS on VMware
- Amazon Quantum Ledger Database (Amazon QLDB)
- Amazon Timestream
- Amazon DocumentDB (with MongoDB compatibility)
- Amazon Lightsail managed databases

Application integration

- AWS Step Functions
- Amazon AppFlow
- AWS B2B Data Interchange
- Amazon EventBridge
- Amazon Managed Workflows for Apache Airflow (MWAA)
- Amazon MQ
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Simple Workflow Service
- AWS AppSync
- AWS AppFabric
- Amazon Chime
- Amazon Pinpoint
- Amazon SES

Management and Governance

- AWS Auto Scaling
- AWS Chatbot
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- AWS Compute Optimizer
- AWS Console Mobile Application

- AWS Control Tower
- AWS Config
- AWS Health Dashboard
- AWS Launch Wizard
- AWS License Manager
- Amazon Managed Grafana
- Amazon Managed Service for Prometheus
- AWS Organizations
- AWS OpsWorks
- AWS Proton
- Service Catalog
- AWS Systems Manager
- AWS Trusted Advisor

Analytics

- Amazon Athena
- Amazon CloudSearch
- Amazon DataZone
- Amazon EMR
- Amazon FinSpace
- Amazon Kinesis
- Amazon Data Firehose
- Amazon Managed Service for Apache Flink
- Amazon Kinesis Data Streams
- Amazon Kinesis Video Streams
- Amazon OpenSearch Service
- Amazon OpenSearch Serverless
- Amazon Redshift
- Amazon Redshift Serverless
- Amazon QuickSight
- AWS Clean Rooms
- AWS Data Exchange
- AWS Data Pipeline
- AWS Entity Resolution
- AWS Glue
- AWS Lake Formation
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Machine Learning

- Amazon Augmented AI
- Amazon Bedrock
- Amazon CodeGuru
- Amazon Comprehend
- Amazon DevOps Guru
- Amazon Forecast

- Amazon Fraud Detector
- Amazon Kendra
- Amazon Lex
- Amazon Lookout for Metrics
- Amazon Lookout for Vision
- Amazon Monitron
- Amazon PartyRock
- Amazon Personalize
- Amazon Polly
- Amazon Rekognition
- Amazon SageMaker
- Amazon Textract
- Amazon Transcribe
- Amazon Translate
- AWS DeepComposer
- AWS DeepRacer
- AWS HealthLake
- AWS HealthScribe
- AWS Panorama

Internet of things (IoT)

- AWS IoT 1-Click
- AWS IoT Analytics
- AWS IoT Button
- AWS IoT Core
- AWS IoT Device Defender
- AWS IoT Device Management
- AWS IoT Events
- AWS IoT ExpressLink
- AWS IoT FleetWise
- AWS IoT Greengrass
- AWS IoT SiteWise
- AWS IoT TwinMaker

Developer Tools

- AWS Application Composer
- AWS Cloud9
- AWS CloudShell
- AWS CodeArtifact
- AWS CodeBuild
- Amazon CodeCatalyst
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline
- Amazon Corretto
- AWS Fault Injection Service

- AWS X-Ray
- AWS Amplify

Security, identity, and compliance

- Amazon Cognito
- Amazon Detective
- Amazon GuardDuty
- Amazon Inspector
- Amazon Macie
- Amazon Security Lake
- Amazon Verified Permissions
- AWS Artifact
- AWS Audit Manager
- AWS Certificate Manager
- AWS CloudHSM
- AWS Directory Service
- AWS Firewall Manager
- AWS Identity and Access Management
- AWS Key Management Service
- AWS Network Firewall
- AWS Resource Access Manager
- AWS Secrets Manager
- AWS Security Hub
- AWS Shield
- AWS IAM Identity Center
- AWS WAF

End User Computing

- Amazon AppStream 2.0
- Amazon WorkSpaces
- Amazon WorkSpaces Core
- Amazon WorkSpaces Thin Client
- Amazon Workspaces Web

Media

- Amazon Elastic Transcoder
- Amazon Interactive Video Service
- Amazon Nimble Studio
- AWS Elemental Appliances and Software
- AWS Elemental MediaConnect
- AWS Elemental MediaConvert
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaStore
- AWS Elemental MediaTailor