

CSCI 5411 Adv. Cloud Architecting – 2024 Fall
Term Project
Architecting Applications on AWS

Table of Contents

Objective	2
Scope.....	2
Tasks.....	2
Deliverables.....	3
Evaluation.....	3
Submission	5
FAQ	5
Appendix: AWS Service Categories.....	6

Objective

The objective of this project is to test your understanding of cloud architecting concepts and technologies by having you host 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.

Scope

1. **Application Development:** Developing an application is **NOT** required for this assignment. However, if you choose to do so, please note that no additional points will be awarded for this effort. You may modify the chosen application, but ensure it remains fully functional and retains all original features.
2. **Using Open-Source Application Code:** You are expected to use an existing open-source application written in a language of your choice. The repository should have at least 100 commits and 50 stars, indicating its stability and popularity. You can use the GitHub search tool (<https://seart-ghs.si.usi.ch/>) or other repository search tools to find suitable applications. If you have any questions about selecting an application, please reach out to our TA, Zainuddin Saiyed.
3. **Application Selection Process:** Applications are assigned on a first-come, first-served basis. Once an application is selected by a student, it cannot be chosen by another student. Make sure to enter both the name and the link of your chosen application into the spreadsheet located in our Teams channel: Term Project. This step is essential to avoid duplication of application choices among students.
4. **Infrastructure as Code (IaC):** Writing Infrastructure as Code (IaC) is mandatory for this assignment. You must use either CloudFormation or Terraform 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.
5. **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.
Although using your personal AWS account is acceptable, be cautious about your spending. Setting up an AWS Budget is recommended to receive alerts if your expenses exceed your set threshold. Note that the AWS Academy Learner Lab has several service restrictions. Familiarize yourself with these restrictions before designing your application. You can find the document "AWS Academy Learner Lab Supported Services" on Brightspace to understand the supported services.

Tasks

1. Select the open-source application. Explain why you choose this application.
2. Design the cloud architecture to host the application on AWS. 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.
 - 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

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.

3. Implement the architecture on AWS.

Deliverables

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.

- **Requirement of the project Report**

Provide a detailed report outlining and justifying your choices. Explain why each service was selected, focusing on how it best suits your application in terms of cost, performance, security, and scalability.

This report should also explain how your architecture adheres architecting principles and best practices. Inclusion of an architecture diagram (or multiple diagrams) and data interaction sequence diagram is expected in this report.

The report must present the hosted application's public URL or other accessible evidence of the running application on AWS.

Evaluation

The total marks of the term 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 accurate and lucid explanations of the utilized services. Ensure that the chosen services fulfill the requirements outlined in the Tasks section and align with the AWS service categories. (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 requirements of the AWS service categories as elucidated in the Tasks section. (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)
- 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)
- 4. Infrastructure of Code (**5 marks**)
 - a. The IaC implementation is thoroughly explained, 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 10 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.

Submission

The project report should be submitted on the Brightspace by the due date. The one-on-one meeting will be scheduled by the marker.

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 assignment multiple times?

Yes, you can. If you have submitted your assignment, but later found that you need to change it, you can just upload a new file. The marker will only be looking at the most recent one as the official submission.

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.

4. Can I rearchitect monolith application into microservices?

This is an architectural decision which you will make as an architect, you are free to explore various architecture choices at hand and choose the best fit based on application, complexity, knowledge at hand, available credits, timelines for delivery.

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