

## SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in DS, IT, CSF

**Cloud Architecture and Technologies (CAT)**

**April 2023 Semester**

**Assignment 2**

**35% of CAT Module – (Team: 70%, Individual 30%)**

**3 July – 6 Aug 2023 (Weeks 12 - 16)**

**Deadline for submission:**

**SOFTCOPY:** Submit in Brightspace by 06 Aug 2023, 23:59

**Penalty for late submission:**

* **10%** of the marks will be deducted for each day (inclusive of Saturdays, Sundays and public holidays) after the deadline for softcopy submission.
* **NO submission will be accepted after 06 August 2023, 23:59 pm.**

1. **Objectives**

The objective of this assignment is to assess students’ understanding of Cloud Services and practical skills to implement a given business’ requirement in AWS.

The students will be assessed on their ability to:

1. Deploy EC2 instances to create an elastic and resilient server infrastructure using auto scaling and elastic load balancers
2. Deploy S3 object storage for cloud storage requirements
3. Create a database to manage business data
4. Implement Lambda functions to monitor events and trigger notifications
5. Implement Lambda functions to run applications when triggered
6. Implement AWS CloudWatch and AWS Simple Notification Service SNS to monitor AWS resources/application.
7. S**cope and Background**

Assume that your team (consisting of 4-5 members) has been appointed by a large IT service provider organization called CCP Pte Ltd as IT cloud consultants.

A company KVM Pte Ltd has tasked CCP Pte Ltd to implement some requirements.

KVM Pte Ltd has an existing on-premise IT system in their own data center as shown below.

Diagram

Description automatically generated

Existing On-premise IT System

The above system hosts an existing web application (as in Appendix). The application uses **Microsoft Windows .Net Core 2.2 framework** and the published code will be provided for in this assignment. No software programming is required.

KVM Pte Ltd wants your team to implement the following mandatory requirement:

* To design and implement highly available resilient, auto-scaling and secure VPC (virtual private cloud) to host the same existing web application (see Appendix) in AWS Cloud. The servers should be built on general purpose servers. They require at least 2 servers at any one time, but able to scale up to 4 servers in times of high demand. To save cost, the number of servers need to reduce back to 2 when there is low demand.
* The web application must be **secure, resilient, scalable and load balanced.** It should also be accessible from the Internet. The design should be based on AWS well-architected design principles of security, reliability, performance efficiency and cost optimization.

The published web application to be deployed for this requirement is provided for and explained in Appendix.

* To propose and implement cloud monitoring (metrics, alerts & alarms) on the web application and AWS resources/services using AWS CloudWatch. An alarm and notification system (using AWS SNS) should also be set up. Automatic appropriate actions should be employed if alarm condition occurred. Appropriate critical metrics should be selected for monitoring and a CloudWatch dashboard customized and created for monitoring the different AWS resources/services.
* Students can decide on the operating systems (whether Linux or Windows) of the EC2 web servers as both operating systems can support the execution of Windows .Net Core application. Possible web server solutions include Nginx, Apache or Windows IIS server. RDS can be used to store its’ data.
* Based on your allocated team number, use the following VPC network parameters for your AWS resources:

|  |  |  |
| --- | --- | --- |
| **Team X** | **VPC Name** | **VPC CIDR Address** |
| **1** | ***VPC-Team1*** | ***10.1.0.0/16*** |
| **2** | ***VPC-Team2*** | ***10.2.0.0/16*** |
| **3** | ***VPC-Team3*** | ***10.3.0.0/16*** |
| **4** | ***VPC-Team4*** | ***10.4.0.0/16*** |
| **5** | ***VPC-Team5*** | ***10.5.0.0/16*** |

KVM Pte Ltd also wants your team to implement **another** mandatory requirement which is:

* To design and implement a **new** AWS **serverless application architecture in** AWS (serverless function) with the following cloud requirements:

1. Create a NoSQL database that is able to store customer data that are posted to the website. The database should have data fields to capture the customer first name, last name, and email address.
2. Create a HTML forms page to capture the customer data and post it to the NoSQL database using HTTP (API gateway). The web page should also be able to show the current records in the NoSQL database.
3. Create a notification system (via SMS or email) that will notify the administrator when
   * a new customer data is added to the NoSQL database.
   * a file is uploaded to a S3 bucket

* For this part, students are to research on how to deploy a serverless application with the integration of AWS services such as API Gateway ( which acts as a "front door" for applications to access data, business logic, or functionality from your backend service), Lambda and DynamoDB.

Some useful URL links for API Gateway, Lambda and DynamoDB are:

* <https://ordina-jworks.github.io/cloud/2018/10/01/How-to-build-a-Serverless-Application-with-AWS-Lambda-and-DynamoDB.html>
* <https://thewebspark.com/2018/08/22/how-create-a-serverless-website-using-aws-lambda-aws-s3-aws-api-gateway-and-aws-dynamodb-with-node-js-full-tutorial/>
* <https://www.velotio.com/engineering-blog/aws-serverless-application>

An **optional** requirement is to:

* Create a serverless function to create thumbnails for pictures uploaded to the pictures S3 bucket. There is no requirement for the thumbnails to be displayed on the web page.

Note:

* Students may use one or more AWS academy accounts from your team to build your solution to deploy the web application and set up the AWS resources/services. But some services that need to be integrated to work have to be in a single account.

**Deliverables**

* 1. **Assignment Report (20% Team.10% INDIVIDUAL)**

Each team is required to submit a single report of 3000 to 4000 words. The report should include the following:

1. Network diagram of the cloud infrastructure showing the different AWS VPC components.
2. A detailed description of the cloud solution implemented. Detailed step by step procedure on implementations is not required.
3. Problems encountered and resolutions implemented, if any
4. Clear explanations on how the above requirements are implemented and met.
5. Code used for the Lambda functions
6. Testing methods and results showing that your solution is working.
7. Explanation on how your cloud solution incorporated the AWS well-architedted design principles of security, reliability, performance efficiency and cost optimization.
8. Explanation on your cloud monitoring plan for the web application and other AWS resources/services. Include the techniques and principles used in cloud monitoring.
9. Any other relevant screenshots and diagrams that explain your design.

Present your report with suitable section and subsection headings. You should also include a table of contents and appropriate citation and a list of references. Assignment cover page is provided in the Appendix. The organization of the report should be neat, professional, and logical.

**Important Note:**

Students from the same team may be given different grades for the team component if the tutor is so convinced by evidence of widely unequal contribution by the members.

|  |
| --- |
| **Plagiarism Warning:**  If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action may also be taken. Similar action will be taken for the student who allows other student(s) to copy his/her work. |

* 1. **Presentation (20% Individual)**

Each member of the team is required to do a presentation not exceeding 10 minutes. Each member must explain at least one part of the solution implemented done by him/her. Every member of the team should be familiar with the overall solution and proposed plans.

* 1. **demonstration (20% TEAM)**

Every member of the team is required contribute and implement the practical requirements. During demonstration (in class), each member must explain at least one part of the solution implemented done by him/her.

* 1. **Feature implementation (30% TEAM)**

| **Requirements** | **Marks** |
| --- | --- |
| ***A) Ability to deploy scalable, resilient, load-balanced servers*** | ***20*** |
| * Infrastructure is resilient * Infrastructure is scalable * Infrastructure is load-balanced | 5  5  10 |
| ***B) Cloud Monitoring and Alarm Notification System*** | ***20*** |
| ***C) Serverless function to post data to the NoSQL database (using AWS API Gateway, Lambda and DynamoDB)*** | ***30*** |
| * Functioning Lambda function & Testing method used * Form Page to submit data & DynamoDB | 20  10 |
| ***D) Notification System*** | ***30*** |
| * Functioning notification system for NoSQL item creation * Functioning notification system for S3 uploads | 15  15 |
| **Total** | **100** |

**Note: Additional marks will be awarded to successful completion of optional business requirement.**

1. **suggestions**

* Each team should plan properly and allocate some time outside of class time to do the assignment. Work should be equally divided among group members.
* Students are strongly encouraged to complete the practical exercise on Deploy NET app on EC2 and RDS in VPC before attempting to implement the assignment requirement.Some research work on serverless application with API Gateway, Lambda and DynamoDB is also required.
* Sources of information should be specifically acknowledged. Plagiarism will not be tolerated.

**APPENDIX**

Below are some screenshots of the web application (.Net core application) that is to be implemented for the assignment (requirement on page 2). The code and published application will be provided. The application uses .**Net Core version 2.2** runtime which can be downloaded from the Internet from Microsoft website ( https://dotnet.microsoft.com/en-us/download/dotnet/2.2).

Graphical user interface, table

Description automatically generated

Note:

* Students need to edit and change the connection string properly in the **appsettings.json** file(as shown) to ensure the application can communicate properly with the database.

Graphical user interface, text, application, email

Description automatically generated

* As shown in the connection string, the user Id and password should be set correctly and the Trusted\_Connection argument set to false.
* The published web application is in the following subfolder of the unzipped folder:
* **Web Application For CCP Assignment\DataGreenHRApplication\DataGreenHRApplication\bin\Debug\netcoreapp2.2\publish**

Although not necessary, the source code is also provided.

Table

Description automatically generated

* SQL Server script file is also provided to create SQL database and table.

**Note- Suspend Auto Launch Process in Auto Scaler**

When performing Auto Scaling, it is important that students suspend the launch process after they have completed the auto-scaling. This is to ensure that the credit will not be drastically reduced as there may be many EC2 running at the same time.

After adding the suspending the auto launch process, remember to terminate (or stop) existing running EC2 instances (which are launched from autoscaler). Once the EC2 instances are terminated, auto scaler will not auto launch another EC2 instance as the launch process is suspended. Once this suspension is removed , the auto scaler will launch the EC2 again. Tutor will remove the suspension to check if autoscaling is configured correctly.

**GRADING CRITERIA**

This assignment constitutes **35%** of this module. Performance Criteria for grading the assignment is as described below. Marks awarded will be based on implemented work including submitted report as well as student’s degree of understanding of work done as assessed during the demonstration.

Students from the same team may be given different grades for the team component if the tutor is so convinced by evidence of widely unequal contribution by the members.

**Grading criteria for the program is given below:**

***A Grade***

|  |
| --- |
| * Submitted network proposal including network diagram fulfils all the requirements. * Implemented work implements all the requirements successfully. * Implemented work has been tested adequately. * Implemented work demonstrates good design principles. * Demonstrates excellent understanding of the implemented cloud solution. * Ability to answer all questions posed during demonstration correctly. * Ability to perform all practical tasks posed during demonstration correctly. |

***B Grade***

|  |
| --- |
| * Submitted network proposal including network diagram fulfils (80%) most of the requirements. * Implemented work implements most of the requirements (80% of the requirements) successfully. * Implemented work has been tested adequately. * Implemented work demonstrates some good design principles. * Demonstrates sound understanding of the implemented cloud solution. * Ability to answer most of the questions posed during demonstration correctly. * Ability to perform most of practical tasks posed during demonstration correctly. |

***C Grade***

|  |
| --- |
| * Submitted network proposal including network diagram fulfils (70%) most of the requirements. * Implemented work implements most of the requirements (70% of the requirements) successfully. * Some of the implemented work has been tested adequately. * Implemented work demonstrates some good design principles. * Demonstrates adequate understanding of the implemented cloud solution. * Ability to answer some questions posed during demonstration correctly. * Ability to perform some practical tasks posed during demonstration correctly. |

***D Grade***

|  |
| --- |
| * Submitted network proposal including network diagram fulfils (50%) at least half of the requirements. * Implemented work implements some of the requirements (50% of the requirements) successfully. * Some of the implemented work has been tested adequately. * Demonstrates basic understanding of the implemented cloud solution. Ability to answer some questions posed during demonstration correctly. * Ability to perform some practical tasks posed during demonstration correctly. |



## SCHOOL OF INFOCOMM TECHNOLOGY

**Cloud Architecture and Technologies (CAT)**

**Assignment - April 2023 Semester**

**35% of CAT Module – (Team: 70%, Individual 30%)**

**3 July – 6 Aug 2023 (Weeks 12 - 16)**

**Deadline for submission:**

**SOFTCOPY:** Submit in Brightspace by 06 Aug 2023, 23:59

|  |  |  |  |
| --- | --- | --- | --- |
| **Tutorial Group:** |  | **Team Number:** | 1 / 2 / 3 / 4 / 5/ 6 / 7  (Please circle one) |
| **Members** | **Student No.** | **Student Name** | **Grade** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Plagiarism Warning:**  If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action may also be taken. Similar action will be taken for the student who allows other student(s) to copy his/her work. |

**Reminder**

* **Penalty for late submission:** 10% will be deducted for each day (including Sunday and Public holiday).
  + Only 1 report submission for each group is required.