**Table of Content**

[Scenario 2](#_Toc185951639)

[Practical Tasks 3](#_Toc185951640)

[Architectural Framework Design 3](#_Toc185951641)

[The London College’s Requirements 3](#_Toc185951642)

[Components for the proposed framework 4](#_Toc185951643)

[Design of the proposed framework 4](#_Toc185951644)

[Configuring Cloud Computing Platform with AWS Framework 6](#_Toc185951645)

[Table 1: Creating VPC, Subnets and Route Tables. 7](#_Toc185951646)

[Table 2: Creating Internet and NAT gateways. 18](#_Toc185951647)

[Table 3: Creating and launching Instance, Security Group rules. 25](#_Toc185951648)

[Auto-scaling: 34](#_Toc185951649)

[Table 4: Creating Image, Target Group, and Load Balancer. 34](#_Toc185951650)

[Table 5: Creating Auto-Scaling Group. 47](#_Toc185951651)

[Table 6: Testing Auto-Scaling Group. 56](#_Toc185951652)

*This is a part of the assignment of the course '****Cloud Computing'*** *provided by International School of Management and Technology, partnered with University of Sunderland.*

*The scenario and tasks are provided by the college.*

*The work is entirely of the author.*

# Scenario

The London College currently has all of its college services hosted on its on-premises servers. However, the college has decided to migrate its services to the cloud. The college has entered into a contract with Siddhartha Cloud Computing Pvt. Ltd., a cloud Services Company located in Kathmandu, Nepal. As a Cloud Administrator hired by Siddhartha Cloud Computing Pvt. Ltd., your role is to assist in the migration process and ensure a smooth transition of services to the cloud.

As a cloud administrator at Siddhartha Cloud Computing Pvt. Ltd., you are assigned to assist The London College in migrating their on-premises servers to the cloud. Your tasks include analyzing the evolution and fundamental concepts of cloud computing, discussing the reasons why organizations should migrate to a cloud computing solution, justifying the tools chosen for realizing a cloud computing solution, and designing an appropriate architectural cloud computing framework for the given scenario.

Once the appropriate cloud computing framework is designed, your next task is to develop cloud computing solutions using service provider frameworks and open source tools. You will configure a cloud computing platform for The London College using the cloud service provider's framework provided by Siddhartha Cloud Computing Pvt. Ltd. During the development process, you need to discuss the issues and constraints that may arise and how to overcome them. Additionally, you will implement a cloud platform using open source tools to enhance the services provided to The London College.

As part of your responsibilities, you need to submit a proper report to the CEO of Siddhartha Cloud Computing Pvt. Ltd. The report should document the activities involved in hosting The London College website and migrating all of its site data to the AWS Cloud. Specifically, you are required to:

* Build a three-tier AWS VPC (Virtual Private Cloud) from scratch.
* Create NAT Gateways to enable outbound internet connectivity for private subnets.
* Create security groups to control inbound and outbound traffic to and from the resources in the VPC.
* Configure the auto-scaling groups for ec2
* Set up an application load balancer to distribute traffic across multiple instances.
* Implement an auto scaling group to automatically adjust the number of instances based on demand.

## Practical Tasks

* **Design** an architectural framework for cloud computing that aligns with the specific requirements of The London College.
* **Configure a Cloud Computing Platform with AWS Framework:**
* **Implement a Cloud Computing Platform using Open-Source Tools.**

# Architectural Framework Design

## The London College’s Requirements

The London College wants to migrate their on-premises servers to the cloud. They plan to host their website and migrate all of its on-site data to the AWS Cloud. They have also asked for these specific requirements to be met:

• Establishing a three-tier AWS VPC from scratch.

• Forming NAT gateways in order that the isolated subnets may reach the web.

• To construct security groups allowing only certain traffic.

• Forming security groups for controlling inputs, outputs to the VPC resources.

• Establishing of EC2 auto-scaling groups.

• Enabling a Load Balancer distributes traffic among multiple instances.

• Auto-Scaling Group deployment and scaling using demand.

In brief, the objective is to construct a three-layer AWS Virtual Private Cloud (VPC) connected to an Internet Gateway, as well as NAT gateways for the private subnet to connect to the general internet and security groups to regulate inward and outward traffics. In addition, it involves creating EC2 auto-scaling groups that scale up and down instances as per demand and installing a load balancer used to distribute traffic evenly across multiple instances.

## Components for the proposed framework

Being the cloud administrator responsible for handling these tasks for the college, I plan on incorporating the following components in order to meet the given requirements.

|  |  |
| --- | --- |
| * Virtual Private Cloud | * Subnets |
| * Security Group | * Route Tables |
| * Elastic Compute Cloud (EC2) instances | * Internet Gateway |
| * NAT Gateways | * An image |
| * Target Group | * Load Balancer |
| * Launch Template | * Auto Scaling Group |

## Design of the proposed framework

Below I have designed a 3-tier VPC for The London College incorporating all the above stated components in Draw.io, a free-to-use software for making diagrams and charts.

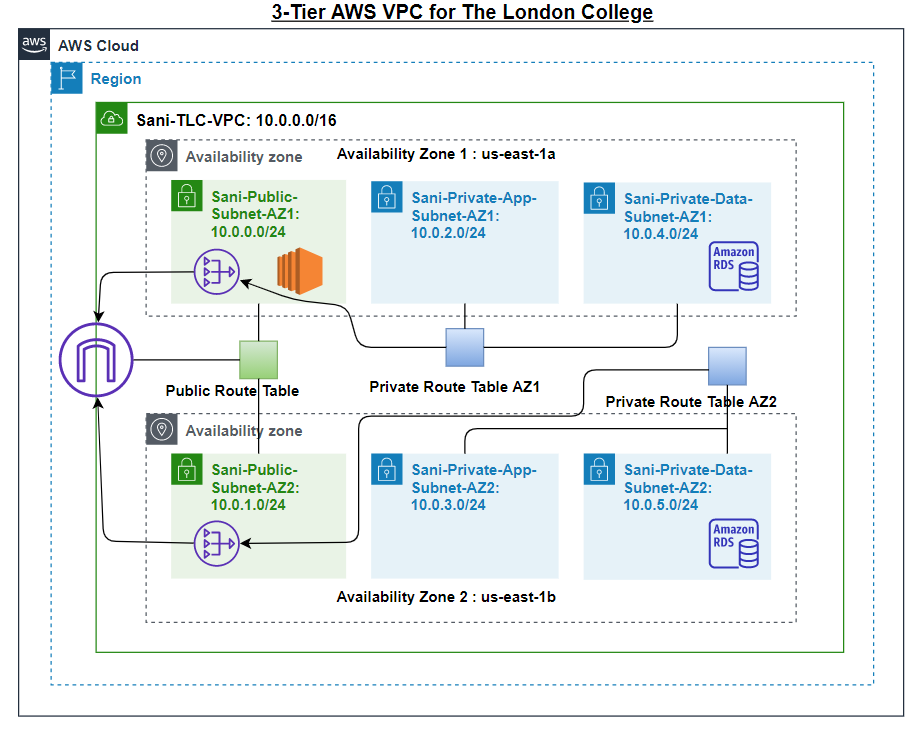


Figure 1: 3-Tier AWS VPC for The London College.

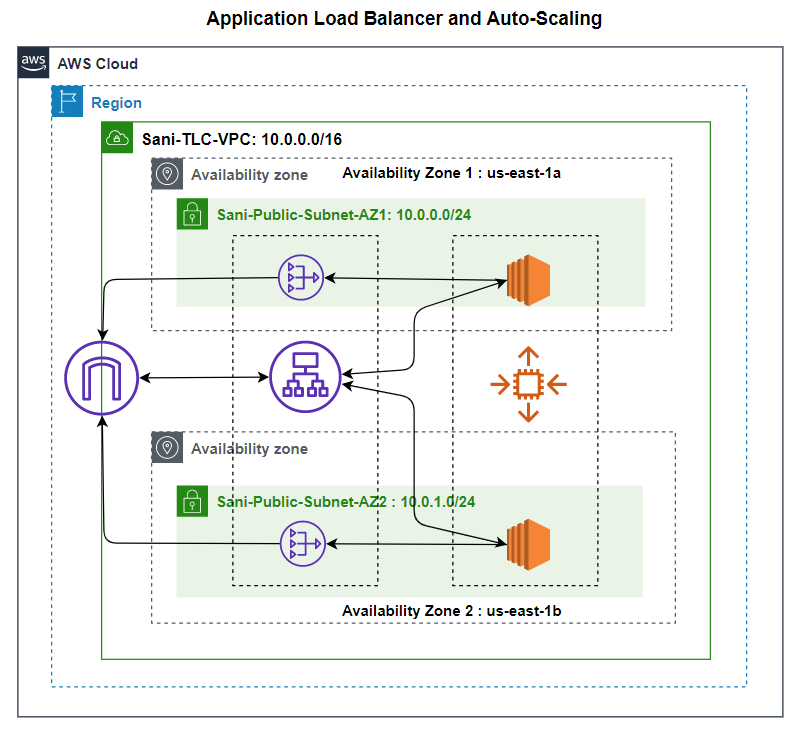


Figure 3: Application Load Balancer and Auto Scaling.

# Configuring Cloud Computing Platform with AWS Framework

I have outlined the thorough procedure for hosting The London College website and seamlessly moving its data to the AWS Cloud in this complete report. Building an efficient three-tier AWS VPC, creating security groups to manage traffic flow, building NAT Gateways to ensure secure outbound internet access in private subnets, Internet gateways for the VPC, setting up auto-scaling groups for EC2 instances, and putting in place an application load balancer to effectively distribute traffic among multiple instances are all illustrations of these strategic actions. The report offers a thorough analysis of every stage by providing an overview of the planning and execution that went into this AWS cloud deployment.

## Table 1: Creating VPC, Subnets and Route Tables.

|  |
| --- |
| Here, we have the Console where we can search for different services provided by the cloud service provider. |
| First, we create a Virtual Private Cloud (VPC) that closely represents a traditional network where we have full control over the networking environment. |
|  |
| Here, we can see the VPC has been created i.e. Sani-TLC-VPC. |
| Next, we create subnets within the VPC. This is the primary way for organizing and dividing resources within the VPC. I will be creating 6 different subnets. Each of the subnets below in 2 Availability Zones (AZ).   1. A public subnet. 2. A private app subnet. 3. A private data subnet.   Different AZs ensures high availability and fault tolerance. |
| 123 |
| sub 3 |
| sub 4 |
| sub 5 |
| sub 6 |
| Here we can see, 6 different subnets with CIDR notation ranging from 10.0.0.0/24 to 10.0.5.0/24 has been created within the same VPC we created earlier. |
| Next, I will be creating route tables. There's already a default route table present but custom route tables help me control the traffic flow. I have created 3: one public route table and two private route tables, one for each availability zone. |
|  |
|  |
| Here, we can see the successful creation of the route tables. |

## Table 2: Creating Internet and NAT gateways.

|  |
| --- |
| Here, I will be cre2ating internet gateway to enable communication between VPC instances and the internet. |
| The VPC needs to be attached to the internet gateway. |
| Next, I will be creating 2 NAT gateways. This enables instances in private subnet to initiate outbound traffic to the internet. |
|  |
| Here we can see the two different NAT gateways. |
| Next, I will be editing subnet associations and routes for the different route tables.  First, we have the public route table. I have associated public subnets in AZ1 and AZ2 to this route and also connected to the Internet gateway. |
|  |
| Second, we have the private route table 1. I have associated private app and data subnets in AZ1 to this route and also connected to the NAT gateway 1. |
|  |
| Lastly, we have the private route table 2. I have associated private app and data subnets in AZ2 to this route and also connected to the NAT gateway 2. |
|  |
| Here in the resource map, we can see everything that has been created in the VPC so far. |

## Table 3: Creating and launching Instance, Security Group rules.

|  |
| --- |
| Now we will be launching a servn instance within the public subnet AZ1 in our previously created VPC named Sani-TLC-Web-Server. |
| I will be creating a key pair for extended security measures. |
|  |
| Here, I have specified the VPC and subnet where I will be launching the web server and also enabled the auto-assigning of public IP address. This is important so people on the internet can access the web server. Here, I have also created a new security group. |
| Here, I have enabled SSH for secure remote connection and HTTP and HTTPS for web connection. |
|  |
| After the creation of the instance, now I will be accessing it and launching the server. |
|  |
| Here is the script that needs to be run in order to install and create the web server. |
| The script is then executed. |
|  |
| Here is the website after the completion of launching. |
|  |

## Auto-scaling:

## Table 4: Creating Image, Target Group, and Load Balancer.

|  |
| --- |
| Now, I am creating an image as it serves a template to launch instances as it ensure consistency and efficiency in scaling applications. |
|  |
| Here, we can see the creation of an image. |
| Next, we will be creating target groups.  The first step is to specify group details. |
|  |
| Here we can see the successful creation of Target Group. |
| Next, I will be selecting the necessary type of load balancer. |
| Here, I will be creating Application load balancer which is important for distributing incoming traffic across multiple targets. This helps in ensuring optimal resource utilization, and availability of application. |
|  |
|  |
|  |
| Here, we can see the creation of Load balancer. |
| Now, I will be creating launch template which provides a standard and scalable approach to define the configuration of instances for auto scaling. |
|  |
|  |
|  |
|  |

## Table 5: Creating Auto-Scaling Group.

|  |
| --- |
| Now, I will be creating auto scaling group from the existing template that we just created earlier. There will be five different steps to be completed in order to create an auto scaling group. This ensures dynamic adjustment of compute resources based on demand, performance and of course maintaining cost efficiency. |
| Step 1: Choose launch template. |
|  |
| Step 2: Choose instance launch options |
|  |
|  |
| Step 3: Configure advanced options |
|  |
|  |
| Step 4: Configure group size and scaling policies. |
|  |
| Step 5: Add notifications. |
| Here, we can see the successful creation of Auto-Scaling group. |

## Table 6: Testing Auto-Scaling Group.

|  |
| --- |
| Now, in order to test our auto-scaling group, let us copy the DNS name and run it in a new tab. |
| Here, we can see the current CPU load to be 0%. |
| To test if our configurations are working properly, we will do a load test. Here, the CPU load is being generated.  cpuload |
| Here, it is under high CPU load.  highload |
| As we can see, multiple instances have been created for dynamic adjustment of compute resources based on demand and optimizing performances. |