Innovation Center for Education

TRUSTED DELIVERY PARTNER for IBM ICE

University of Petroleum and Energy Studies

Internship - Low Level Design

on

Host Dynamic Website on AWS

SUBMITTED BY:

1. ANKUR GUPTA (R110219018,CCVT)
2. ANUPAM KUMARI (R110219022,CCVT)
3. ASTHA KUMARI (R110219027,CCVT)
4. DEVANG TYAGI (R110219041,CCVT)

GUIDED BY
Yogesh Ghorpade

INDUSTRY MENTOR

Table of Contents

1. Introduction

Back in the early days of the internet, websites were purely "static." The content on each page was coded in HTML and did not change unless edited and published by the webmaster (who knew how to code in HTML).

There are still plenty of static websites today, but modern sites are typically built using a database-driven content management system (CMS) like WordPress. Sites built with a database allow pages to be generated dynamically [1]. Each time a visitor accesses a page within a dynamic site, the HTML is "pulled" from the database and sent to the user's web browser, dynamically creating the page.

The benefit of having a Dynamic website is as a marketer you can easily update, create, and publish new content all the time without having to know how to code. This makes it much easier to keep your website fresh and engaging.

Amazon Web Services (AWS) provides a reliable, scalable, secure, and highly performing infrastructure for the most demanding web applications. This infrastructure matches IT costs with customer traffic patterns in real-time. This project describes the hosting websites dynamically using Amazon Web Services that can change the old path of deploying websites through servers and databases.

AWS service maintains all inactions that can start from deploying data in S3 Bucket and creating a computing device that can act as our virtual server and access from anywhere. We use an elastic Load Balancer (ELB) service that can help check the performance of the websites, whether the website is accessing slow or fast and whether the health of the website is also maintained. Virtual Private Cloud (VPC) service that prevents the website and provides security to the website and by that the perfect website hosting will be done by using the services of Amazon Web Services.

This project comprises of demonstration of how one can deploy a dynamic website with AWS by uploading your website content into an S3 bucket and creating an EC2 instance to host a web app on it as in this scenario EC2 acts like a public server for all people from the world can visit this server.

There are three main processes in this proposed system, which are designing a website and uploading/deploying the website into S3 Bucket and connecting the website instance with Elastic load balancer [2]. Then it will provide the website instances with a good storage capacity and maintains traffic load perfect with the best security approach using VPC.

1.1 Scope of the document

The Scope of the document is to develop a documentation with preliminary design (Or low level design) – thus termed "preliminary" – during the Initiation Process and will be reviewed and agreed upon in the Planning Process.

The document Scope Statement identifies the low level deliverables and requirements that must be met in order to deliver a desired end result (A dynamic website hosted on AWS) with the specified features and functions of the project.

1.2 Intended audience

The dynamic website "Digio" provides a virtual platform for "Gym management" hence the target audience are the gym trainers with a basic requirement of laptop or any device with an access to internet .

1.3 System Overview

System Requirements

Recommended Operating Systems.

1. Windows 7 or later.
2. Linux
3. Mac-OS

Recommended Browser

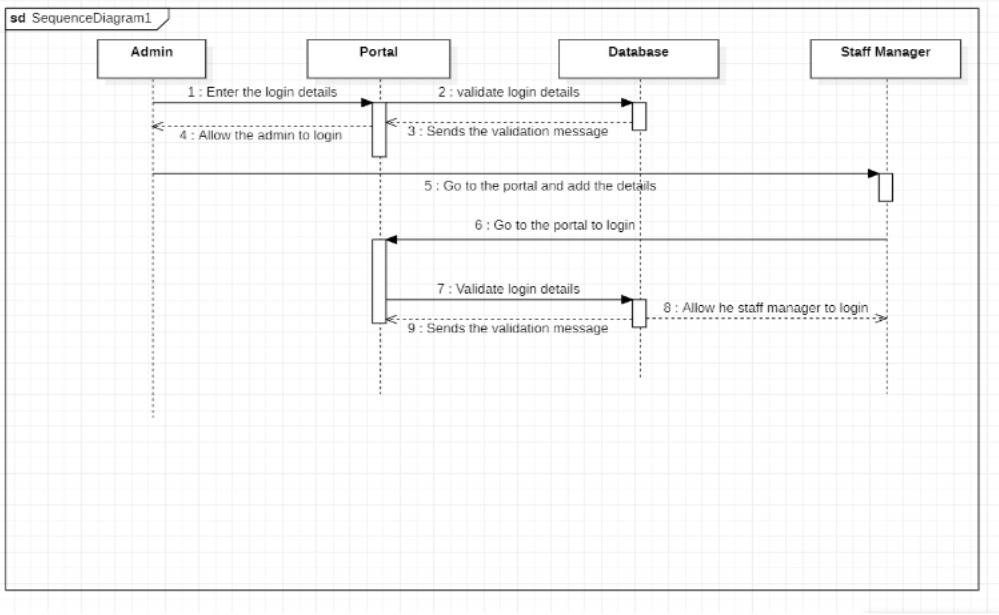
1. Chrome
2. Edge

Hardware Requirements

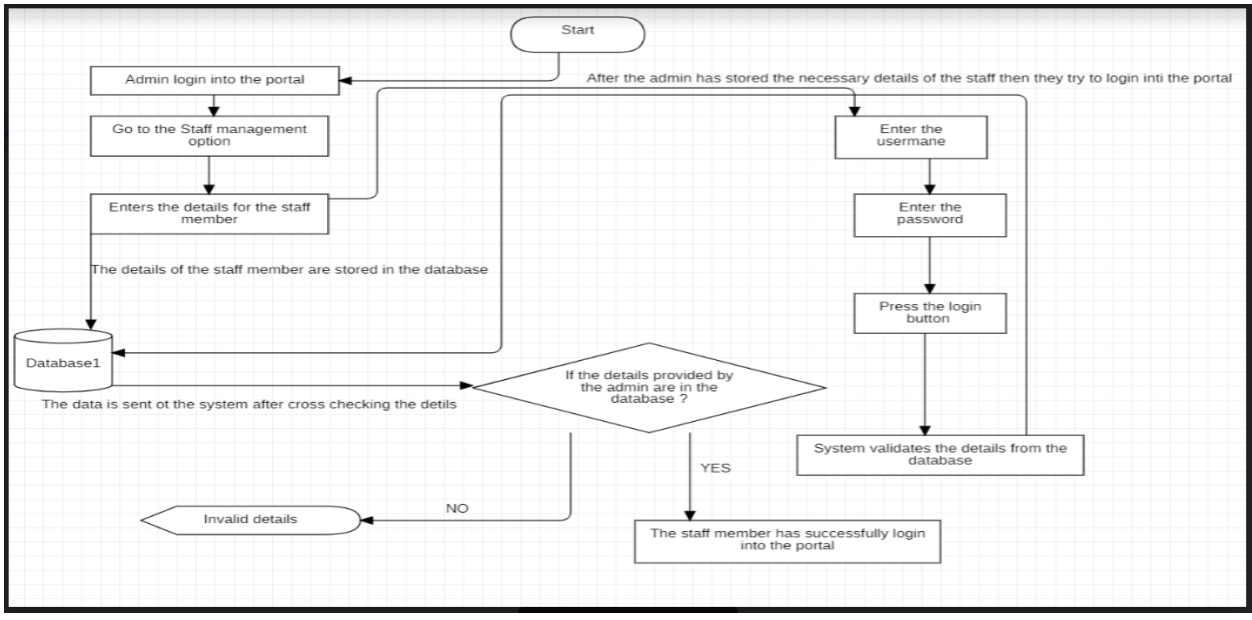
1. Processor: Minimum 1 GHz.
2. Memory (RAM): Minimum 2 GB.

2. Low Level Diagram

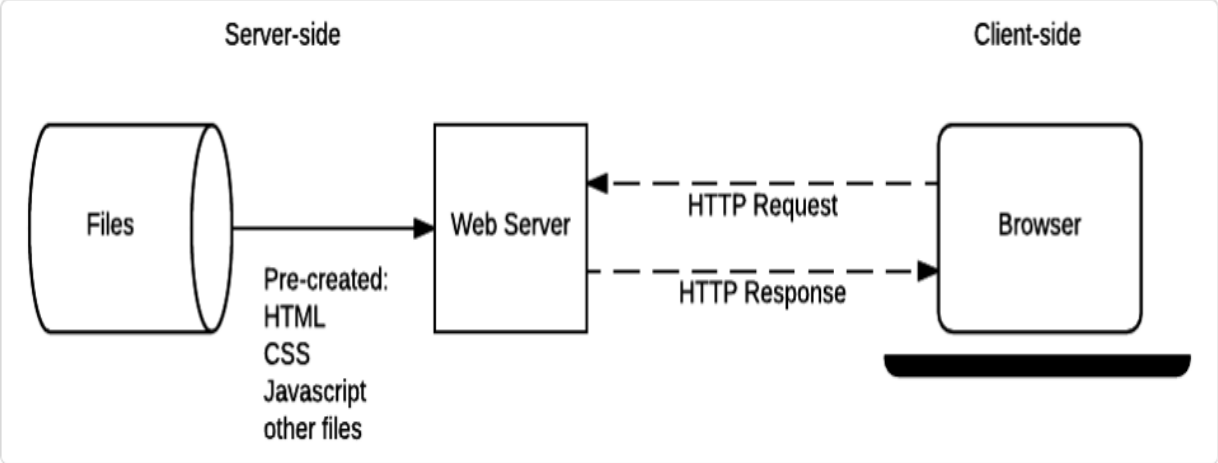
1.1 Sequence Diagram



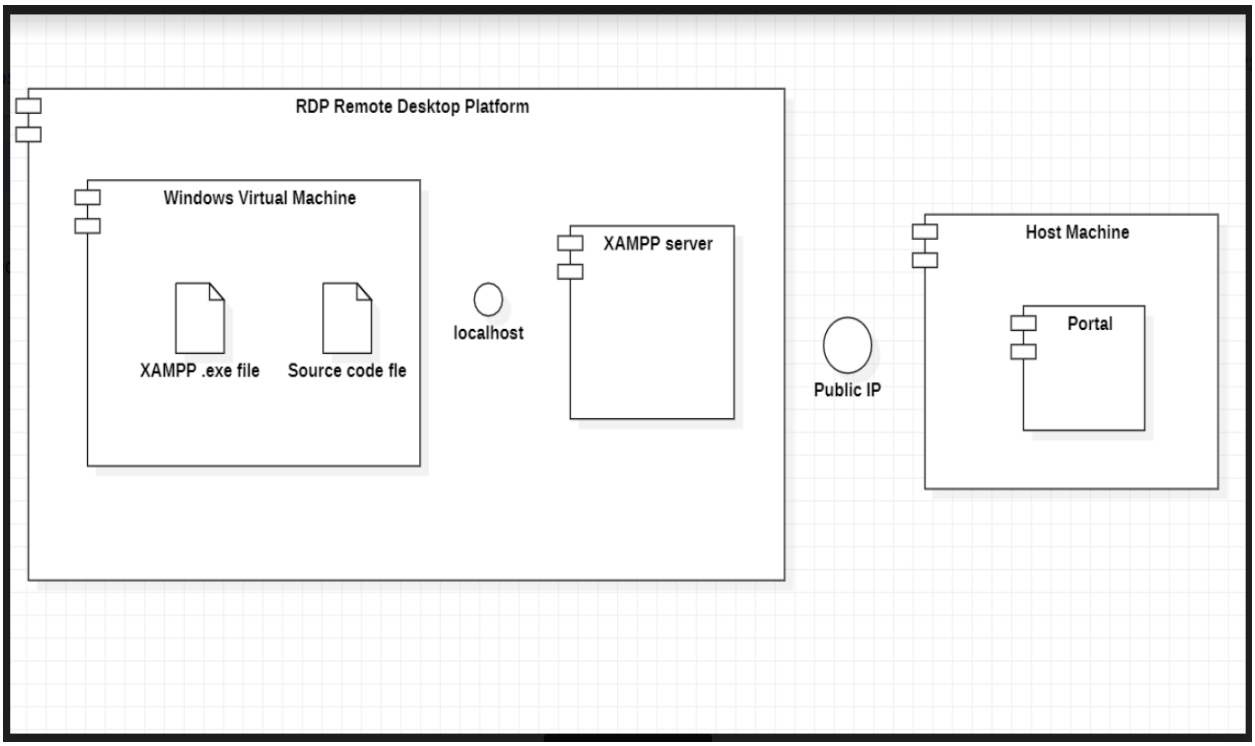
1.2 Navigation/UI Implementation



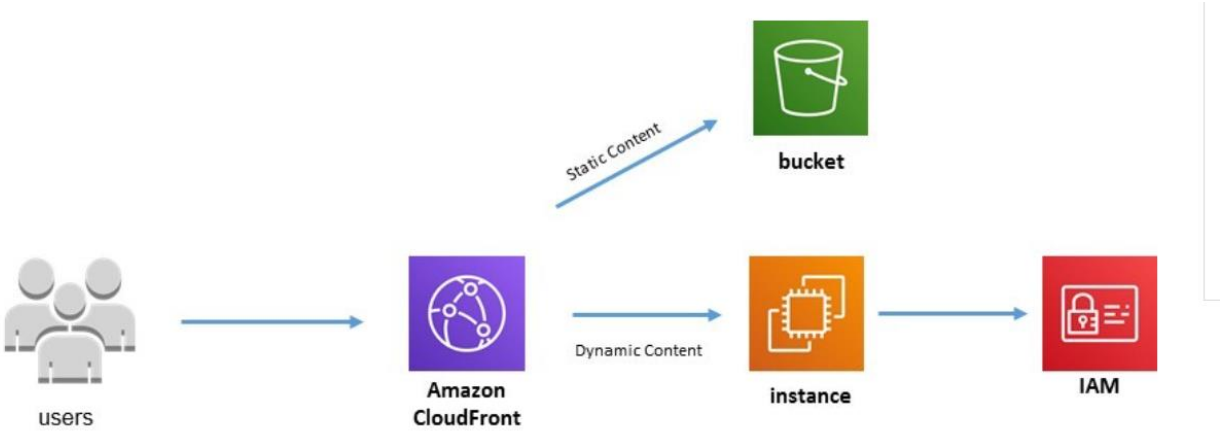
1.3 Client /Server Side Implementation



1.4 Component Design Implementation



1.5 Configurations/ Setting



3. Data Design

1.1 List of Key Schemas/Tables in database

Table Name – admin

Name	Type
user_id	integer
username	varchar
password	varchar
name	varchar

Table Name – announcements

Name	Type
id	integer
message	varchar
date	date

Table Name – attendance

Name	Type
id	integer
user_id	varchar
curr_date	text
Curr_time	text
present	tinyint

Table Name – equipment

Name	Type
id	integer
name	varchar
amount	integer
quantity	integer
vendor	varchar
description	varchar
address	varchar
contact	varchar
date	date

Table Name – members

Name	Type
user_id	integer
fullname	varchar
username	varchar
password	varchar
gender	varchar
dor	date
services	varchar
amount	integer
paid_date	date
psywar	integer
plan	varchar
address	varchar
contact	varchar
status	varchar
attendence_count	integer
curr_weight	integer
int_bodytype	varchar
curr_bodytype	varchar
progress_date	date
reminder	integer

Table name – rates

Name	Type
id	integer
name	varchar
charge	varchar

Table Name – reminder

Name	Type
id	integer
name	varchar
message	text
status	text
date	datetime
user_id	integer

Table Name – staffs

Name	Type
user_id	integer
username	varchar
password	varchar
email	varchar
fullname	varchar
address	varchar
designation	varchar
gender	varchar
contact	integer

Table Name – todo

Type	Name
id	integer
task_status	varchar
task_desc	varchar
user_id	integer

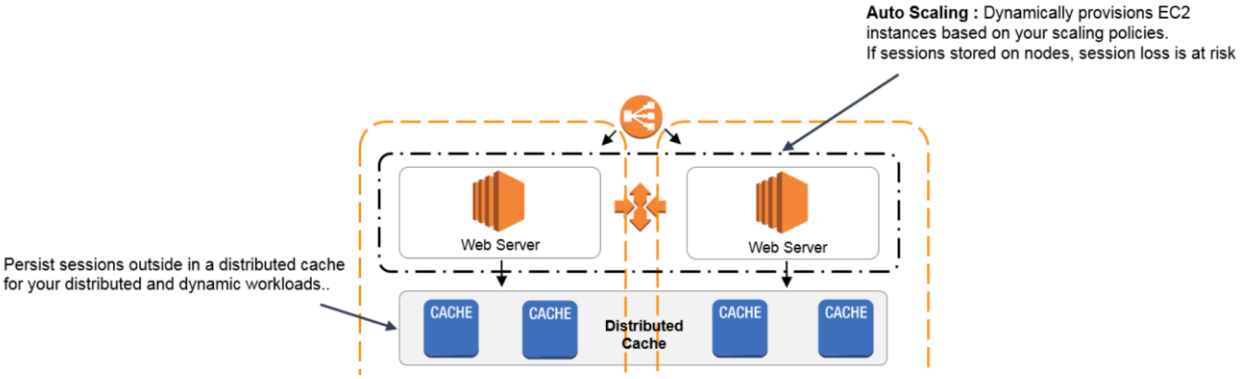
1.2 Key design considerations in data design

Key considerations when using AWS for web hosting

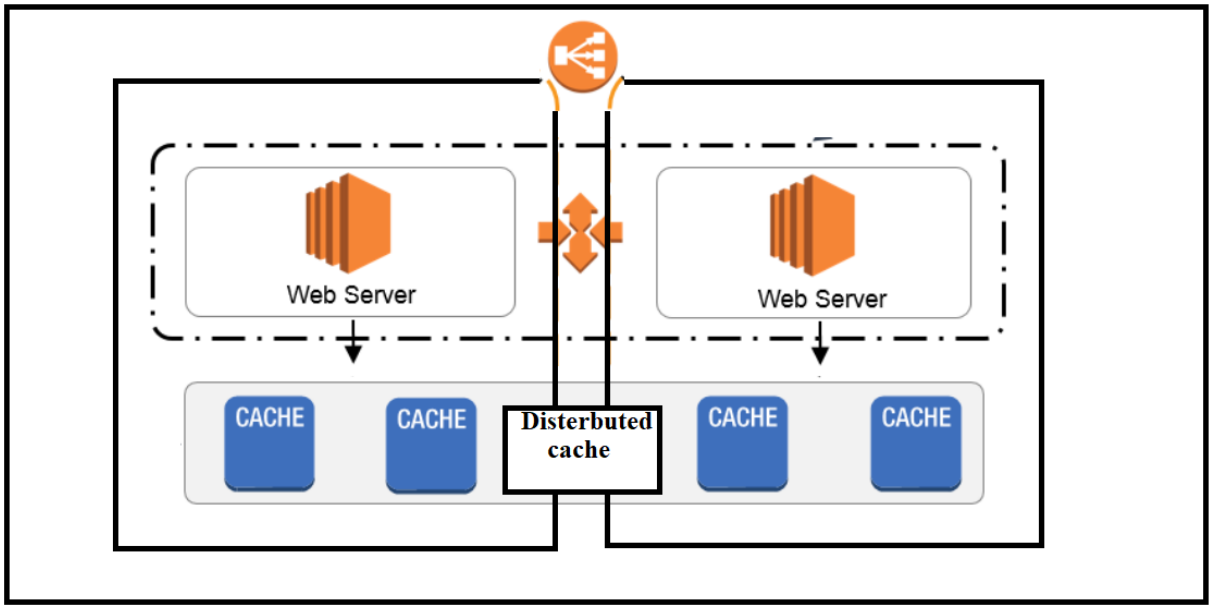
There are some key differences between the AWS Cloud and a traditional web application hosting model. The previous section highlighted many of the key areas that you should consider when deploying a web application to the cloud. This section points out some of the key architectural shifts that you need to consider when you bring any application into the cloud. You cannot deploy physical network appliances in AWS. For example, firewalls, routers, and load balancers for your AWS applications can no longer reside on physical devices, but must be replaced with software solutions.
There is a wide variety of enterprise-quality software solutions, whether for load balancing or establishing a VPN connection. This is not a limitation of what can be run on the AWS Cloud, but it is an architectural change to your application if you use these devices today.
Probably the most important shift in how you might architect your AWS application is that Amazon EC2 hosts should be considered ephemeral and dynamic.
Any application built for the AWS Cloud should not assume that a host will always be available and should be designed with the knowledge that any data in the EC2 instant stores will be lost if an EC2 instance fails.When a new host is brought up, you shouldn't make assumptions about the IP address or location within an Availability Zone of the host.

4. Details of other frameworks being used

1.1 Session Management



1.2 Caching



5. Key Notes

- Create S3 Bucket:
  We use S3 bucket to put our website's data.

- IAM Role:
  After uploading all necessary file into S3 bucket related to our website we have to assign IAM roles so that our EC2 instance will access S3.

- EC2 Instance:
  We have to create an EC2 instance to install Apache (/var/www/html) and copy the content of S3 to the HTML directory.

- Install a LAMP web server on Amazon Linux 2 .

6. References

[1]. Sam Alapati, Darl Kuhn, Arup Nanda AWS Certified Sys-Ops Administrator Associate www.authorityofaws.com [online], July 2019.

[2]. John Paul Mueller, Luca Massaron AWS For Admins For Dummies https://www.amazon.com/Admins-Dummies-John-PaulMueller, March 2016.

[3]. Raoul Alongi [online] The Most Complete Guide to Amazon Web Services from Beginner to Advanced Level, May 2019.

[4]. Julian Hunt, Amazon Web Services Beginners User Guide. The Ultimate Tutorial, July 2018.

[5]. Kent Erickson, Pat Voulter, Henry-World, Guide for Beginner's. AWS Tutorial https://www.amazon.com/dp/, Jan 2019.