IBM Innovation Center for Education

PHEMESOFT
Software That Matters
TRUSTED DELIVERY PARTNER for IBM ICE

# University of Petroleum and Energy Studies

UPES
UNIVERSITY OF TOMORROW

# Host Dynamic Website on AWS

## PROJECT SYNOPSIS

Digio

**BACHELOR OF TECHNOLOGY**

CSE specialization in CCVT

SUBMITTED BY

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

GUIDED BY

Yogesh Ghorpade

**Tittle**

1. Name : Devang Tyagi
2. Roll No: R110219041
3. Present official Address with E-mail, telephone No: B-67 Takshila Colony Meerut devt40363@gmail.com, 7252811356 .
4. B.Tech, CSE specialization in CCVT(Cloud Computing and Virtualization Technology).
5. CSE, Batch 1.
6. Host Dynamic Website On AWS.

## Table of Content

## Table of Images

## Introduction Page

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.

**Methodology/Planning of work**

Step 1: Create S3 Bucket
You will need to create an S3 bucket to put your website's files and folders.
From the S3 dashboard, click on Create bucket. Give the bucket a unique name, the name you choose must be globally unique.
Click on Disable for Bucket Versioning. You can also add a tag to the bucket for easy identification.
Step 2: Upload web files to the S3 bucket After creating the bucket, you need to upload your website's files and folders into it.
From the S3 dashboard, click on the name of the bucket you just created.
On the Objects tab, you can see that the bucket is currently empty, click on the Upload button. This should take you to the Upload page. Drag and drop the website files that were downloaded from this.
After the necessary files and folders have been added, scroll down and click on Upload.
Step 3: Create IAM Role Now, EC2 wants to pull code from S3. So you want to create IAM Role to give EC2 permission to access S3.
From the IAM dashboard, click on Roles. Then Click on Create role. Choose EC2 and click Next: Permissions [3]. Search for S3 and check AmazonS3FullAccess. Then click Next: Tags. Click on Next: Review. Give the role name and description. Then click on Create role.
Step 4 : Create an EC2 instance
You will need to create an EC2 instance to install Apache ( /var/www/html ) and copy the content of S3 to the HTML directory. To do this, from the Services drop-down, select EC2 from the Compute section. This should display the EC2 dashboard. From the EC2 dashboard, click on Launch Instance. And click on Next: Configure Instance Details. Determine 1 for the Number of instances, default vpc for Network, and Default in us-east-1a for Subnet. Choose ec2s3role or whatever you named for the IAM role [4]. Select Create a new security group. Give it Name: DynamicWebsiteSG and description: SG for DynamicWebApp. For SSH rule select My IP for Source. Click on Add Rule and select HTTP for Type and Anywhere for Source. The last rule select HTTPS for Type and Anywhere for Source. Click on Review and Launch. Click on Launch. Select Create a new key pair. Give the name for the key pair and download it.
Step 5: SSH with MobaXterm
Now, you want to connect to EC2 by using MobaXterm. First, you should copy public IPv4 address of the EC2 instance [5]. Open MobaXterm and start a new remote session by clicking on Session. Click on SSH. Click on Advanced SSH settings, check to Use private key and browse the location of key. Click OK. Now, you connected to EC2 successfully.
Step 6: Install a LAMP web server on Amazon Linux 2

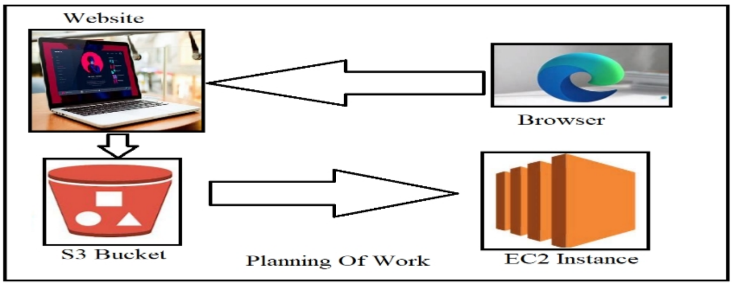

Fig. 1 Work flow of the project

## Facilities required for proposed work

**Software:**

1. MobaXterm

**Cloud Computing Platform:** AWS

**Services of AWS used:**

1. EC2
2. S3
3. IAM ROLES
4. VPC

**Hardware :** Laptop

## 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.