# Practical - 4

* **Title :-**

Connect with Google Cloud SQL and Run SQL Queries for Database Analysis.

* **Aim :-**

To establish a secure connection to a Google Cloud SQL instance, execute SQL queries for data manipulation and retrieval, and analyzes the result.

* **Procedure / Task :**

Step-1 Create and configure Cloud SQL Instance

→ In the google Cloud Console, go to the Cloud SQL Instance page.

→ Click "create Instance" and choose the desired database engine (Ex:- MySQL)

→ Assign an instance ID, set the root user password, and create the instance.

Step-2 Authorize Client IP and connect

→ Add your client machine's public IP address to the instance's authorized networks. ~~under~~

→ Connect to the instance using command:

:- gcloud sql connect [Instance_Name] --user = root

Step-3 Run SQL Queries

→ Use SQL commands to create db and table. then insert and retrieve data.

- Create Database dbName1;
  use dbName1;
- Create Table employees (id Int, name varchar(50), Salary float);
- Inser Into employees values (1, 'Virat', 50000);
- Select * from employees;

Result :-

- Successfully connected to Google Cloud SQL instance.
- Executed SQL queries and retrived data.

Parul® University | NAAC A++

Pg. No. : .....
Date : .....

# Practical - 5

**Title :-**

Creating an Amazon S3 Bucket for Cloud storage.

**Aim :-**

To Configure and launch a new, globally unique Amazon S3 bucket for storing and managing files and data objects on Amazon Web Services

**Procedure / Task**

Step-1 Sign in to AWS Management Console

Step-2 Navigate to Amazon S3 Service

- In the AWS console search bar, type "S3" and select S3 (Scalable storage in the cloud) to open the Amazon S3 console.

Step-3 Initiate Bucket Creation

- Click on the "Create Bucket" button in the S3 dashboard

Step-4 Configure Bucket Details

- Enter a globally unique bucket name.
- Select the AWS Region where you want the bucket to reside.

Step-5 Complete and create

- Review the option and click "create bucket"

Result:

A new, empty S3 bucket is created and ready for uploading files, folders or any object

Parul® University | NAAC A++

Pg. No.: .....
Date: .....

# Practical - 6

**Title:-**

Creating an Amazon EC2 Instance with windows OS.

**Aim:-**

To provision a new, remotely accessible windows virtual machine on AWS.

**Procedure / Task**

Step-1 Log in to AWS console.

Step-2 Open EC2 Dashboard.

- In the services search bar, type "EC2" and select EC2 to open the dashboard.

Step-3 Launch a New Instance.

- Click the "Launch Instance" button

Step-4 Configure Instance.

- Enter an Instance name.
- Choose a "Windows" Amazon Machine (AMI) such as microsoft windows Server (2019/2022)
- Select the instance type (Ex. t2.micro for free tier).

Pg. No.:
Date:

Step-5 Set key pair

- Create a new key pair or use an existing one.

Step-6 Network and Storage settings.

- Ensure RDP (Remote Desktop Protocol) traffic is allowed in the inbound security rules.

Step-7 Review and launch.

- Review your settings and click "Launch Instance".
- Wait for instance status to show as "Running".

Step-8 Connect to windows Instance

- Select your instance from the EC2 dashboard.
- Click "Connect", choose "RDP Client" and download the remote desktop file.
- Retrieve the Aministrator password.

Step-9 Remote Desktop Access

- Open the .rdp file with Remote Desktop Connection, enter the username and decrypted password.

result :-

A windows EC2 instance is running, accessible through RDP.

Parul® University | NAAC A++

Pg. No. : .....
Date : .....

# Practical-7

**Title :-**

Building and deploying a Containerized Application on GCP.

**Aim :-**

To containerize an application, push the container image to Google Container Registry (GCR), and deploy the container on Google Cloud Run for serverless hosting.

**Procedure / Task**

Step-1 Prepare Application Code.

- Create a simple application folder with your app files, for example, a basic HTML page.
- Ensure the presence of a Dockerfile in the root folder.

Step-2 Create Dockerfile.

→
```
From nginx:alpine
COPY . /usr/share/nginx/html
Expose 80
```

Step-3 Build a docker image locally

- Open terminal and navigate the app folder

Run:

docker build -t gcr.io/gcp-ok17/my-app-image.

Step-4 Configure Google Cloud SDK

- Install and authenticate Google Cloud SDK
- Run:

gcloud auth configure-docker

Step-5 Push Image to Google Container Registry

- Push your built image.

docker push gcr.io/gcp-ok17/my-app-image

Step-6 Deploy Container on Google Cloud Run

- Deploy the container image to cloud Run:

:- gcloud run deploy my-app-service --image gcr.io/gcp-ok17/my-app-image --platform managed --asia pacific(mumbai) Region --allow-unauthenticated

Step-7 Access your Application

- Once deployed, Cloud Run provides a public URL
- Open the URL in a browser to see your running app.

## Result :-

The containerized application is successfully deployed and running on Google Cloud Run.

# Practical-8

**Title :**

Deploy and Monitor a Web App Using Azure App Service.

**Aim :**

To deploy a web application on Azure App Service and monitor its health, usage and performance using Azure monitoring tools.

**Procedure / Task**

Step-1 Create an Azure Account and Login

Step-2 Create an App Service Plan

- In the Azure portal, search for "App service plans".
- Click "Create" and configure the plan by selecting subscription, resource group, region and pricing tier.

Step-3 Create a Web App.

- Search for "App Services" in the portal.
- Click "Create," then fill in details: app name, subscription, resource group.
- Click "review + create" and then "create".

Step-4 Deploy your Web Application

- Use one deployment methods such as: local Git Deployment

Ex:- Using Azure CLI to deploy a package app:

```
az webapp deployment source config-zip
--resource-group <ResourceGroupName> --name <AppName>
--src <path of file>
```

Step-5 Configure Monitoring

- Go to your Web App in the Azure portal.
- Under "Monitoring," enable Application Insights if not already enabled.

Step-6 View Metrics and logs

- In the Azure portal, open the App Service.
- Use Application Insights dashboard to see requests, response times, failures, dependencies.

Result:-

The web application is deployed successfully on Azure App Service and monitored actively via Azure Application.

Parul® University | NAAC A++

Pg. No. : .....
Date : .....

# Practical -9

**Title :-**

Sign Up, Explore Azure ML Studio, and create & Run an Azure ML Pipeline.

**Aim :-**

To get started with Azure ML studio, create a machine learning pipeline, and run experiments through the Azure ML Pipeline interface.

Procedure / Task

Step-1 Sign up and Access Azure ML Studio

- Sign up for a free Azure account if you don't have one.
- Navigate and create new machine learning workspace or use an existing one

Step-2 Explore Azure ML Studio Interface

- Familiarize with the dashboard: Datasets, Expts Pipelines, Models and Endpoints.
- Go to "Pipelines" from the left navigation pane.

Step-3 Create a New Azure ML Pipeline

- Click on "Create pipeline" and select the pipli

environment.

- Drag-and-drop modules to build your pipeline workflow.

→ Add datasets
→ Insert components like data ~~group~~ prep, training, scoring, evaluation.

- Configure each step by setting parameters and data inputs

Step-4 Run the Pipeline

- Click "Submit" to run the pipeline.
- Monitor the pipeline run status, logs and outputs through the interface.

Step-5 View Result

- Once the pipeline completes, review output datasets, trained models, and logs.

Step-6. Save and Reuse pipeline.

- Save the pipeline for future reuse.

Result:
Successfully signed up for Azure ML Studio, created a simple ML pipeline, executed it, & review the result.

# Practical - 10

**Title :-**

Visualize Data with Amazon QuickSight: Connect Data Source & Generate Visuals.

**Aim:**

To connect Amazon QuickSight with a data source, create a dataset, and generate multiple visualizations for analysis and presentation.

**Procedure / Task**

Step-1 Sign up for Amazon QuickSight

- Go to the AWS Console, search for "QuickSight" and signup if you haven't already.

Step-2 Set up Data Source

- Prepare your test dataset (eg:- csv file in S3 or another AWS-supported data source).
- Ensure your AWS permissions allow QuickSight access to this data source.

Step-3 Connect QuickSight to Data Source

- In QuickSight, click "Datasets" then "New Dataset."
- Choose the type (Amazon S3, RDS, Redshift etc.)

- For S3 : Provide the S3 URI or upload manifest.json as guided.
- For Databases: Enter Connection info (endpoints, db name, credentials)
- Confirm access – Quicksight may request additional permission.

Step-4 Import dataset

- Select the uploaded/connected dataset.
- Click "Save & visualize" to proceed

Step-5 Create visualizations.

- Click "New Analysis" > Select your dataset > "Create analysis"
- Use the "visual types" menu to pick chart styles (e.g. bar chart, pie chart, line graph)
- Drag and drop fields from your data into Value, Group/Category, or Filters areas.
- Configure and format visuals as desired.

Step-6 Customize and combine Visuals.

- Add more visuals to the same analysis for deeper insights.

Step-7 Share or publish Dashboards

- Combine visualizations onto a dashboard.
- Save and share the analysis within your organization.

Result:

- Amazon QuickSight will display dataset as a variety of interactive.

# Practical - 11

## Title

Develop, Build and deploy a container Application on Google Compute Engine

## Aim:

To containerize an application and deploy it as a running container on a VM instance in GCP

## Procedure / Task

Step-1 Prepare Application and Dockefile.

- Create app or static HTML site.
- Add a dockerfile:
  :- From nginx:alpine copy. /user/share/nginx/ht

Step-2 Build your container image.

- Open terminal, navgate to app folder.
- Build the image:
  :- docker build -t gcr.io/gcr.00042/my-app:latest.

Step-3 Push image to Google Container Registry

- Authenticate Docker to GCR
  :- gcloud auth configure-docker

- Push the image
  :- gcr.io/gcr-00042/my-app:latest

Step-4 Create a VM Instance with Container Support in GCE.

- Go to Google Cloud Console > Compute Engine > VM Instances > Create Instance.
- Set Region asia-South1
- Under "Boot disk" Select "Container-optimized OS"
- Expand "Container" section, click "Deploy container"
- Enter gcr.io/project_id/my-app:latest as the container image.
- Configure port.
- Finish creation by clicking "create"

Step-5 Access your Application.

- Once the VM is Running, copy its external IP from the console.
- Access http://External_IP in your browser to view your application.

Result:-

Your application is containerized, stored in GCR, and deployed on a Google compute engin