1. Write a Docker Script to perform the following actions.

* Download the Docker image for SQL Server Developer Edition
* Have a folder called DBScripts in the Docker root container where all the SQL Server Database scripts are present
* The .sql file in the DBScripts folder should run in Ascending order in the Database when the Docker image is built
* The SQL Server Developer edition should be exposed to the Docker host machine so that the Reporting tools / SSMS can connect to it from the host machine

please test with the following SQL scripts in the DBScripts folder.

* 01-create-database.sql

CREATE DATABASE testDB;

* 02-create-table.sql

USE DATABASE testDB;

CREATE TABLE Test

(

Id int,

Data varchar(50)

);

* 03-insert-data.sql

INSERT INTO Test (

Id, Data

)

VALUES (

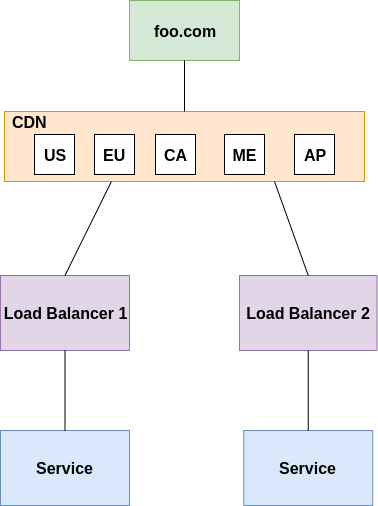
1, 'A'

);

The expected outcome should be

* Docker container with the SQL Server Developer tools installed
* SQL Port exposed outside to the host image
* The Database and Table created with the Sample

1. We have a high availability application running with the following Architecture.



* Explain the Architecture model of this and the Benefit
* How do you roll out an update to the service without any downtime impact to the app foo.com

Generally for availability users of your web service will hit it via a load balancer (like nginx) which then routes requests to a pool of instances of your application.

This means if one instance falls over, you have availability because the load balancer will just send the traffic to the instances that *are* working.

For that reason, when you deploy you can take down one instance, upgrade, then another, etc until they are all updated. During this you will always have at least one running.

In addition this lets your application "horizontally scale" very easily as it's trivial to add more instances of your application behind the load balancer.

There are more involved ways of doing this but they all mainly work on the premise of some kind of load balancer / router managing traffic so that there is always a running app available

* If you have to deploy this App to the Cloud what services would you use and explain the Architecture. You can mention for Cloud of your choice (AWS / Amazon / Google Cloud (GCP) / Oracle / Alibaba Cloud)

1. Write a Python program to build an automated Data Pipeline. The program should be a single file python code and should be able to do the following.
   1. Navigate to a particular folder and execute other Python scripts which contains some kind of Data Ingestion / Transformation
   2. Take the following as command line input (bash style preferred)
      1. A folder name – Mandatory - The name of the folder to work upon
      2. Script Order – Optional - Ascending / Descending – The list of files order to work upon. Default Behavior – Ascending
      3. File name – Optional – The exact file name to run. Default Behavior – All the files
      4. Outfile – Optional – Log the output to the mentioned file name. Default behavior – Console
      5. Help – Optional - Display the help and usage information
   3. The following should be displayed as Output (beautification / Table preferred)
      1. The Folder where it is executed
      2. The file/files it is executed (in the order of execution)
      3. The output of the files executed. Just the console output returned should be okay