|  |  |
| --- | --- |
|  |  |
| Cloud Project |  |
|  |  |
|  | 26 May 2020AWS Website |
|  | Mazen Hossamledeen182209 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | |  | |
|  | Description *Highly available and scalable web hosting can be a complex and expensive proposition. Traditional scalable web architectures have not only needed to implement complex solutions to ensure high levels of reliability, but they have also required an accurate forecast of traffic to provide a high level of customer service. Dense peak traffic periods and wild swings in traffic patterns result in low utilization rates of expensive hardware. This yields high operating costs to maintain idle hardware, and an inefficient use of capital for underused hardware.*              Single VPC Scenario         Architecture *Used AWS Services:*  *- VPC*  *- RDS*  *- EC2*  *Used Tools:*  *- PuTTY*  *- PuTTYgen*  *- WinSCP*  *- MySQL Workbench* Components**VPC**:Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.**RDS**:Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.**EC2**:Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.ImplementationCreated a VPC with Private and Public Subnets to hold the website required components inside. Then created a security group for that VPC, to be publicly accessed, where inbound rules were added to allow traffic to connect from the internet.Next step was to create a private access security group for the, yet to be mentioned, database service in order to keep the database instance private. Then adding inbound rules to only allow web server traffic.In this step RDS MySQL DB instance is created to maintains the data used by a web application.After that, a web server is created to connect to the Amazon RDS DB instance created earlier. And this is achieved by creating an Amazon EC2 (Amazon Linux AMI) instance in the public subnet of the VPC.Then, the Apache web server was installed on the EC2 instance to be able to connect with PHP. This is where PuTTY & PuTTYgen came in handy. After successfully installing Apache, WinSCP was used to transfer the website files to the instance and facilitated the process of editing these files later, instead of using the CMD for interactions.  * Lastly a file (“dbinfo.inc”) was created to hold the RDS DB instance information which enabled the Apache web server to communicate with the DB.  Performance AnalysisAt first, while using PuTTY CMD to access the Linux instance it was a tedious process to constantly view and edit the files, but after using WinSCP it became much easier.Overall AWS is unconditionally better at handling a web server than just using a single personal computer, as some of its traits being:- A Cost-Effective Alternative to Oversized Fleets Needed to Handle Peaks- A Scalable Solution to Handling Unexpected Traffic Peaks- An On-Demand Solution for Test, Load, Beta, and Preproduction EnvironmentAs for the final tests, the website was tested multiple times from multiple locations and devices and still works smoothly, and regarding the DB and the DB connection they are both functioning as expected with no bugs or whatsoever.Website Link:<http://ec2-54-81-156-62.compute-1.amazonaws.com/home.html> GitHub link:  <https://github.com/MazenHossam/Cloud-Computing> Demo Link:<https://youtu.be/SWZ--Axqq1E> | | |  | |
|  | References: <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/TUT_WebAppWithRDS.html>  <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>  <https://www.tutorialspoint.com/php/php_mysql_login.htm>  <https://www.youtube.com/watch?v=Ng_zi11N4_c>  <https://youtu.be/xzCgeRxSzy4> | |  | |  | |