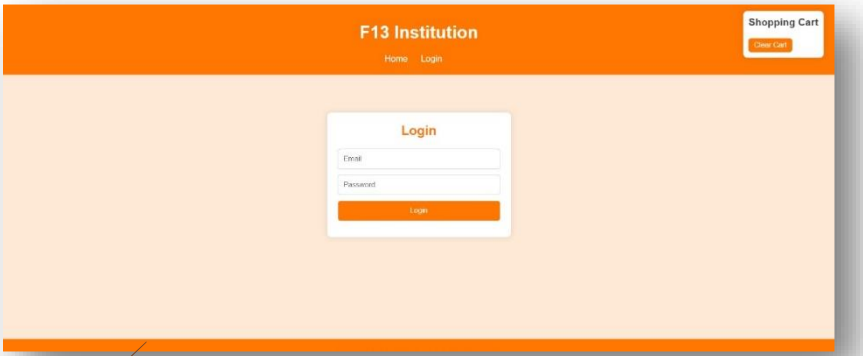
4/3/2024

Deploying Dynamic Website on AWS using SQL Database



Submitted By:

Tushar Sehdev
Museib Patel
Mohd Abdul Ghani
Dhatchana Moorthy Mathan

Submitted To:

F13 Technologies

INDEX

1. Introduction

2. Deploying a PHP and MySQL Website on AWS

3. AWS Services Overview

4. Deployment Steps

5. Architectural Diagram

6. Performance Analysis

7. Cost Analysis

8. Advanced topics of SQL database.

9. Comparative analysis of SQL with other databases.

10.Conclusion

11. References

INTRODUCTION

In the dynamic landscape of web development and cloud computing, deploying dynamic websites on Amazon Web Services (AWS) is a pivotal skill. As businesses transition their digital infrastructure to the cloud, AWS stands out as a dominant force, offering a comprehensive suite of tools and services. This project explores the seamless integration of cloud-based solutions, leveraging AWS services like Amazon EC2, Amazon RDS, AWS Lambda, and Amazon S3 to host and manage dynamic web content efficiently and at scale.

Our focus extends to the intricacies of deploying dynamic websites on AWS, specifically integrating SQL databases. With a sophisticated combination of services, including Amazon EC2 for computing power, Amazon RDS for database management, and AWS Lambda for serverless execution, we navigate the complexities of this integration. Join us on this journey as we unravel the synergy between dynamic web content and AWS, ushering in a new era of digital innovation and efficiency.

This project aims to unravel the intricacies of deploying dynamic web content on AWS, leveraging its extensive suite of offerings such as Amazon EC2, Amazon RDS, AWS Lambda, Amazon S3, and more.

Deploying a PHP and MySQL Website on AWS

Project Overview:

Welcome to our presentation on deploying a PHP and MySQL website on Amazon Web Services (AWS). This project focuses on creating and hosting a dynamic website powered by PHP for server-side scripting and a MySQL database for efficient data management.

Why AWS?

AWS has been selected as the cloud platform for this endeavor due to its outstanding capabilities and features perfectly suited for modern web hosting. Let's delve into why AWS stands out as our preferred choice.

Scalability:

AWS offers exceptional scalability options, enabling us to seamlessly adapt to fluctuations in user traffic. Whether your website experiences a sudden surge in visitors or requires additional resources for specific events, AWS provides both vertical and horizontal scaling solutions to meet these demands effortlessly.

Reliability:

Reliability is paramount in website hosting, and AWS excels in this aspect. With a reputation for high availability and minimal downtime, AWS infrastructure ensures that your website remains accessible to users consistently.

Flexibility:

Flexibility is a cornerstone of AWS's offerings. Throughout this project, you'll discover a multitude of services and configuration options tailored to our PHP and MySQL website's unique requirements. This flexibility allows us to customize our infrastructure for optimal performance and cost efficiency.

Competitive Advantage:

By opting for AWS, we gain a significant competitive edge. With AWS managing the complexities of infrastructure, we can allocate more resources and time.

AWS Services Overview

Amazon Web Services (AWS) provides an extensive array of cloud services, each tailored to cater to distinct requirements in constructing, deploying, and administering applications in the cloud. In this presentation segment, we will present an outline of the fundamental AWS services utilized for the deployment of our PHP and MySQL website.

Amazon RDS (Relational Database Service):

Managed Database Service: Amazon RDS is a managed database service that simplifies the process of setting up, operating, and scaling relational databases. We'll use it to host our MySQL database, ensuring data integrity and scalability.

Amazon EC2 (Elastic Compute Cloud):

Virtual Servers: Amazon EC2 provides resizable compute capacity in the cloud. We'll use EC2 instances to host our PHP web application. These instances can be easily scaled up or down to handle varying levels of traffic.

Security Groups:

Security Groups act as virtual firewalls for our EC2 instances and RDS databases. They allow us to define inbound and outbound traffic rules, enhancing the security of our infrastructure.

Apache Web Server:

Open-Source Web Server: Apache HTTP Server, often referred to as Apache, is an open-source web server software widely used for serving web content and executing PHP scripts. It plays a crucial role in delivering our website to users.

Amazon VPC (Virtual Private Cloud):

Isolated Network Environment: Amazon VPC allows us to create isolated network environments within AWS. We'll use VPC to control network traffic, define IP address ranges, and ensure the security and isolation of our resources.

AMAZON RDS

Amazon RDS, or Amazon Relational Database Service, is a managed database service offered by AWS. It allows us to set up, operate, and scale relational databases with ease. In our case, we're using MySQL as the database engine for our PHP website.

Benefits of Amazon RDS:

Here are the benefits of Amazon RDS

1. Automated Backups: Amazon RDS automates database backups, ensuring regular backups of your data. This feature simplifies data protection and allows for easy restoration in case of data loss or corruption.

2. Scalability: Amazon RDS provides vertical scalability, allowing you to easily adjust the compute and memory resources of your database instance to handle changing workloads. This scalability feature ensures optimal performance as your application grows.

3. High Availability: Amazon RDS offers high availability options such as Multi-AZ deployments, which replicate your database across multiple availability zones. This redundancy minimizes downtime and ensures continuous availability of your application, even during infrastructure failures.

4. Managed Service: With Amazon RDS, routine database maintenance tasks such as patching, monitoring, and backups are managed by AWS. This frees up your time and resources, allowing you to focus on developing and optimizing your applications rather than managing database infrastructure.

5. Security: Amazon RDS enhances database security with features like network isolation, encryption at rest and in transit, and integration with AWS Identity and Access Management (IAM). These security measures help protect your data from unauthorized access and ensure compliance with regulatory requirements.

Amazon Ec2

Amazon EC2, or Elastic Compute Cloud, provides resizable compute capacity in the cloud. It's essentially virtual servers that we can use to host our PHP application and serve web content.

Benefits of Amazon EC2:

Full Control: With EC2 instances, we have full control over the virtual server's configuration, including the choice of operating system, software installations, and server settings.

Scalability: EC2 instances can be easily scaled up or down based on demand. This makes it ideal for handling traffic spikes or adjusting resources as needed.

Variety of Instance Types: AWS offers a wide range of EC2 instance types optimized for different use cases, from general-purpose instances to memory-optimized or GPU-powered instances.

Customizable Networking: You can configure VPCs (Virtual Private Clouds) and network settings to control traffic flow and security.

Amazon EC2, or Elastic Compute Cloud, provides resizable compute capacity in the cloud. It's essentially virtual servers that we can use to host our PHP application and serve web content.

Security Groups

Security Groups are AWS's built-in firewall for EC2 instances. They act as virtual firewalls to control inbound and outbound traffic to your instances.

Importance of Security Groups:

Network Security: Security Groups are crucial for securing your EC2 instances and RDS databases. By defining rules for allowed traffic, you control who can access your resources.

Granular Control: You can specify rules based on ports, IP ranges, and protocols. This allows you to restrict access to only necessary services and IP addresses.

Fine-Tuned Access: Security Groups provide granular control over which services are accessible, minimizing security risks and ensuring data protection.

APACHE WEB SERVER

The Apache HTTP Server, commonly referred to as Apache, is an open-source web server software. It's a crucial component for hosting our PHP-based website and serving web content.

Importance of Apache:

Web Content Delivery: Apache's primary role is to serve web content, including PHP scripts. It handles incoming HTTP requests and sends the appropriate responses to clients' web browsers.

Configuration: We'll configure Apache to work seamlessly with our PHP application. This involves specifying how requests should be handled, including routing PHP requests to the PHP interpreter.

Customization: Apache allows us to customize server settings to match the requirements of our website, making it a versatile and adaptable web server.

AMAZON VPC

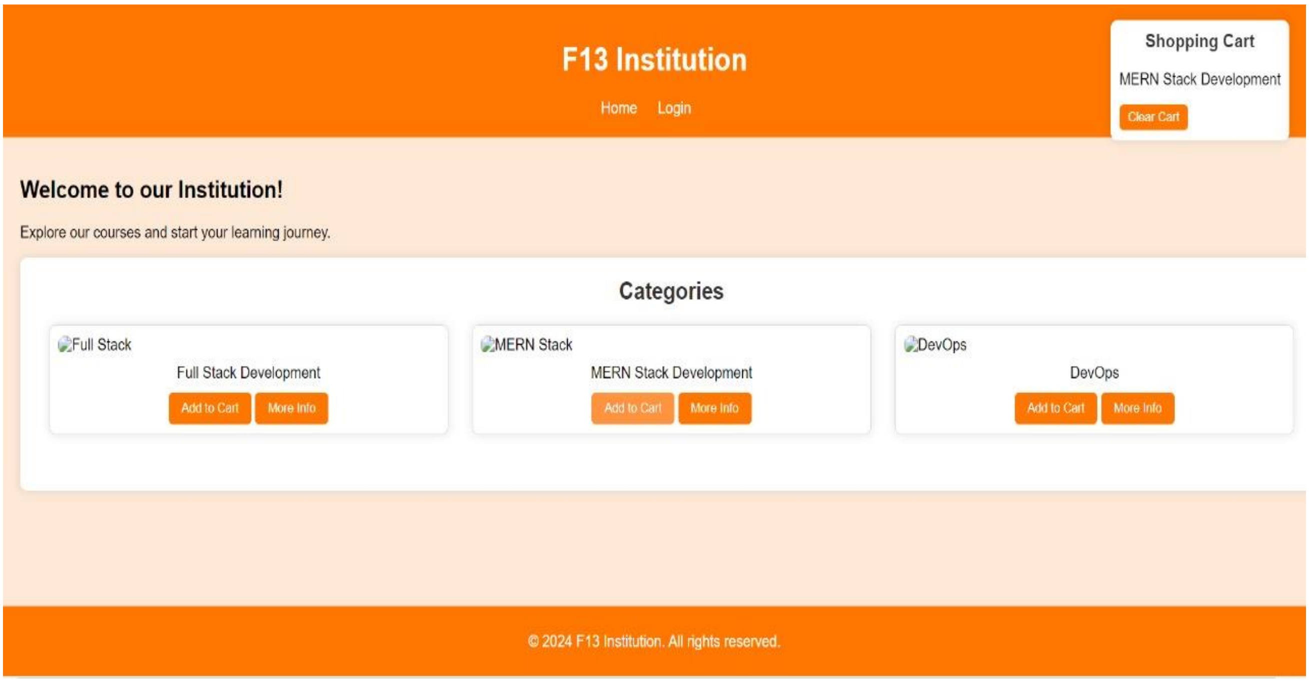
Amazon VPC, or Virtual Private Cloud, is a virtual network environment that you create within the AWS cloud. It provides network isolation and segmentation for your AWS resources.

Importance of Amazon VPC:

Network Isolation: Amazon VPC allows us to isolate our resources from other AWS customers, providing a secure network environment.

Control over IP Addressing: You can define IP address ranges, subnets, and route tables within your VPC, giving you control over how traffic flows to and from your resources.

Security and Routing: VPCs enable you to configure network security through security groups and network ACLs (Access Control Lists). You can also set up custom routing to direct traffic as needed.



F13 Institution

Shopping Cart

MERN Stack Development

Welcome to our Institution!

Explore our courses and start your learning journey.

Categories

Full Stack Development

MERN Stack Development

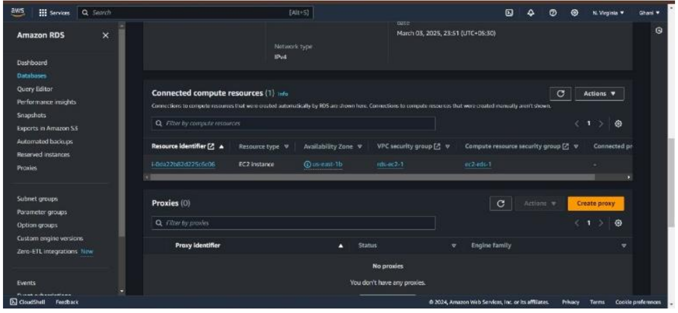
DevOps

© 2024 F13 Institution. All rights reserved.

Deployment Steps

Step 1: Set up Amazon RDS for MySQL:

Begin by setting up Amazon RDS as our MySQL database server. Configure the database instance with the necessary parameters, including database engine, size, and security settings. Amazon RDS will handle database management tasks such as backups, maintenance, and scaling.



March 03, 2025, 23:51 (UTC+05:30)

Network type
IPv4

Connected compute resources (1) Info

Connections to compute resources that were created automatically by RDS are shown here. Connections to compute resources that were created manually aren't shown.

Filter by compute resources

Resource identifier	Resource type	Availability Zone	VPC security group	Compute resource security group	Connected pr
i-0da22b82d225c6c06	EC2 instance	us-east-1b	rds-ec2-1	ec2-rds-1	-

Proxies (0)

Filter by proxies

Proxy identifier	Status	Engine family

No proxies

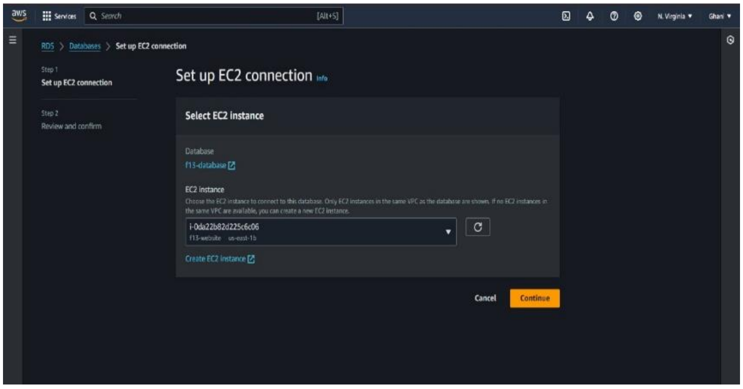
You don't have any proxies.

© 2024, Amazon Web Services, Inc. or its affiliates.

Img:1

Step 2: Launch Amazon EC2 Instances:

Next, we'll launch Amazon EC2 instances to host our PHP web application. Choose the appropriate EC2 instance types based on your application's requirements. These instances will serve as our web servers.



Set up EC2 connection Info

Select EC2 instance

Database
f13-database

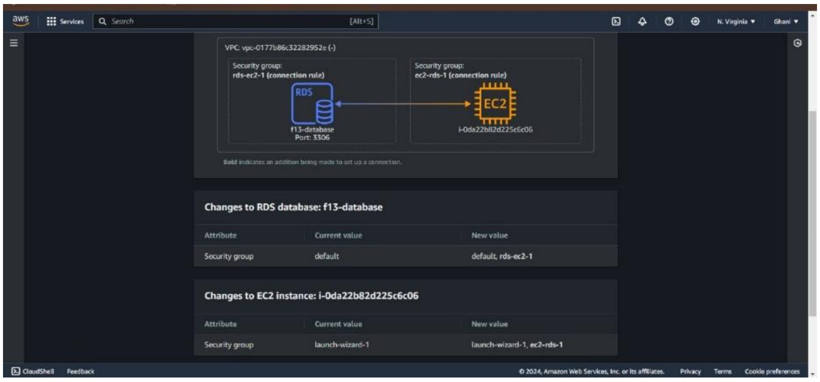
EC2 instance
Choose the EC2 instance to connect to this database. Only EC2 instances in the same VPC as the database are shown. If no EC2 instances in the same VPC are available, you can create a new EC2 instance.

i-0da22b82d225c6c06
f13-website us-east-1b

Img:2

Step 3: Configure Security Groups:

Security is paramount. In this step, we'll configure Security Groups to define inbound and outbound traffic rules for our EC2 instances and RDS database. This ensures that only authorized traffic can access our resources.



VPC: vpc-0177b86c32282952e (-)

Security group: rds-ec2-1 (connection rule)

f13-database
Port: 3306

Security group: ec2-rds-1 (connection rule)

i-0da22b82d225c6c06

Bold indicates an addition being made to set up a connection.

Changes to RDS database: f13-database

Attribute	Current value	New value
Security group	default	default, rds-ec2-1

Changes to EC2 instance: i-0da22b82d225c6c06

Attribute	Current value	New value
Security group	launch-wizard-1	launch-wizard-1, ec2-rds-1

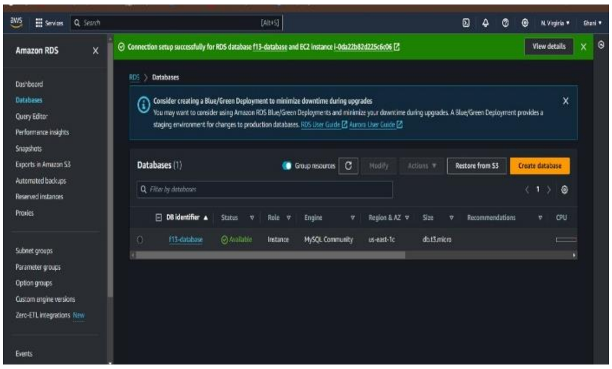
Img:3

Step 4: Install and Configure Apache Web Server:

With our EC2 instances in place, we'll install and configure the Apache Web Server. This includes setting up virtual hosts, configuring PHP, and defining how Apache handles incoming requests.

Step 5: Connect Web Servers to RDS Database:

Establish a connection between our web servers (EC2 instances) and the RDS database. Configure the database credentials in your PHP application to enable data retrieval and storage.

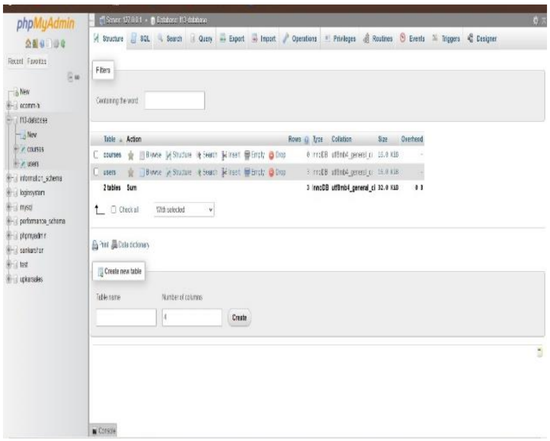


Connection setup successfully for RDS database f13-database and EC2 instance i-0da22b82d225c6c06

Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU
f13-database	Available	Instance	MySQL Community	us-east-1c	db.t3.micro		



Server: 127.0.0.1 » Database: f13-database

Containing the word:

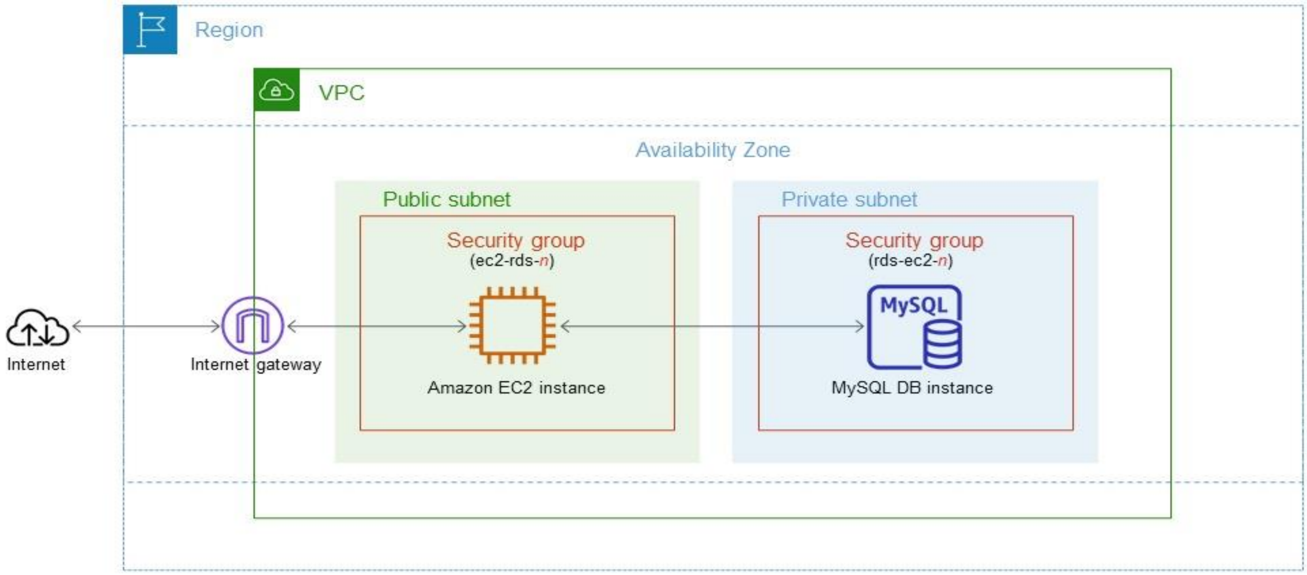
Table	Action							Rows	Type	Collation	Size	Overhead
courses	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	☆	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
2 tables	Sum							3	InnoDB	utf8mb4_general_ci	32.0 KiB	0 B

□ Check all With selected:

Table name:
Number of columns: 4

Img:4
Img:5

Architectural Diagram



- Web Application Tier:

Amazon EC2 Instances: These virtual servers host your web application code written in PHP. You can have multiple EC2 instances for scalability and redundancy.

- Database Tier:

Amazon RDS: This managed relational database service hosts your MySQL database. You can choose the appropriate instance type based on your performance and storage requirements.

Multi-AZ Deployment: For high availability and fault tolerance, you can deploy your RDS instance in multiple Availability Zones.

- Networking:

Security Groups: Security groups act as virtual firewalls, controlling inbound and outbound traffic to your EC2 instances and RDS databases.

- Storage:

Amazon S3: You can use S3 to store static assets such as images, CSS files, and JavaScript files. These assets can be served directly from S3 or through a content delivery network (CDN) like Amazon CloudFront for faster delivery to users.

CONCLUSION

In conclusion, let's recap the key highlights from our presentation:

- We delved into the benefits of leveraging AWS for hosting our PHP and MySQL website, highlighting its scalability, reliability, and flexibility.

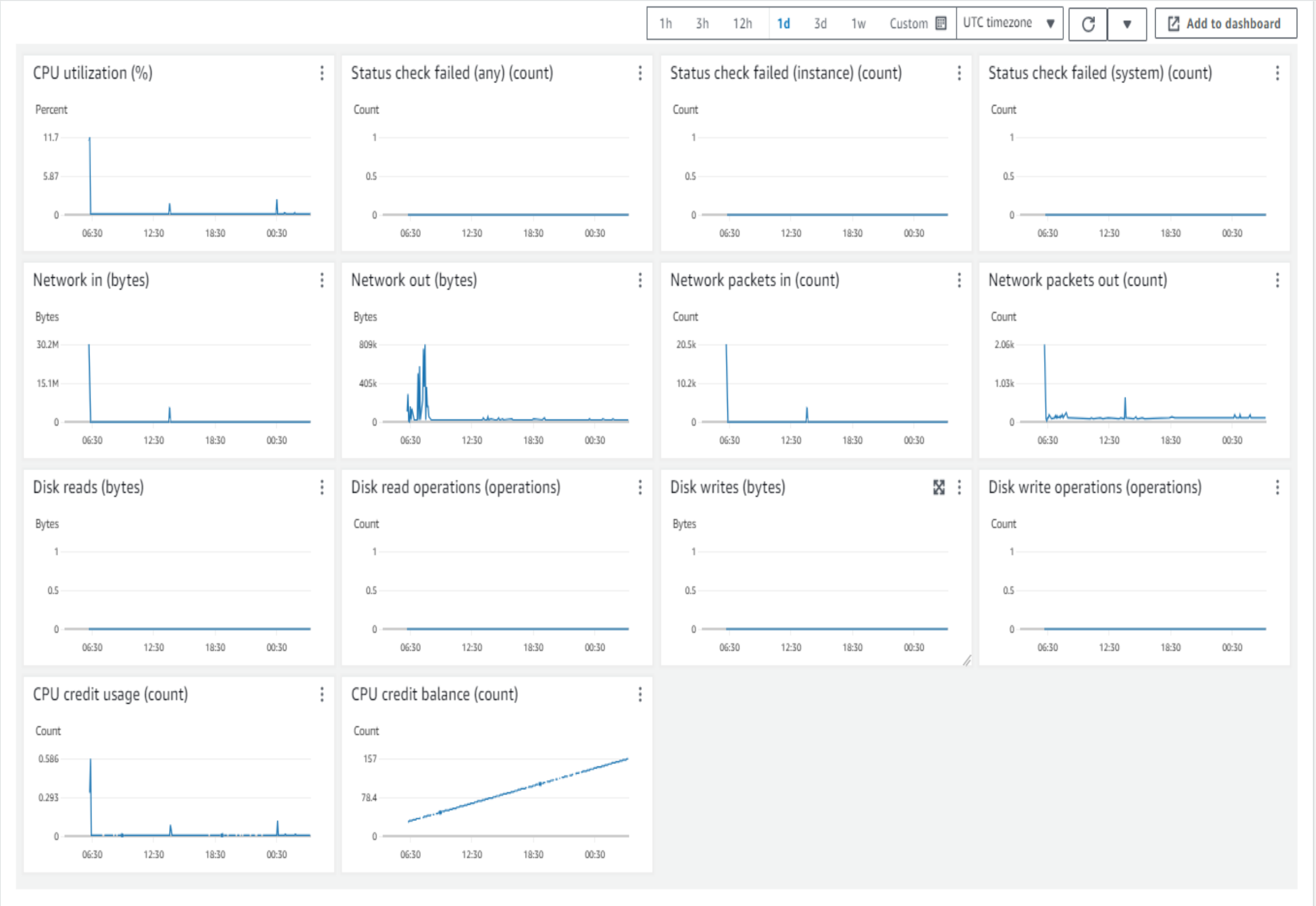
- Essential AWS services such as Amazon RDS, Amazon EC2, Security Groups, Apache Web Server, Amazon VPC, and Elastic IPs were introduced, each playing a crucial role in our deployment process.

- To enhance security, RDS is deployed within a private subnet group, ensuring a heightened level of protection for our database.

- Both EC2 and RDS instances are equipped with separate Security Groups, further bolstering our infrastructure's security posture.

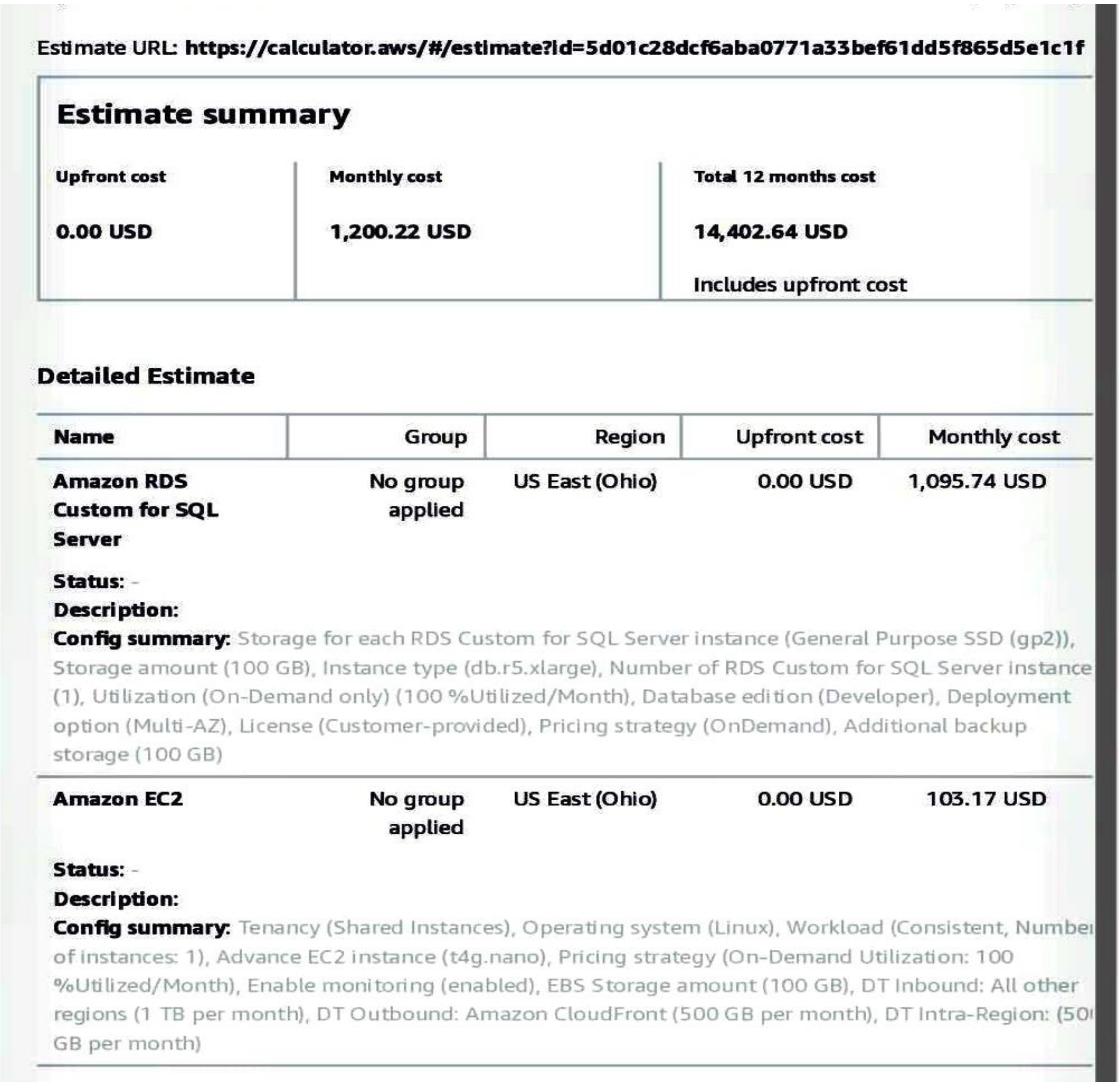
- Additionally, a Load Balancer is seamlessly integrated with our EC2 instances to optimize performance and distribute traffic efficiently across our deployment.

Performance Analysis



- After Deployment the system is tested for 1 day. Everything is working well.

Cost Analysis



Estimate URL: **https://calculator.aws/#/estimate?id=5d01c28dcf6aba0771a33bef61dd5f865d5e1c1f**

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	1,200.22 USD	14,402.64 USD Includes upfront cost

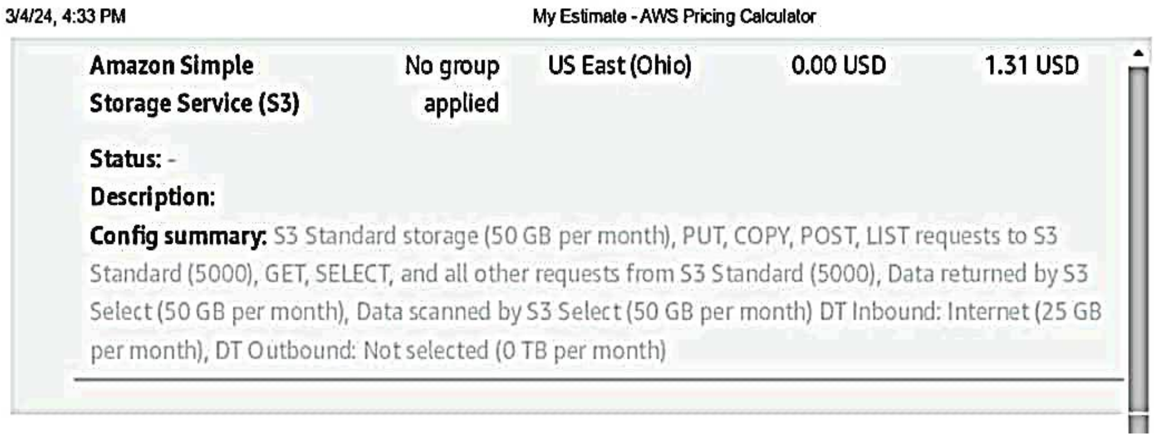
Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon RDS Custom for SQL Server	No group applied	US East (Ohio)	0.00 USD	1,095.74 USD

Status: -
Description:
Config summary: Storage for each RDS Custom for SQL Server instance (General Purpose SSD (gp2)), Storage amount (100 GB), Instance type (db.r5.xlarge), Number of RDS Custom for SQL Server instance (1), Utilization (On-Demand only) (100 %Utilized/Month), Database edition (Developer), Deployment option (Multi-AZ), License (Customer-provided), Pricing strategy (OnDemand), Additional backup storage (100 GB)

Amazon EC2	No group applied	US East (Ohio)	0.00 USD	103.17 USD

Status: -
Description:
Config summary: Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Numbe of instances: 1), Advance EC2 instance (t4g.nano), Pricing strategy (On-Demand Utilization: 100 %Utilized/Month), Enable monitoring (enabled), EBS Storage amount (100 GB), DT Inbound: All other regions (1 TB per month), DT Outbound: Amazon CloudFront (500 GB per month), DT Intra-Region: (50 GB per month)



3/4/24, 4:33 PM My Estimate - AWS Pricing Calculator

Amazon Simple Storage Service (S3)	No group applied	US East (Ohio)	0.00 USD	1.31 USD

Status: -
Description:
Config summary: S3 Standard storage (50 GB per month), PUT, COPY, POST, LIST requests to S3 Standard (5000), GET, SELECT, and all other requests from S3 Standard (5000), Data returned by S3 Select (50 GB per month), Data scanned by S3 Select (50 GB per month) DT Inbound: Internet (25 GB per month), DT Outbound: Not selected (0 TB per month)

Advanced topics of SQL database.

1. SQL Databases in AWS:

AWS offers various managed SQL database services, including Amazon RDS (Relational Database Service), Amazon Aurora, and Amazon Redshift. These services support various database engines like MySQL, PostgreSQL, SQL Server, Oracle, and more.

2. Database Replication:

Replication is the process of copying data from one database to another. In the context of AWS, you can set up replication in several ways:

Amazon RDS Multi-AZ: This provides high availability by replicating your database synchronously across Availability Zones.

Read Replicas: You can create read replicas for read-heavy workloads, improving performance. These can be in different Availability Zones or regions.

3. Distributed Architecture:

Distributed databases are designed to handle large datasets and high transaction volumes by distributing data and processing across multiple servers or nodes.

Amazon Aurora: Aurora is designed to be highly available and fault-tolerant, with a distributed architecture that replicates data six ways, across three Availability Zones.

Amazon Redshift: Redshift is a fully managed data warehouse that uses a distributed, columnar architecture for fast query performance.

4. High Availability:

High availability ensures that your database remains accessible and operational even in the face of failures.

Amazon RDS Multi-AZ: As mentioned earlier, this feature replicates your primary database to a standby instance in another Availability Zone, providing automatic failover in case of an issue.

Amazon Aurora: Aurora offers read and write quorum, ensuring high availability even if some instances fail.

5. Disaster Recovery Strategies in AWS:

Disaster recovery is critical for minimizing downtime and data loss in the event of catastrophic failures or disasters.

Backup and Restore: Regularly backup your database using Amazon RDS automated backups or snapshots. You can restore from these backups in case of a disaster.

AWS Backup: AWS Backup is a centralized backup service that can be used for cross-account and cross-region backup of RDS instances and other AWS resources.

6. Data Encryption and Security:

In addition to these strategies, it's essential to ensure data encryption and security to protect your databases from unauthorized access and data breaches.

Encryption: Use SSL/TLS for data in transit and AWS Key Management Service (KMS) for data at rest encryption.

Security Groups and Network ACLs: Implement proper network security controls to restrict access to your database instances.

7. Monitoring and Alerts:

Utilize AWS CloudWatch and database-specific metrics to monitor the health and performance of your databases. Set up alarms and notifications to proactively respond to issues.

8. Scaling Strategies:

Implement auto-scaling and capacity planning to handle changing workloads effectively.

9. Database Backup and Restore Testing:

Regularly test your backup and restore procedures to ensure they work as expected during a disaster recovery scenario.

10. Compliance and Auditing:

Ensure your database configurations meet industry-specific compliance requirements, and audit database activities for security and compliance.

In conclusion, the working of SQL databases in AWS involves a combination of managed services, replication, distributed architecture, high availability, disaster recovery strategies, and robust security measures.

Comparative analysis of SQL with other databases.

1. SQL Databases:

Strengths:

Structured Data: SQL databases are great for structured data with well-defined schemas. They ensure data integrity and consistency.

Mature Query Language: SQL offers a mature and powerful query language for complex data retrieval and manipulation.

Weaknesses:

Scalability: Traditional SQL databases can face challenges in scaling horizontally to handle large datasets or high traffic loads.

Performance: While optimized for read-heavy workloads, SQL databases may not perform as well for write-heavy or highly concurrent applications.

2. NoSQL Databases:

Strengths:

Scalability: NoSQL databases are highly scalable and can handle large volumes of data and traffic through horizontal scaling.

Weaknesses:

Limited Query Capabilities: Querying in NoSQL databases can be less powerful than SQL, often requiring multiple queries or data processing to achieve the same result.

3. NewSQL Databases:

Strengths:

Scalability: NewSQL databases aim to combine the best of both SQL and NoSQL worlds, providing horizontal scalability while maintaining SQL-like consistency and transactional support.

Performance: They are designed to provide high performance for read and write operations.

Weaknesses:

Complexity: Some NewSQL solutions can be complex to set up and manage, requiring more expertise.

Cost: The complexity and performance benefits of NewSQL solutions may come at a higher cost.

4. Graph Databases:

Strengths:

Relationship Modeling: Graph databases excel in modeling and traversing complex relationships, making them ideal for social networks, recommendation engines, and fraud detection.

Performance for Connected Data: Graph databases are optimized for queries involving connected data nodes.

Weaknesses:

Scalability: Graph databases may face challenges in horizontal scalability for very large graphs.

Conclusion

In this comprehensive report, we've explored the complexities of managing SQL databases within the AWS ecosystem, a vital component of modern database management. AWS offers a range of services like Amazon RDS, Aurora, and Redshift, providing flexible and scalable solutions for various database needs.

Key topics covered include database replication, distributed architecture, high availability, disaster recovery, data encryption, security protocols, monitoring, scaling, backup testing, and compliance considerations, all contributing to a robust SQL database environment on AWS.

Additionally, we've compared SQL databases with other types like NoSQL, NewSQL, and graph databases, highlighting their strengths and weaknesses for informed decision-making.

In summary, AWS enables businesses to effectively deploy, manage, and secure SQL databases, offering flexibility and scalability for mission-critical applications. By implementing best practices and security measures, organizations can ensure data integrity and compliance while leveraging the benefits of AWS services.

Understanding SQL database management nuances and exploring alternative database options are essential for navigating the evolving data management landscape. By adopting these strategies, organizations can harness SQL databases' power in AWS, ensuring data integrity, availability, and security.

References

YouTube Channels

- TechWorld with Nana
- Abhishek Veeramalla
- Tony Teachers Tech
- DevOps & Cloud with Aryya

Websites

- App.diagrams.net
- Amazon.com

Tools

- ChatGPT & Copilot
- Ms Word
- Ms Power Point
- Ms Visio