Assignment 3 Submission

Submission ID: 672b9b655976a1aad44bd251

Submitted On: 2024-11-06 12:37:57

Net ID: jy451478

Answer 1:

Given the needs of an international news agency handling high volumes of multilingual text data, Amazon Aurora with PostgreSQL compatibility is a suitable choice due to its high availability, scalability, and support for advanced text search.

1. High Availability : Aurora automatically creates replicas of data across three Availability Zones (AZs) and provides automated failover to up to 15 Aurora replicas. If no replicas are present, Aurora will attempt to create a new instance in case of a failure, ensuring maximum uptime and reliability for the application.

2. Scalability : Aurora offers seamless storage scaling, automatically increasing from a minimum of 10GB to a maximum of 128TB as needed. This makes it ideal for handling high volumes of multilingual data without performance degradation, enabling the organization to grow its data footprint smoothly.

3. Distributed Access : Aurora?s architecture replicates data across multiple AZs within a single AWS Region, allowing distributed access for teams in different locations with low latency and high data consistency.

4. Advanced Text Search : PostgreSQL?s compatibility provides Aurora with advanced full-text search capabilities via tsvector and tsquery , which are more efficient for complex text searches than MySQL. This feature is particularly valuable for news agencies that rely on fast,

accurate text searching across multilingual datasets.

Answer 2:

To support a read-intensive application with complex queries, a growing user base, and significant data inflow, Amazon Aurora PostgreSQL with a db.r7g.2xlarge instance is recommended at launch, with planned scaling for future growth. Here?s a concise setup and growth strategy:

Instance Type Selection

* Instance Type : Choose db.r7g.2xlarge from the memory-optimized R7 instance family, powered by Graviton3 processors, which offer up to 25% better performance over Graviton2, ideal for handling complex, read-intensive workloads.
* Specifications :
* vCPUs : 8
* Memory : 64 GB
* Network Performance : Up to 10 Gbps

Storage Setup

* Initial Storage : Start with 1TB of Aurora?s SSD-backed storage with automatic scaling enabled, allowing seamless growth up to 128TB as needed.
* Scalability : Aurora?s auto-scaling storage ensures the database grows to meet data demands, handling inflow from 50GB/day to an estimated 73GB/day by the end of the first year.

Scaling Plan

* Read Replicas : Set up Aurora read replicas across multiple Availability Zones to distribute read load, enhance performance, and improve high availability.

* Primary Instance Scaling : As user load increases (from 50,000 to an estimated 200,000 within a year), scale the primary instance to db.r7g.4xlarge and subsequently to db.r7g.8xlarge to maintain performance for complex queries.

Monitoring and Optimization

* AWS RDS Performance Insights : Use Performance Insights to track and optimize query performance and CPU/memory utilization.

* AWS CloudWatch : Enable CloudWatch to monitor resource usage and proactively plan upgrades as thresholds approach.

Answer 3:

The backup and recovery strategy for maintaining at least a week's worth of data, combining RDS automated backups, point-in-time recovery, and cross-region replication for added resilience:

1. Automated Backups

* Enable RDS Automated Backups with a 7-day retention period to ensure daily backups meet data recovery needs. Aurora?s incremental backups optimize storage by only capturing data changes since the last backup.

2. Point-in-Time Recovery (PITR)

* PITR allows for restoring the database to any second within the 7-day window, providing flexibility to recover from accidental deletions or errors at specific points in time.

3. Multi-Region Replication

* Enable Multi-Region Replication to maintain a backup of the database in a secondary AWS region. This enhances disaster recovery by ensuring an up-to-date replica in a different region, supporting both recovery and high availability.

Answer 4:

Encryption at Rest

* Use AWS Key Management Service (KMS) to manage encryption keys for data at rest. KMS provides automated key management and rotation, ensuring all stored data (including backups and replicas) remains encrypted without manual intervention.

Encryption in Transit

* Apply TLS/SSL encryption for all data in transit, ensuring that data is encrypted from the user's device to the database endpoint. This protects sensitive information from interception over the network.

This strategy secures the architecture end-to-end by encrypting data both at rest and in transit, utilizing AWS-managed encryption tools to simplify and automate the process.

Answer 5:

VPC Design Overview

* Create a VPC across 3 Availability Zones (AZs) to ensure high availability and fault tolerance.

Subnet Breakdown

* Public Subnets

* Place 2 public subnets in 2 different AZs . These will host NAT gateways , enabling instances in private subnets to access the internet while blocking inbound internet traffic.

* Private Subnets for EC2

* Place 2 private subnets in 2 different AZs to run EC2 instances for the agency?s application. This ensures high availability and redundancy.

* Private Subnets for Aurora

* Place 2 private subnets in 2 different AZs to run Aurora instances : one for the primary database and the other for a replica . This ensures high availability for the database layer.

-> High Availability : Spreading resources across 2 AZs ensures redundancy and failover, reducing the impact of an AZ failure.

-> Security : Public subnets host NAT gateways to allow controlled internet access for private subnet resources. Private subnets secure sensitive workloads (EC2 and Aurora).

-> Scalability : This design provides scalability for both EC2 and Aurora instances across multiple AZs, handling growing application demands.

Answer 6:

Security Group Configuration for RDS (PostgreSQL):

1. Inbound Rules:

* Allow access on port 5432 for PostgreSQL:

* Source : EC2 Security Group (ensures only EC2 instances in your VPC can access RDS).

* Port : TCP 5432 (PostgreSQL port).

1. Outbound Rules:

* Allow all outbound traffic :

* Destination : 0.0.0.0/0 (allows RDS to access the internet if needed, e.g., for updates or external communication).

Security Group Configuration for EC2 Instances:

1. Inbound Rules:

* Allow HTTP and HTTPS traffic from the internet:

* Source : 0.0.0.0/0 (public access for web applications).

* Ports : TCP 80 (HTTP), TCP 443 (HTTPS).

* Allow inbound traffic from the Elastic Load Balancer (ELB) and access to the RDS database :

* Source : ELB Security Group (for load-balanced traffic).

* Port : TCP 80 (HTTP for ELB).

* Allow port 5432 for PostgreSQL if EC2 needs direct access to RDS (e.g., for administrative tasks).

1. Outbound Rules:

* Allow all outbound traffic to access NAT gateways:

* Destination : 0.0.0.0/0 (allows EC2 to access external resources via the NAT gateway).

* ? Ports : TCP 80, 443 (for HTTP/HTTPS traffic and updates).

Answer 7:

NACLs Configuration:

1. Public Subnet NACLs:

* Inbound Rules :

* Allow HTTP (80) and HTTPS (443) from 0.0.0.0/0 .

* Outbound Rules :

* Allow All traffic to 0.0.0.0/0 .

2. Private Subnet NACLs:

* Inbound Rules :

* Allow traffic from public subnet CIDR (e.g., 10.0.0.0/16) on necessary ports (e.g., TCP 5432 for

PostgreSQL).

* Outbound Rules :

* Allow All traffic to 0.0.0.0/0 via NAT Gateway.

Key Differences:

* Public Subnet : Permits HTTP/HTTPS inbound, all outbound traffic.

* Private Subnet : Restricts inbound to public subnet; allows outbound via NAT Gateway.

Answer 8:

DDoS Mitigation Strategy Using AWS Services:

* AWS Shield:

* Shield Standard : Automatically protects against the most common DDoS attacks (e.g., SYN/ACK floods, DNS query floods) at the network level.

* Shield Advanced : Offers enhanced protection with real-time monitoring, 24/7 DDoS Response Team (DRT), and advanced attack detection to mitigate larger and more sophisticated attacks.

* AWS WAF (Web Application Firewall):

* Protects web applications by filtering malicious HTTP/HTTPS traffic. Custom rules can be configured to block common attack patterns (e.g., SQL injection, cross-site scripting).

* Amazon CloudFront:

* Content Delivery Network (CDN) : Distributes traffic across global edge locations, reducing the impact of DDoS attacks by caching content and preventing backend overload.

* Works with AWS Shield and WAF for enhanced DDoS protection.

Answer 9:

To handle a triple increase in user demand and a ten-fold rise in data inflow, here's an optimized and concise strategy:

1. Auto Scaling:

* Vertical Scaling : Increase EC2 instance sizes (e.g., using more powerful instance types) and scale memory to handle higher workloads.

* Horizontal Scaling : Implement Auto Scaling Groups (ASG) for dynamic scaling based on demand, adding/removing EC2 instances as needed.

2. Caching:

* Amazon ElastiCache : Implement caching for frequently accessed data to reduce database load and improve response times.

3. Database Scaling:

* Amazon RDS with Read Replicas : Use RDS Read Replicas to offload read-heavy workloads, improving performance.

* Amazon Aurora : For write-heavy workloads, use Aurora, which scales automatically with increased storage and maintains high availability.

4. Load Balancing and Traffic Distribution:

* Use Elastic Load Balancer (ELB) to distribute incoming traffic evenly across multiple EC2 instances and ensure high availability.

5. Monitoring and Optimization:

* Use CloudWatch to monitor performance and adjust resources dynamically to meet demand.

Answer 10:

Monitor Key Metrics : Track CPU usage, memory, read/write I/O, and active sessions.

Visual Representation : Provides graphs to identify performance spikes and demand.

Identify Bottlenecks : Detect issues like high CPU consumption, lock waits, and I/O latency.

Pinpoint Problematic SQL : Identify specific SQL statements causing performance issues.

Informed Actions : Use insights to optimize queries, scale instances, or tune resources for optimal performance.

Answer 11:

To ensure data integrity and consistency in an RDS database, we can implement the following strategies:

1. ACID Compliance : Ensure that transactions follow ACID properties (Atomicity, Consistency, Isolation, Durability) for reliable data consistency.
2. Normalization : Apply normalization to reduce data redundancy and maintain integrity within the database.
3. Constraints : Use constraints (e.g., PRIMARY KEY, FOREIGN KEY, UNIQUE) to enforce data integrity rules. If a rule is violated, the transaction is aborted, ensuring data consistency.
4. Automated Backups : Enable automated backups and point-in-time recovery to restore the database to a consistent state during failures.
5. Encryption : Enable encryption at rest and in transit to safeguard data integrity and security.

Answer 12:

For an effective hurricane disaster recovery plan, we can use the following approach:

* Multi-Region Setup : Deploy the RDS database in two geographically distant regions, with one as the primary and the other as a secondary region for failover.
* Cross-Region Replication : Use Amazon RDS read replicas to maintain a live copy of the database in the secondary region. This enables immediate data availability in the event of a primary region failure.
* Automated Backups : Enable automated backups for the primary RDS instance and configure

them to be copied to the secondary region. This ensures data protection and point-in-time recovery across both regions.

Answer 13:

For a robust failover mechanism using RDS Multi-AZ, the following steps ensure minimal service disruption:

* Automatic Multi-AZ Replication : Amazon RDS creates a primary database and maintains a synchronous replica in a different Availability Zone (AZ). During a failure, RDS automatically redirects traffic to the standby instance, eliminating manual intervention and maintaining service continuity.

* Enhanced Multi-AZ with Dual Standbys : To further minimize disruption, we can use an RDS configuration with one primary and two standby replicas. This setup enables automatic failovers within approximately 35 seconds, providing twice the write latency performance compared to a single standby.

References

[1] ?Amazon RDS Aurora Features: High Availability and Durability.?, aws.amazon.com [online]. Available: https://aws.amazon.com/rds/aurora/features/#High_availability_and_durability. [Accessed: July 11, 2024].

[2] ?Full-Text Search in PostgreSQL: A Comprehensive Guide.?, dev.to [online]. Available: https://dev.to/nightbird07/full-text-search-in-postgresql-a-comprehensive-guide-3kcn [Accessed: July 11, 2024].

[3] ?Amazon RDS DB Instance Classes.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.DBInstanceClass.html [Accessed: July 11, 2024].

[4] ?Optimize Costs for Microsoft Workloads on Graviton.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/prescriptive-guidance/latest/optimize-costs-microsoft-workloads/net-graviton.html. [Accessed: July 11, 2024].

[5] ?Amazon RDS Performance Insights.?, aws.amazon.com [online]. Available: https://aws.amazon.com/rds/performance-insights/. [Accessed: July 11, 2024].

[6] ?Point-in-Time Recovery.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/aws-backup/latest/devguide/point-in-time-recovery.html. [Accessed: July 11, 2024].

[7] ?KMS Encryption at Rest.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/network-firewall/latest/developerguide/kms-encryption-at-rest.html. [Accessed: July 11, 2024].

[8] ?VPC NAT Gateway.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html. [Accessed: July 11, 2024].

[9] ?Private Subnets and NAT.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-example-private-subnets-nat.html. [Accessed: July 11, 2024].

[10] ?VPC Network ACLs.?, docs.aws.amazon.com [online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html. [Accessed: July 11, 2024].

[11] ?AWS Shield.?, aws.amazon.com [online]. Available: https://aws.amazon.com/shield/. [Accessed: July 11, 2024].

[12] ?AWS WAF.?, aws.amazon.com [online]. Available: https://aws.amazon.com/waf/. [Accessed: July 11, 2024].

[13] ?Amazon CloudFront.?, aws.amazon.com [online]. Available: https://aws.amazon.com/cloudfront/. [Accessed: July 11, 2024].

[14] ?Amazon RDS Multi-AZ Deployments.?, aws.amazon.com [online]. Available: https://aws.amazon.com/rds/features/multi-az/. [Accessed: July 11, 2024].