Scenario: The organization is an international news agency that gathers and analyses vast amounts of multilingual text data from various sources. They have a globally distributed team of journalists and analysts who need access to the database. Their current database setup, a traditional relational database, is not scaling well with their growing data volume and increasing access demands from different geographic locations. They need a database solution that can provide high performance, high availability, and durability, along with advanced text search capabilities for their Java based application.

Question 1: For an international news agency dealing with high volumes of multilingual text data and requiring advanced text search capabilities for their Java-based application, which Amazon RDS database engine would you choose and why?

Answer: I would recommend using Amazon Aurora with PostgreSQL as the database engine. Here is why:

a. It has robust support for full-text search. It offers features like text indexing, stemming, ranking, and search functions that are essential for efficiently handling and searching through vast amounts of multilingual text data [1].
b. Amazon Aurora provides high availability by replicating the database across multiple Availability Zones (AZs). In the event of a failure in one AZ, Aurora automatically fails over to a healthy AZ, minimizing downtime and data loss. 2 copies of our data are contained in each AZ, with a minimum of 3 AZs. So, 6 copies of our data. Transparently handle the loss of up to 2 copies of data without affecting database write availability and up to 3 copies without affecting read availability.
   Aurora PostgreSQL, you have the following options [6]:

   1. PostgreSQL native logical replication
   2. PostgreSQL logical replication using the pglogical extension
   3. Physical replication with Aurora global database

   Aurora global database is designed for globally distributed applications, allowing a single Aurora database to span up to 5 secondary Regions with up to 90 Aurora replicas. [6]. And it also does It physically replicates your data with no impact on database performance, enables fast local reads with low latency in each Region, and also have managed planned failover and manual unplanned failover from region-wide outages making it highly available across multiple regions as our application is globally distributed.

c. allows us to create read replicas to offload read traffic and automatically load-balances connections across those replicas. This is crucial for handling the increasing access demands from their globally distributed team.
d. Aurora automatically takes continuous backups and replicates data across multiple locations, ensuring data durability. This is crucial for a news agency dealing with critical and constantly evolving data.

   And one more reason why I choose Aurora PostgreSQL that RDS PostgreSQL allows to create 15 additional asynchronous read replicas outside the cluster in addition to the two reader instances thereby scaling up your read capacity to 17 instances [7]. Amazon RDS Read Replicas provide enhanced performance and durability for RDS database (DB) instances. For read heavy database workloads, the additional read replicas provide the option to elastically scale out beyond the capacity constraints of the two readable instances inside the Multi-AZ deployment with two readable standbys. You can create one or more read replicas and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput.

Question 2: Given that the initial user base is around 50,000 with a data inflow of approximately 50GB/day, and you anticipate the user base will grow to approximately 200,000 within a year, with an estimated data inflow growth rate of 10% per quarter. Additionally, the application is read-intensive with frequent complex queries. What instance type and storage capacity would you select for the above selected RDS database at launch, and how would you plan for the projected growth?

Answer: Selecting an appropriate Amazon RDS instance type and storage capacity for the given scenario, we need to consider both the current requirements and the projected growth of the user base and data inflow.

I would recommend starting with the Amazon RDS db.m5 4x large instance type. These instance types offer a good balance of 16 vCPU and 64GiB memory resources and are well-suited for database workloads with complex queries [2]. In future as aurora scales automatically we can auto scale the instance if needed to next generation

Since data inflow is approximately 50GB/day, so we can assume that approx. 2000 GB/month with 10% on top of it. So, that comes to 25 Tb per year on average. Its PostgreSQL-compatible database engines are customized to take advantage of that fast distributed storage. The underlying storage grows automatically as needed. An Aurora cluster volume can grow to a maximum size of 128 tebibytes (TiB) [8]. Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration [8].

Question 3: How would you devise a backup and recovery strategy capable of restoring at least a week's worth of data?

Aurora backs up cluster volume automatically and retains restore data for the length of the backup retention period. Aurora automated backups are continuous and incremental, so we can quickly restore to any point within the backup retention period. No performance impact or interruption of database service occurs as backup data is being written. we can specify a backup retention period from 1–35 days when you create or modify a DB cluster. Aurora automated backups are stored in Amazon S3 [9].

Amazon RDS creates a storage volume snapshot of your DB cluster, backing up the entire DB cluster and not just individual databases. Because the snapshot includes the entire storage volume, the size of files, such as temporary files, also affects the amount of time it takes to create the snapshot. Unlike automated backups, manual snapshots aren't subject to the backup retention period. Snapshots don't expire. For very long-term backups, exporting snapshot data to Amazon S3 is recommended. Moreover, we can make use of event bridges to trigger a serverless lambda function to take snapshots at regular intervals and once a snapshot is created or if any event fails, we can use SQS and SNS topics to get alerts for ourselves so that we can ensure data is not affected.

Question 4: The organization requires a secure architecture that provides data encryption and access control. What would be your strategy?

Amazon Aurora encrypted DB clusters use the industry standard AES-256 encryption algorithm to encrypt data on the server that hosts Amazon Aurora DB clusters [10]. After data is encrypted, Amazon

Aurora handles authentication of access and decryption of data transparently with a minimal impact on performance. Data that is in transit between the source and the read replicas is encrypted, even when replicating across AWS Regions [10]. Amazon Aurora uses an AWS KMS key to encrypt resources. Using AWS KMS, we can create customer managed keys and define the policies that control how these customer managed keys can be used. we can rotate the key in accordance with your own policies, destroy the key, and restrict access to the key using KMS policies and IAM policies by utilizing a KMS key in AWS KMS.

Using Identity Access Management (IAM), we can control who can be authenticated (signed in) and authorized (have permissions) to use Amazon RDS resources. An IAM user is an identity within AWS account that has specific permissions for a single person or application. Give users and roles only the essential permissions by adhering to the principle of least privilege.

Scenario: With the basic database infrastructure in place, the news agency now wants to ensure the robustness and security of its network. The agency seeks a network configuration that optimally balances accessibility and security. For their reporting work, they've started running certain tasks on EC2 instances, which require secure internet access.

Question 1: Design a Virtual Private Cloud (VPC) for the agency's AWS infrastructure. Specifically, how many subnets would you create, in which Availability Zones (AZs), and why?

Choosing a single VPC is simple and cost-effective by selecting an appropriate IP address range for the VPC that is large enough to accommodate the expected growth of resources. Create two public and two private subnets, each in a different Availability Zone (AZ). Public subnets are associated with a route table that has a default route pointing to an Internet Gateway (IGW). Private subnets are associated with a route table that does not have a route to the IGW. This ensures that resources in private subnets are not exposed to the public internet. Private subnets can host RDS database servers, application servers, and other sensitive backend services. Place a Network Address Translation (NAT) Gateway in a public subnet to allow EC2 instances/services in the private subnets to initiate outbound connections to the internet. Define security groups for EC2 instances to control inbound and outbound traffic at the instance level. Place an Elastic Load Balancer in the public subnets to distribute incoming traffic to the EC2 instances running reporting tasks, enhancing availability and fault tolerance. Implement AWS CloudWatch and AWS CloudTrail for monitoring and logging to keep an eye on your network's security and performance.

Question 2: How would you configure Security Groups for your RDS instances and other EC2 instances that need access to the internet, considering both inbound and outbound rules?

RDS typically have two Security Groups (SG) associated with them: one for the RDS instance and another for EC2 instances that need to access the database. Including only that IP address in the security group rather than the entire range of that IP if access must be granted for a specific server.

Security Groups for RDS Instances:

RDS Security Group (Database SG):

> Inbound Rules: Allow inbound traffic only from the EC2 instances that need to access the database by specifying the source Security Group ID to make this rule more secure. Allowing inbound traffic from other necessary resources, such as application servers or services, based on their specific IP addresses or Security Group IDs.

Outbound Rules: Allow all outbound traffic connections to the internet. RDS instances need to communicate with other AWS services or making HTTPs requests to external APIs for updates or backups.

EC2 Security Group (Application Security Group):

Inbound Rules: For EC2 instances web servers, we have the inbound rules to allow traffic from all over the internet from any device connected to the internet. creating an inbound rule allowing incoming HTTP (port 80) or HTTPS (port 443) traffic from any source (0.0.0.0/0).

Outbound Rules: By default, EC2 instances are allowed to initiate outbound connections to the internet.

Question 3: Discuss the rules you would set for Network Access Control Lists (NACLs) at the subnet level to provide an additional layer of security. How would these rules differ for public and private subnets?

Network Access Control Lists (NACLs) serve as an added layer of security within a Virtual Private Cloud (VPC), operating as stateless firewalls to manage both incoming and outgoing network traffic according to specific rules [12]. These rules differ for public and private subnets based on the requirements of the applications and the data flow.

For Public Subnet NACL Inbound Rules:

In a public subnet, allowing incoming traffic from the internet for specific ports required by essential services, such as HTTPS (port 443) and SSH (port 22). Additionally, configure rules to specify which services or CIDR blocks can access instances on specific ports.

For Public Subnet NACL Outbound Rules:

In a public subnet, enable outbound traffic from the public subnet to a Network Address Translation (NAT) gateway. The NAT gateway assigns an elastic public IP to the instance, concealing the original IP. It then forwards the request to an internet gateway, using the NAT Gateway's elastic IP address to access the internet. If the instance is accessed by other AWS services, configure outbound rules for specific ports required by those services, such as HTTPS (port 443) and SSH (port 22). Furthermore, establish rules for other services or CIDR blocks that need to access the instance on specific ports.

For Private Subnet NACL Inbound Rules:

In a private subnet, restrict incoming traffic from the internet by denying all incoming traffic from external sources. Close all ports except for those that can be accessed by other AWS services. Block incoming traffic from unwanted sources and permit only essential access.

Question 4: As a proactive measure, the news agency wants to minimize the impact of a potential Distributed Denial of Service (DDoS) attack. Suggest a mitigation strategy using AWS services. How would this protect your VPC and the resources within it?

Making use of Shield Standard, a managed DDoS protection service that safeguards applications running on AWS. AWS Shield Standard defends against the most common, frequently occurring network and transport layer (Layer 3 and 4) DDoS attacks to maximize the availability of AWS services [11]. Resources, such as EC2 instances and RDS databases, are automatically guarded against volumetric, state-exhaustion, and some application layer threats when you use AWS Shield Standard. For customized protection against sophisticated (Layer 3 to 7) threats targeting application, we can subscribe to AWS Shield Advanced [11]. It provides more sensitive detection and tailored mitigations against large and complex DDoS attacks, near real-time visibility into attacks, firewall for defence against Layer 7 attacks as well as gives you 24-7 access to the AWS Shield Response Team (SRT).

Scenario: A major world event leads to a sudden surge in user demand, with the number of users expected to triple, and a ten-fold increase in data inflow rate.

Question 1: In light of a major world event, user demand and data inflow are projected to increase significantly (triple the user base and a ten-fold increase in data inflow). How would you modify the architecture to accommodate this surge?

To accommodate the surge in user demand and data inflow, several modifications to needs to be done to the existing architecture:

a. Consider increasing the capacity of infrastructure from existing db.r5.4x large to db.r5.12xlarge or higher (if available at that point of time) [1]. resources can vertically scale servers by upgrading CPU, memory, and storage, or horizontally scale by adding more instances; can leverage Auto Scaling to automatically adjust resources based on demand.
b. Implementing caching mechanisms to reduce the load on the database. Caching can help serve frequently accessed data without repeatedly hitting the database, thus improving response times, and reducing database load.
c. Using Content Delivery Networks (CDNs) to distribute and serve static content like images, videos, files, and many more to users from edge locations closer to them. This can reduce the load on central infrastructure.
d. Optimizing database for high read and write workloads. This may involve using read replicas, thus diverting read traffic from the primary database.
e. By implementing load balancers to evenly distribute incoming traffic across multiple server instances. This ensures high availability and load distribution.
f. Deploying the RDS instance in a multi-AZ configuration for high availability and disaster recovery. This involves maintaining a standby replica in a different Availability Zone, allowing for automatic failover in case of primary database unavailability.
g. Implementing redundancy and disaster recovery strategies to ensure business continuity in case of failures or disruptions.

Question 2: How would you utilize RDS Performance Insights to monitor your database performance and identify bottlenecks?

To monitor database performance and identify bottlenecks using Amazon RDS Performance Insights:

Amazon RDS Performance Insights is a database performance tuning and monitoring feature that helps in quickly detecting performance problems with an easy-to-understand dashboard that visualizes database load and determine when and where to take action [4]. It uses lightweight data collection methods that don't impact the performance of applications and makes it easy to see which SQL statements are causing the load, and why [4]. Look for red bar/spikes in database load, high wait times, or resource contention. These can be indicative of performance bottlenecks. Focus on the wait events to understand what is causing delays. Common culprits include high CPU usage, I/O latency, or locking issues [5]. Enable Performance Insights on RDS instance, by monitoring performance metrics of common culprits and helps to monitor performance them closely to identify and optimize complex queries. And this helps to analyze query execution plans in further improving complex query performance.

Question 3: When you design your RDS database architecture, how would you ensure data integrity and consistency?

To ensure data integrity and consistency, Implementing transactions to ensure that a series of database operations either all succeed, or all fail. Ensure that the transactions follow ACID properties. This helps maintain data consistency. By enforcing data integrity constraints like primary keys, foreign keys, and unique constraints. These constraints prevent invalid or inconsistent data from being entered into the database. By Implementing database replication across Multi-AZ's and failover strategies to ensure data availability and consistency in case of hardware failures or other disasters. By encrypting data at rest as well as while in transit will ensure that data is not changed that ensuring data integrity as well as consistency. Leverage the database's built-in features like constraints and triggers to enforce complex business rules and maintain data consistency. Constraints ensure that data meets specific conditions and prevent the introduction of inconsistent or invalid data.

Scenario: A hurricane threatens the location of one of your major data centres. Your architecture must withstand the catastrophe and provide reliable disaster recovery capabilities.

Question 1: A hurricane threatens the location of one of your major data centres. Create a disaster recovery plan that includes replication of the RDS database across regions. How would you incorporate your plan on this?

Answer: Disaster recovery planning for a major data centre facing a hurricane is crucial to ensure minimal downtime and data loss. Here's how you can create a disaster recovery plan that incorporates replication of the RDS database across regions and utilizes AWS RDS Multi-AZ deployments for minimal service disruption:

a. Database Replication Across Regions: Utilizing AWS RDS Multi-AZ with two readable standbys deployments to replicate primary database to a secondary region that is geographically distant from the primary data centre. This ensures data redundancy and minimizes the risk of data loss. In the event of a failure, one of the two remaining standbys will take over and serve the workload
b. Setting up a DNS failover mechanism to route traffic to the secondary region in case of a disaster. we can use Amazon Route 53 to automate the failover process based on health checks.
c. Periodically testing the failover process to ensure it works as expected. Simulating disaster scenarios to make sure recovery plan is reliable.

Question 2: There is a need to develop a failover mechanism using AWS RDS Multi-AZ deployments to ensure minimal downtime during disaster recovery scenarios. How does your mechanism guarantee minimal service disruption?

Answer:

a. Configuring RDS instance as a Multi-AZ deployment. In Multi-AZ mode, AWS automatically replicates primary database to a standby instance in a different Availability Zone (AZ) within the same region or we can choose different region.
b. With Multi-AZ, if the primary database instance becomes unavailable due to hardware failure or maintenance, AWS will automatically promote the standby instance to become the primary, ensuring minimal downtime.

c. By utilizing read replicas for offloading/diverting read traffic from primary database. This can improve performance and reduce the load on the primary database instance.
d. Periodically testing backup, taking manual DB snapshots of our RDS & storing them in s3 bucket of cross region and restore process to ensure recovery in case of data corruption or other issues.
e. Setting up health checks and CloudWatch alarms to monitor the status of RDS instances. Defining thresholds and actions to be taken in case of any issues will trigger an alarm if anything happens not as intended.
f. Ensure the team is trained and prepared to execute the failover process where team can follow detailed runbooks that outline the steps to take in case of a disaster.

Automated backups are limited to a single AWS Region while manual snapshots and Read Replicas are supported across multiple Regions [3]. So, we are choosing read replicas for disaster recovery.

References

[1] *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/dms/latest/sql-server-to-aurora-postgresql-migration-playbook/chap-sql-server-aurora-pg.tsql.fulltextsearch.html#chap-sql-server-aurora-pg.tsql.fulltextsearch.pg. [Accessed: 03-Nov-2023].

[2] *Amazon.com*. [Online]. Available: https://aws.amazon.com/rds/instance-types/. [Accessed: 03-Nov-2023].

[3] *Amazon.com*. [Online]. Available: https://aws.amazon.com/blogs/database/implementing-a-disaster-recovery-strategy-with-amazon-rds/. [Accessed: 03-Nov-2023].

[4] *Amazon.com*. [Online]. Available: https://aws.amazon.com/rds/performance-insights/. [Accessed: 03-Nov-2023].

[5] *Amazon.com*. [Online]. Available: https://aws.amazon.com/blogs/database/analyzing-amazon-rds-database-workload-with-performance-insights/#:~:text=Performance%20Insights%20quickly%20and%20easily,by%20modifying%20the%20database%20instance. [Accessed: 03-Nov-2023].

[6] *Amazon.com*. [Online]. Available: https://aws.amazon.com/blogs/database/understand-replication-capabilities-in-amazon-aurora-postgresql/. [Accessed: 03-Nov-2023].

[7] *Amazon.com*. [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2023/05/amazon-rds-postgresql-15-read-replicas-multi-az-readable-instances/. [Accessed: 03-Nov-2023].

[8] "AWS Aurora PostgreSQL," *Massdriver.cloud*. [Online]. Available: https://www.massdriver.cloud/marketplace/aws-aurora-postgresql. [Accessed: 03-Nov-2023].

[9] *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html. [Accessed: 03-Nov-2023].

[10] *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Overview.Encryption.html. [Accessed: 03-Nov-2023].

[11] [Online]. Available: http://file:///C:/Users/Dell/Desktop/SankeerthAssignment/Assignment/A-3/5902-RDS-Scenario.pdf. [Accessed: 03-Nov-2023].

[12] Amazon.com. [Online]. Available: https://docs.aws.amazon.com/managedservices/latest/userguide/restrict-nacl.html. [Accessed: 03-Nov-2023].