Task 4: Backup and Disaster Recovery Plan

Selecting a Disaster Recovery Strategy for Our Critical Healthcare Business: Based on our requirement to ensure **minimum downtime and data loss**, I have shortlisted two disaster recovery strategies:

a. Elastic Disaster Recovery (Elastic DR): Elastic DR is an AWS-managed service that enables rapid recovery of applications and data by continuously replicating workloads to a secondary AWS region. In the event of a disaster, Elastic DR automates the failover process, allowing services to be restored within a predefined Recovery Time Objective (RTO) and Recovery Point Objective (RPO).

b. Active/Active Configuration: An Active/Active setup involves running duplicate production environments simultaneously across multiple AWS regions. Both environments actively handle traffic, ensuring high availability and instantaneous failover without service interruption, thereby minimizing both downtime and data loss.

Criteria	Elastic Disaster Recovery (Elastic DR)	Active/Active Configuration
Recovery Time Objective (RTO)	≤ 15 minutes	≤ 1 minute
Recovery Point Objective (RPO)	≤ 5 minutes	≤ 1 minute
Cost	Lower cost due to single active environment and pay-as-you-go replication	Higher cost due to maintaining dual active environments
Complexity	Moderate setup with automated failover processes	High complexity in synchronizing data and managing dual environments
Maintenance	Simplified with centralized management	Increased maintenance overhead

Active/Active Configuration is the recommended disaster recovery strategy for our critical healthcare platform. This strategy uses the geolocation routing, which looks at the IP address of the request to determine where it is coming from. If we add health checks, if an end point fails, then would roll up to find the next closest.

This decision is made by considering several key factors that help achieve our requirements for minimal downtime and data loss:

1. **RTO ≤ 1 Minute:** The Active/Active setup makes sure that both production environments are live and capable of handling traffic simultaneously. In the event of a failure in one region, traffic is instantly rerouted to the other without any noticeable downtime. This should meet our critical RTO requirement of ≤ 1 minute.
2. **RPO ≤ 1 Minute:** Real-time data synchronization between the active environments makes sure that data loss is almost none. Any changes made in one environment are immediately reflected in the other, so this gives us an RPO target of ≤ 1 minute.
3. Both environments are operational at all times, providing continuous availability of our healthcare data. So we get uninterrupted access to patient records, treatment plans, and other critical healthcare data. Because we have an active connection, our services will remain available even in case of maintenance or outage.

4. Active/Active configurations uses AWS Elastic Load Balancing (ELB) and Amazon Route 53 to distribute traffic this ensures that the resource utilization properly on both environments. We can even scale our resources based on the demand.

Alternative Strategy: If the **cost** of maintaining an Active/Active Configuration becomes a constraint, I recommend adopting **Elastic Disaster Recovery (Elastic DR)**. Elastic DR offers a cost-effective solution with an RTO of ≤ 15 minutes and an RPO of ≤ 5 minutes, still ensuring robust disaster recovery capabilities while optimizing expenses.

1. Backup Strategy Implementation and Steps with Elastic DR

Step 1.1: Set Up AWS Backup

1. Navigate to the AWS Backup service console and Create Backup Plan: Either choose a pre-built backup plan or customize a plan as per the needs of the organisation.
2. We can then configure resources by adding the resources we want to back up (RDS databases, EBS volumes, DynamoDB tables).
3. Configure backup retention policies.
4. Set Backup Frequency:
   Full Backups: Weekly (e.g., every Sunday at 2 AM UTC) Scope: Capture the entire dataset, including databases, application servers, and storage systems.
   Incremental Backups: Daily (e.g., every night at 2 AM UTC) Scope: Backup only the data that has changed since the last backup, reducing storage requirements and backup time.
   Differential Backups: Twice daily (e.g., every 6 AM and 6 PM UTC) Scope: Backup data changes since the last full backup, providing a balance between full and incremental backups.
5. Cross-Region Backup: We then enable cross-region backup replication to replicate backups across multiple AWS regions.

Step 1.2: Backup to Amazon S3

1. Create an S3 Bucket. By going to the S3 console and create a new S3 bucket. Enable versioning and encryption to secure the data.
2. Enable Cross-Region Replication: In the S3 bucket settings, we enable cross-region replication to replicate the data to another AWS region.
3. Set Up Lifecycle Policies: Create lifecycle policies to automatically transition older backups to cheaper storage classes like S3 Glacier.

Step 1.3: Automate EC2 Snapshots

1. Navigate to the EC2 console and create Snapshot Automation by using AWS Backup or AWS Data Lifecycle Manager to automate snapshots of EBS volumes attached to the EC2 instances. Schedule daily snapshots for EC2 instances.
2. Cross-Region Replication: Use AWS Backup to replicate snapshots across regions.

2. Redundancy Setup

Step 2.1: Enable RDS Multi-AZ

1. Go to the RDS Console and navigate to Amazon RDS and then enable Multi-AZ Deployment. This will automatically create a standby database in another Availability Zone.

Step 2.2: Configure Cross-Region Replication

1. Go to the S3 console and configure replication rules in the bucket's management tab.Choose the destination region and destination bucket. Enable replication for all objects in the bucket.

Step 2.3: Use AWS Elastic Disaster Recovery (AWS DRS)

1. Go to AWS Elastic Disaster Recovery service and set Up Source Servers. Select critical EC2 instances or workloads to replicate to another region. Configure replication settings to the target region.

3. Recovery Time Objective (RTO) and Recovery Point Objective (RPO) Setup

Step 3.1: Define RTO and RPO for Each Resource

1. List all critical resources such as RDS databases, EC2 instances, and S3 buckets. Define RTO (how quickly recovery must happen) and RPO (maximum acceptable data loss window).

Step 3.2: Automate Recovery Processes

1. Create CloudFormation templates for automating the recovery process. Templates should define infrastructure (EC2, VPCs, RDS) and automate redeployment in another region.
2. Use AWS Lambda for Failover Automation: Create Lambda functions to trigger failover to the disaster recovery region automatically when CloudWatch detects an outage.

Step 3.3: Continuous Database Backup for RDS

1. Enable Point-In-Time Recovery (PITR): In the RDS console, enable continuous backups with PITR for critical databases to achieve near-zero RPO.
2. Set up retention policies to keep PITR backups for a defined period (in our case we select 30 days).

4. AWS Services Configuration

Step 4.1: Configure AWS Backup

1. Automate Backup Jobs: In AWS Backup, configure automated backup jobs for EC2 instances, EBS volumes, RDS databases, and DynamoDB tables.
2. Use the backup plans we defined before, enable Cross-Region Replication and ensure that AWS Backup is replicating the backups to another region.

Step 4.2: Set Up S3 for Data Backup

1. Create S3 Buckets with Versioning and Encryption: Ensure that all backup data stored in S3 is versioned and encrypted using AWS KMS (Key Management Service).
2. Enable Glacier Archiving: Set lifecycle policies to automatically move backups older than a certain period (e.g., 30 days) to Amazon Glacier for long-term storage. This is very crucial in our large growing database.
3. Periodically test EC2 instance restoration by launching a new instance from a snapshot to make sure that the process working.

5. Monitoring, Testing, and Documentation

Step 5.1: Set Up Monitoring with CloudWatch

1. Enable CloudWatch Alarms: Set up alarms to monitor the health of your backups, EC2 instances, RDS databases, and any other critical infrastructure. Integrate with AWS SNS to notify the operations team when any backup or disaster recovery issue arises.

Step 5.2: Test Disaster Recovery Quarterly: Simulate disaster recovery events on a regular basis (quarterly). Use AWS Elastic Disaster Recovery to perform failover drills in a non-production environment.

Step 5.3: Use AWS Trusted Advisor

We could even use AWS Trusted Advisor to get recommendations on backup health, security, and cost optimization. If we regularly review the suggestions we can get an overview of whether the backup and recovery plan is aligned with AWS best practices or not.