***AWS TASKS:***

***TASK-01***

Auto Scaling based on CPU Utilization with EC2 Auto Scaling

\* Set up EC2 Auto Scaling to automatically adjust the number of instances based on CPU utilization metrics.

\* Configure CloudWatch Alarms to trigger scaling actions in real-time when CPU thresholds are exceeded.

\* Monitor auto scaling events and instances in real-time to ensure optimal resource utilization and application performance

Database Replication and Failover with RDS Multi-AZ

\* Configure RDS Multi-AZ deployment for automatic synchronous replication of your database to a standby instance.

\* Monitor Multi-AZ replication status and database health metrics in real-time.

\* Trigger automatic failover to the standby instance in case of primary database failure.

Log Analysis with CloudWatch Logs Insights

\* Utilize CloudWatch Logs Insights to analyze log data from various AWS services in real-time.

\* Create custom log queries to identify trends, errors, or patterns in logs for troubleshooting or security analysis.

\* Monitor log analysis results in real-time and set up automated actions or alerts based on specific log query results.

Network Load Balancing with Application Load Balancer and Target Tracking Scaling

\* Configure an Application Load Balancer (ALB) to distribute traffic across multiple EC2 instances in real-time.

\* Set up target tracking scaling with ALB to automatically adjust the number of instances based on the target's average request count or CPU utilization.

\* Monitor ALB metrics and target tracking scaling activities in real-time for efficient traffic management and optimal application performance.

Network Traffic Monitoring with VPC Flow Logs and CloudWatch Logs

\* Enable VPC Flow Logs to capture and log network traffic metadata at the subnet or ENI level.

\* Stream VPC Flow Logs to CloudWatch Logs for real-time analysis, monitoring, and troubleshooting.

\* Set up real-time log alerts or use CloudWatch Logs Insights to identify network traffic anomalies or security threats.

Volume Monitoring with CloudWatch Metrics and Alarms

\* Enable CloudWatch Metrics for EBS volumes to collect real-time performance metrics such as throughput and latency.

\* Set up CloudWatch Alarms to trigger notifications or actions when volume performance metrics exceed predefined thresholds.

\* Monitor volume metrics and alarms in real-time to identify and address performance bottlenecks or capacity issues.

Instance Monitoring and Performance Analysis with EC2 Systems Manager

\* Utilize EC2 Systems Manager to collect real-time metrics and logs from instances.

\* Set up CloudWatch Dashboards to monitor CPU, memory, disk utilization, and network traffic in real-time.

\* Configure CloudWatch Alarms to trigger notifications when performance thresholds are breached.

Access Control with IAM Roles and Instance Metadata

\* Create IAM roles with time-limited permissions for EC2 instances.

\* Use instance metadata to retrieve temporary credentials for real-time access to AWS services.

\* Monitor and rotate IAM role credentials regularly to ensure security and minimize the risk of unauthorized access.

Create a mechanism which should send me notification over email if there is change in IAM policy, role.

Create a mechanism which should send me notification over email if there is any change in s3 service like creating bucket, any change in bucket policy etc.

***TASK-02***

1. Real-Time Database High Availability and Failover with RDS Multi-AZ, Read Replicas, and Amazon Aurora Global Database

Configure RDS Multi-AZ deployment for real-time synchronous replication and automatic failover for high availability.

Implement RDS Read Replicas or Amazon Aurora Global Database for real-time replica creation and global data replication for read scalability and disaster recovery.

Monitor replication lag, failover events, and replica performance in real-time to ensure data consistency and minimal downtime.

2. Real-Time Database Performance Monitoring and Optimization with RDS Performance Insights and Enhanced Monitoring

Enable RDS Performance Insights and Enhanced Monitoring for real-time collection and analysis of database performance metrics.

Utilize query-level and instance-level performance data to identify and optimize database bottlenecks in real-time.

Implement automated actions and alarms based on performance thresholds to proactively maintain optimal database performance.

3. VPC Peering with Multi-Account Architecture

Establish VPC peering connections between VPCs in different AWS accounts to create a multi-account architecture.

Implement cross-account access controls using IAM roles and policies to govern communication between the peered VPCs.

Monitor the peering connections and enforce centralized visibility and control across the multi-account environment.

4. VPC Peering with Transit Gateway

Create a transit gateway and establish VPC peering connections with multiple VPCs.

Configure route propagation and route tables to enable seamless communication between the peered VPCs and on-premises networks.

Monitor the transit gateway and peering connections for any issues and perform troubleshooting as necessary.

5. Real-Time User Authentication and Authorization with AWS Single Sign-On (SSO)

Set up AWS Single Sign-On (SSO) to manage user authentication and authorization across multiple AWS accounts and applications.

Configure real-time role-based access policies in SSO to grant or revoke access based on user attributes or group membership.

Monitor SSO logs and events to detect and respond to any unauthorized access attempts in real-time.

***TASK-03***

1. Create 2 private 2 public subnet and instead of using NAT gateway use NAT instance, and implement NLB in that Enable Proxy Protocol also Configure Session Affinity and Enable Connection Idle Timeout and Implement Target Group Stickiness

2. Create 2 private 2 public subnet and instead of using NAT gateway use NAT instance, and implement NLB in that Enable Proxy Protocol and Implement Target Group Stickiness

3. Enable ALB logs and store in S3 Bucket

4. Enable AS with warmup time and dynamic scaling policy and implement ins on it

5. Implement Cross-Account NLB and enable access logs and store logs in s3 in account A

6. Configure event notifications on your S3 buckets to trigger actions when specific events occur, such as object creation, deletion, or restoration. Set up cross-account access to allow other AWS accounts or IAM users to access your S3 buckets

7. Create a mechanism to get the CPU utilisation and Free Memory of system on email. I should get alert every hour

8. Copy EBS snapshots to different regions for cross-region disaster recovery and data migration. This helps protect your data and provides geographical redundancy.

9. Enable CloudFront Access Logs and in origin use loadbalancer and also Implement Geo-Restrictions and also implement ssl using ACM

10. Set Up CloudFront Real-Time Logs and use s3 as origin , and Use signed URLs and cookies to control access to your content and grant temporary permissions for specific users or time periods.

11. I want to access my s3 bucket form ec2 instance which is not associated with nor NAT or IGW

12. Create a mechanism if my ec2 instance’s CPU or disk usage is 90% or above I should get the notification on email.

13. Host a 2 service in single server which is Nginx and Httpd perform this in ubuntu and amazon linux

14. Host a static website using nginx but change the root dir to /mnt perform this in ubuntu and amazon linux

15. Perform reverse proxy on nginx

***Task04***

1. Schedule start and stop ec2 instance using lambda
2. Schedule snapshot of ec2 instance using lambda
3. Filter out running instance which is running more than one hour and it should sent notification on mail
4. Three-tier architecture (it should be in private)
5. Host student.war in elastic bean stock
6. Host nginx services and attach SSL certificate on it.

***TASK 05:***

1. Restore deleted objects from s3 bucket
2. Create IAM user who should have only cli access with the help of thet user host s3 static bucket
3. Use an intelligent-tiering to automatically move objects between storage class based on access patterns
4. Set up lifecycle policy to automatically delete expired objects
5. S3 object multipart storage upload
6. Generate pre-signed url for temporary access to s3 objects
7. Enable server access logging in s3 bucket
8. S3 lambda DynamoDB integration
9. Create student information excel sheet

Column: std name, age, education, fee, course

Import this data into DynamoDB and vice versa

1. Explore vpc endpoint and integrate it with DynamoDB
2. Difference between gateway and interface?