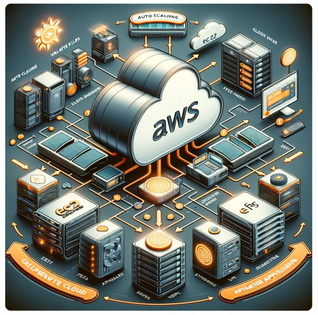

Capstone Project-1

1. Project Title — Resilient and Scalable Web Application Deployment on AWS
2. Project Description
   1. This project involves designing and implementing a highly available and scalable web application infrastructure on AWS.
   2. The architecture will leverage AWS services to ensure
      1. Fault tolerance
      2. Load balancing
      3. Secure user access
   3. The core of the project is to deploy a web application that can handle varying loads efficiently and maintain high availability across multiple Availability Zones (AZs).
3. Objectives
   1. High Availability — Achieve minimal downtime for the web application by utilizing multiple Availability Zones.
   2. Scalability — Use AWS Auto Scaling to adjust resources automatically in response to traffic changes, ensuring efficient performance.
   3. Security — Implement security measures focusing on security groups and secure communication.
   4. Resilience — Develop a resilient application setup that can withstand failures and traffic spikes without manual intervention.
4. Core AWS Services Utilization
   1. Virtual Private Cloud (VPC) — Set up a custom VPC to provide a isolated network environment. This VPC will have public and private subnets across different AZs for enhanced security and availability.
   2. Elastic File System (EFS) — Leverage EFS for scalable file storage, which can be concurrently accessed by instances for storing shared application data.
   3. Elastic Compute Cloud (EC2)
      1. Utilize EC2 instances to host the web application.
      2. These instances will serve as the compute resources running the application, benefiting from AWS's secure, resizable compute capacity.
   4. AWS Auto Scaling — Configure Auto Scaling to dynamically adjust the number of EC2 instances, ensuring that the application scales efficiently with demand.pic
   5. Application Load Balancer (ALB) — Utilize an ALB to distribute incoming traffic across multiple EC2 instances in different AZs, enhancing the fault tolerance and availability of the application.
   6. Route 53 — Employ Route 53 for domain management and to route end-user requests to the application in a reliable and cost-effective manner.
5. Project Phases:
   1. Design Phase — Architect the solution, focusing on the application's
      1. Security
      2. Scalability
      3. Availability requirements
   2. Implementation Phase
      1. Create the VPC, subnets, and security groups
      2. Configure EFS
      3. Setup Custom AMI For Auto Scaling
      4. Set up and test Auto Scaling
      5. Deploy the ALB
      6. Integrate Route 53 for domain management
   3. Testing and Optimization Phase — Conduct functional and load testing to ensure the application's performance and scalability meet requirements.
   4. Documentation Phase — Produce detailed documentation covering the architecture, configuration, and deployment process.
6. Deliverables
   1. Architectural diagrams and design documentation
   2. Implementation and configuration guide
   3. Performance and optimization report
   4. A comprehensive project presentation detailing the deployment strategy, encountered challenges, and solutions