

Architecture Overview

The 3-tier architecture consists of three layers:

1) Web Tier (Presentation Layer): This layer is hosted on a public subnet and is responsible for handling user interactions. It typically includes:

- **AWS ECS (Elastic Container Service):** Manages the deployment and scaling of containerized applications.
- **Application Load Balancer (ALB):** Distributes incoming traffic across multiple ECS instances, ensuring high availability and fault tolerance.

2) Logic Tier (Application Layer): Located in a private subnet, this tier processes business logic and communicates with the web tier and database tier. It includes:

- **AWS ECS:** Similar to the web tier, it runs application containers that handle requests from the web tier.
- **Auto Scaling Group:** Automatically adjusts the number of ECS instances based on traffic demand, ensuring optimal performance during peak loads.

3) Database Tier (Data Layer): This layer is also in a private subnet and manages data storage. It typically includes:

- **Amazon RDS (Relational Database Service):** Provides scalable relational database management, ensuring data durability and availability.

Components Involved:

- **Public Subnet:** Hosts resources that need to be accessible from the internet, such as the web tier.
- **Private Subnet:** Contains resources that should not be directly accessible from the internet, such as the logic and database tiers.
- **Route Table:** Manages how traffic is directed between subnets and to/from the internet.
- **Application Load Balancer:** Ensures even distribution of incoming application traffic across multiple targets in the web tier.
- **Auto Scaling Group:** Monitors application performance and scales instances up or down based on predefined metrics.

Handling Increased Traffic

To manage increased traffic effectively:

- The Application Load Balancer distributes incoming requests across multiple instances in the web tier, preventing any single instance from becoming a bottleneck.
- The Auto Scaling Group can automatically increase the number of ECS instances in response to high traffic loads, ensuring that application performance remains stable.
- Caching strategies can be implemented at various levels (e.g., using Amazon ElastiCache) to reduce load on the database by serving frequently requested data from memory.

Handling Failure Scenarios

In case of failures:

- The Application Load Balancer can reroute traffic away from unhealthy instances to healthy ones, maintaining service availability.
- The Auto Scaling Group can replace failed instances automatically, ensuring that the desired number of healthy instances is maintained.
- Amazon RDS offers features like automated backups and Multi-AZ deployments to ensure data durability and availability even during database failures.

This architecture provides a robust framework for deploying scalable applications while ensuring high availability and resilience against failures.