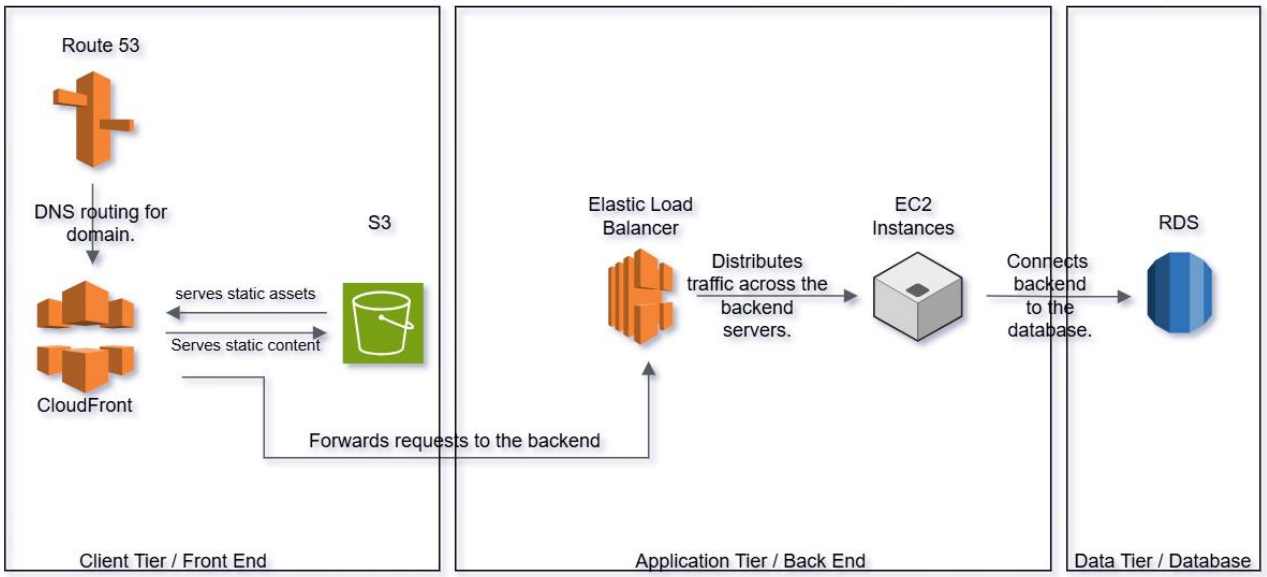
3-Tier AWS Architecture for Portfolio Website



Overview: This architecture diagram outlines a 3-tier AWS infrastructure designed to support a portfolio website built with React for the frontend and Django for the backend. The diagram demonstrates how various AWS services integrate to provide a scalable, secure, and efficient deployment environment.

1. Client Tier / Front End:

- **Amazon Route 53:** This service handles DNS routing for the domain name, directing user traffic to the appropriate AWS resources.
- **Amazon CloudFront:** A content delivery network (CDN) that caches and distributes static assets, such as HTML, CSS, and JavaScript files, from Amazon S3 to users, ensuring faster load times and improved performance.
- **Amazon S3:** Stores static content, including the website's frontend files. S3 provides durable and scalable storage for these assets.

2. Application Tier / Back End:

- **Elastic Load Balancer (ELB):** Distributes incoming traffic across multiple EC2 instances to ensure even load distribution and high availability of the Django application.
- **EC2 Instances:** Virtual servers running the Django backend, handling dynamic content and business logic. EC2 instances provide the compute resources necessary for the application’s operations.

3. Data Tier / Database:

- **Amazon RDS (Relational Database Service):** Manages the relational database where application data is stored. RDS provides scalable, secure, and managed database services, ensuring data integrity and availability.

Connections and Flow:

- **Route 53 to CloudFront:** Route 53 directs DNS requests to CloudFront, which handles content delivery.
- **CloudFront to S3:** CloudFront fetches and serves static assets stored in S3, delivering them to users efficiently.
- **CloudFront to ELB:** For requests requiring dynamic content, CloudFront routes traffic to the ELB.
- **ELB to EC2 Instances:** The ELB balances the traffic among EC2 instances, which run the Django application.
- **EC2 Instances to RDS:** The Django backend interacts with RDS to perform data operations, such as retrieving or storing information.

Additional Considerations:

- **Security:** Ensure EC2 instances are protected by appropriate security groups and that the architecture operates within a secure Virtual Private Cloud (VPC).
- **Monitoring and Management:** Incorporate AWS CloudWatch for monitoring application performance and AWS IAM (Identity and Access Management) for managing permissions and access controls.

This architecture provides a robust foundation for hosting a portfolio website, ensuring scalability, security, and efficient content delivery.