1. **Frontend: React Application**
    a. **Service**: Amazon S3 for static file storage and Amazon CloudFront for CDN.
    b. **Setup**: Deploy the static files from the React app to an S3 bucket and serve them using CloudFront, which will provide low latency through edge locations.
2. **Backend: Python API (Flask/Django)**
    a. **Service**: AWS Elastic Beanstalk or Amazon EC2 for easy management and scaling.
    b. **Setup**: Deploy your backend application using Elastic Beanstalk, which automatically manages the scaling and application health monitoring.
3. **Database: MySQL**
    a. **Service**: Amazon RDS (Relational Database Service) for MySQL.
    b. **Setup**: Deploy your MySQL database on Amazon RDS, which provides multi-AZ deployment for high availability, automatic backups, and easy scaling.
4. **Load Balancer and Auto Scaling**
    a. **Service**: AWS Elastic Load Balancer (ELB) and Auto Scaling Group.
    b. **Setup**: Use a load balancer to distribute incoming traffic across multiple instances of your backend application. Configure an Auto Scaling Group to ensure the right number of instances are running based on demand.
5. **Security**
    a. **Service**: AWS Identity and Access Management (IAM) for proper permissions, Amazon VPC, Security Groups, and AWS WAF (Web Application Firewall) for additional protection.
    b. **Setup**:
        i. Place your S3 bucket in a VPC if necessary.
        ii. Use IAM roles to provide necessary access to your services securely.
        iii. Implement Security Groups to restrict access to your backend and database.

iv. Use WAF to protect against common web exploits.

6. **Domain and SSL**
   a. **Service**: Amazon Route 53 for DNS and AWS Certificate Manager for SSL certificates.
   b. **Setup**: Route user requests to your CloudFront distribution for the frontend and to your Elastic Beanstalk environment for the backend. Use AWS Certificate Manager to secure your application with SSL.
7. **Monitoring and Logging**
   a. **Service**: Amazon CloudWatch and AWS X-Ray.
   b. **Setup**: Utilize CloudWatch for monitoring resource usage and application performance. Use AWS X-Ray to debug and analyze performance issues in your application.