**Subject:** Proposal of AWS Architecture for Fastier Web Application

Hi Lilly,

Thank you for reaching out to us regarding the performance and reliability issues you're experiencing with Fastier's web application. We understand the importance of addressing these challenges swiftly to ensure a seamless user experience, especially during periods of rapid growth. After careful consideration, our team has devised a comprehensive AWS architecture tailored to meet Fastier's needs and support its continued success.

Here's the proposed architecture along with explanations of each component, why it was chosen, and how costs are calculated:

**1. Architecture Overview:**

(Architectural Diagram is uploaded here)

**2. Components:**

**- Amazon ECS (Elastic Container Service):** We recommend migrating your application to Amazon ECS, which will enable you to run and manage Docker containers at scale. ECS offers flexible deployment options, high availability, and seamless integration with other AWS services.  
    
**- Amazon RDS (Relational Database Service):** RDS will host your PostgreSQL database, providing high availability, automatic backups, and scalability. This managed service ensures data integrity and minimizes administrative overhead.  
    
**- Amazon S3:** Utilize Amazon S3 for storing static assets and user-generated content. With built-in redundancy and scalability, S3 ensures reliable and efficient storage for your application.  
    
**- Amazon CloudFront:** Implement CloudFront as a content delivery network (CDN) to cache and deliver content closer to your users, reducing latency and improving performance.  
    
**- AWS CodePipeline:** Set up a CI/CD pipeline using CodePipeline to automate the deployment process. This will enable you to deliver updates to your application quickly and efficiently, minimizing downtime and streamlining development workflows.  
    
**- Amazon CloudWatch:** Monitor application performance and resource utilization using CloudWatch. Set up alarms to notify you of any issues or anomalies, allowing for proactive management and troubleshooting.

**3. Why It Was Chosen:**

**- Scalability:** ECS and RDS support horizontal scaling, allowing your application to handle increased traffic and workload seamlessly.  
    
**- Reliability:** RDS ensures data durability and availability, while CloudFront improves content delivery performance and resilience.  
    
**- Cost-effectiveness:** By leveraging managed services like ECS and RDS, you can optimize costs by only paying for the resources you use. Additionally, CloudFront's pay-as-you-go pricing model helps minimize expenses associated with content delivery.

**4. Cost Calculation:**

  - Costs may vary depending on factors such as instance types, storage usage, data transfer, and CDN usage.  
    
  - ECS costs are based on the number and type of containers deployed, along with associated resources like CPU and memory.  
    
  - RDS costs are determined by instance size, storage capacity, and database operations.  
    
  - CloudFront charges are based on data transfer and request volume, with pricing varying by region and edge location.

**5. Considerations:**

  - While ECS was chosen for its flexibility and scalability, alternative solutions like AWS Lambda or AWS Fargate could be explored based on specific requirements and workload characteristics.  
    
  - We're committed to helping you navigate the AWS ecosystem and choose the best-fit solutions for Fastier's needs and budget.

We believe that this architecture will address the performance and reliability challenges you're facing while providing a scalable and cost-effective foundation for Fastier's continued growth. Please feel free to reach out if you have any questions or require further assistance. We're here to support you every step of the way.

Looking forward to hearing from you.

Best regards,  
Purna Visesh Devraj  
AWS Solutions Architect