Hi Lilly,

Thank you for reaching out regarding your web application hosting needs. It's great to hear about the growth of Fastier despite the challenges you're facing with your current infrastructure.

Based on your requirements and current setup, I would recommend migrating to a more scalable and resilient architecture leveraging AWS services. Here's a proposed architecture:

1. **Amazon RDS for PostgreSQL**: Migrate your PostgreSQL database to Amazon RDS. This will provide a managed database service with high availability, automatic backups, and scalability options.
2. **Amazon S3 for Static Content**: Store your static assets (such as images, CSS, and JavaScript files) in Amazon S3. This will offload the serving of static content from your application server, improving performance and reducing the load on your server.
3. **AWS Elastic Beanstalk for Application Deployment**: Deploy your Flask application using AWS Elastic Beanstalk. Elastic Beanstalk automatically handles the deployment, scaling, and management of your application, making it easier to manage and scale as your traffic grows.
4. **Amazon CloudFront for Content Delivery**: Use Amazon CloudFront as a CDN to cache and deliver your dynamic and static content from edge locations closer to your users. This will improve the latency and performance of your application, especially for users located far from your application's origin server.
5. **Auto Scaling Groups for EC2 Instances**: Instead of running your application on a single EC2 instance, use Auto Scaling Groups to automatically scale the number of EC2 instances based on demand. This will ensure that your application can handle traffic spikes and reduce the risk of server crashes due to running out of memory.
6. **Amazon Route 53 for DNS Routing**: Use Amazon Route 53 for DNS routing to route traffic to your application. Route 53 also provides health checks, which can be used to implement a basic disaster recovery plan by routing traffic away from unhealthy instances.
7. **Amazon CloudWatch for Monitoring and Logging**: Set up monitoring and logging using Amazon CloudWatch to monitor the performance and health of your application. CloudWatch can help you identify performance bottlenecks and troubleshoot issues quickly.

By implementing this architecture, you'll have a scalable, resilient, and cost-effective solution that can accommodate your current growth trajectory and future expansion plans.

If you need further assistance or have any questions, please feel free to reach out.

Kind regards,

Bhaskar Banerjee

Solutions Architect