

**To:** Lilly.sawyer@fastier.com

**From:** aws@aws.com

**Subject:** Scalable and Resilient AWS Architecture Recommendation for Fastier

Hi Lilly,

Thank you for reaching out, and congratulations on Fastier's recent success! We've reviewed your current architecture and understand the challenges you're facing, slow response times during peak periods, memory-related crashes, deployment downtimes, and lack of disaster recovery.

As your web app is built with React (a Single Page Application), backed by a Python Flask API and PostgreSQL database, all hosted on a single EC2 instance, the current architecture is understandably struggling to keep up with growth. Based on your needs and the flexibility of a startup, we're proposing a **scalable, highly available architecture using AWS Elastic Beanstalk**, along with complementary AWS services that simplify deployment, enable auto-scaling, and improve reliability.

Here's a detailed breakdown of the recommended architecture and the rationale behind each component:

---

## 1. Elastic Beanstalk (with Auto-Scaling EC2 Group)

Elastic Beanstalk (EB) is the centerpiece of your new backend. It wraps together services like EC2, RDS, and Load Balancing into an easy-to-manage environment.

- **Why we chose it:**

EB supports Python (your backend language), manages infrastructure automatically, and offers **auto-scaling, rolling or blue/green deployments, and health monitoring**. It helps reduce downtime during deployments, a key concern you raised.

- **How it's billed:**

EB itself is free, you only pay for the AWS resources it provisions (EC2, RDS, ELB, etc.). Costs will vary based on traffic since auto-scaling adds/removes instances depending on demand.

---

## 2. EC2 Instances (Within Elastic Beanstalk)

Your Flask app will run on a scalable set of EC2 instances.

- **Why we chose it:**

EC2 offers control and performance, while Beanstalk automates the provisioning. With auto-scaling, you won't be stuck with just one t3.medium instance anymore.

- **How it's billed:**

You're billed hourly for each instance. More instances = more cost, but auto-scaling ensures you're only paying for what you use during peak demand.

---

### 3. Elastic Load Balancer (ELB)

Distributes incoming web traffic to your EC2 instances.

- **Why we chose it:**  
Ensures even traffic distribution and improves fault tolerance. It detects unhealthy instances and reroutes traffic automatically.
  - **How it's billed:**  
Billed per hour the load balancer runs + data processed.
- 

### 4. Amazon RDS (PostgreSQL)

Replaces your single-instance PostgreSQL DB with a managed, scalable database.

- **Why we chose it:**  
Offers automated backups, patching, and high availability (if multi-AZ is enabled). Much more reliable than a DB running inside EC2.
  - **How it's billed:**  
Charged per hour based on instance type and size. Storage and backup retention also impact pricing.
- 

### 5. Amazon S3

Hosts your static React frontend (HTML, CSS, JS assets).

- **Why we chose it:**  
S3 is fast, secure, and perfect for hosting static content. Your React SPA can be served with low latency, globally.
  - **How it's billed:**  
Pay for the amount of data stored and transferred. Costs are low unless you have very large assets or high download traffic.
- 

### 6. Amazon Route 53

Manages your domain name and DNS routing.

- **Why we chose it:**  
Easy integration with S3 and ELB. Route 53 enables you to point your domain (like fastier.com) to your hosted frontend and backend seamlessly.
  - **How it's billed:**  
Charged per hosted zone + number of DNS queries.
- 

### 7. AWS CodePipeline

Automates deployment from source code to Beanstalk (backend) and S3 (frontend).

- **Why we chose it:**

Eliminates manual deployment errors and provides **continuous integration and deployment (CI/CD)**. Paired with GitHub or CodeCommit, it builds and deploys your code on push.

- **How it's billed:**

Billed per pipeline per month. Since Fastier may have just 1–2 pipelines, the cost stays predictable and low.

---

## **Optional: Multi-AZ Deployment**

We recommend **adding another Availability Zone (AZ)** to improve disaster recovery. This means EC2 and RDS will be duplicated in a second data center, allowing failover if one zone fails.

- **How it's billed:**

Effectively doubles the infrastructure cost (since resources are duplicated), but adds serious resilience.

---

## **Cost Summary & Variability**

- Your **core costs** will come from EC2, RDS, and ELB—all usage-based.
  - CodePipeline and Route 53 are low, fixed costs.
  - S3 and data transfer charges will depend on traffic and storage.
  - Auto-scaling keeps you from overpaying during quiet periods.
  - The AWS [Pricing Calculator](#) can help forecast monthly costs more accurately based on your usage.
- 

## **Final Thoughts**

This setup gives you:

Seamless scaling for growth

Zero-downtime deployments

Improved performance and availability

Simplified infrastructure management

We'd love to discuss this further and help you with setup and migration. Please let us know a good time for a call or demo!

Best regards,

Solutions Architect, AWS