Prototype Monthly Cost Estimate and K8s vs ECS

This breakdown shows the monthly cost to run the prototype in the cloud. The prototype used Kubernetes (EKS), a Spring Boot backend, and a React frontend. No load balancer or storage was included.

Why 720 Hours?

Cloud providers give prices per hour. To estimate a month, we assume 30 days. 30 days × 24 hours per day = 720 hours. So we multiply the hourly rate by 720 to get the monthly cost.

1. EKS Control Plane

The control plane manages the cluster and keeps everything organized. AWS charges $0.10 per hour. $0.10 × 720 hours = $72, which is rounded to about $73 per month.

2. EC2 Worker Node

Worker nodes are the machines where the applications run (Spring Boot and React). On AWS, these are EC2 instances. The cost depends on the instance size:

Option	Specs	Hourly	Calc (×720h)	Monthly
t3.small	2 vCPUs, 2 GB RAM	$0.0208	$0.0208 × 720	~$15
t3.medium	2 vCPUs, 4 GB RAM	$0.0416	$0.0416 × 720	~$30

3. Total Prototype Cost

To get the total, we add the control plane cost to the worker node cost: - Control plane: ~$73 per month - Worker node: $15 (t3.small) or $30 (t3.medium) So the totals are: - $73 + $15 = ~$88 per month - $73 + $30 = ~$103 per month The React frontend runs on the same worker node as the backend, so it does not add extra cost. Networking and storage are so small in this demo that they can be ignored.

4. Why Kubernetes Instead of ECS

I looked at both Kubernetes (EKS) and Amazon ECS while working on the prototype. ECS is simpler and doesn't have a fixed control plane fee, but Kubernetes made more sense for a few reasons:

• I was able to develop and test locally using Minikube in the same way it would run on AWS.
• Kubernetes works across different cloud providers, which gives more flexibility for the future.
• It has a very large community and ecosystem, which makes it easier to find support and tools.

Even though EKS adds about $73 per month for the control plane, the flexibility and future-proofing made Kubernetes the better choice for this prototype.