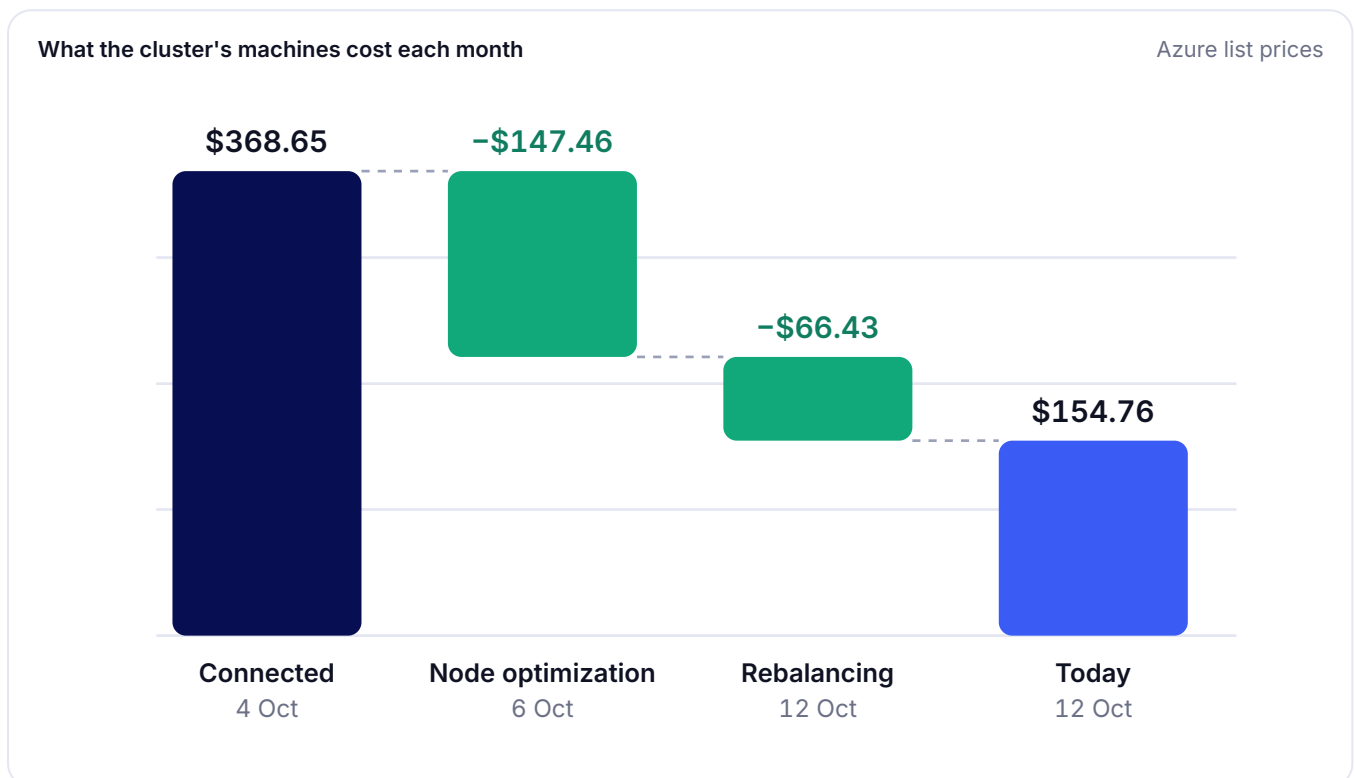


VALIDATION LAB · CORAL ON AZURE KUBERNETES SERVICE · OCTOBER 2026

How we reduced our own cluster's machine bill by 58%

Before we ask anyone to trust Coral with their cluster, we run it on ours. In October 2026 we connected Coral to the Kubernetes cluster that runs Zoorik's own development platform on Azure. Eight days later the cluster cost **\$154.76 a month instead of \$368.65**, and every app was still running.



-58%
 machine bill

\$213.89
 saved every month,
 about \$2,567 a year

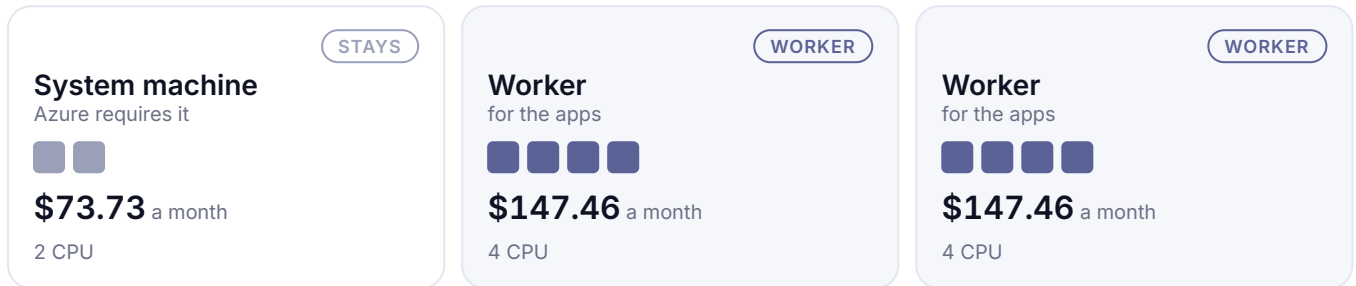
10 → 6
 CPU, on 2 machines
 instead of 3

8 days
 from connecting
 Coral to the result

Coral is Zoorik's Kubernetes cost optimization for AKS and EKS. It right-sizes apps, removes machines a cluster no longer needs, and rebalances onto cheaper machines, all within each app's safety limits.

1 The starting point

Our development cluster runs Coral's own API, our consoles and our deployment tooling. It is a typical small team cluster: a system machine that Azure requires, and two worker machines for the apps. Like most clusters, it was sized for safety, not for what the apps actually used.

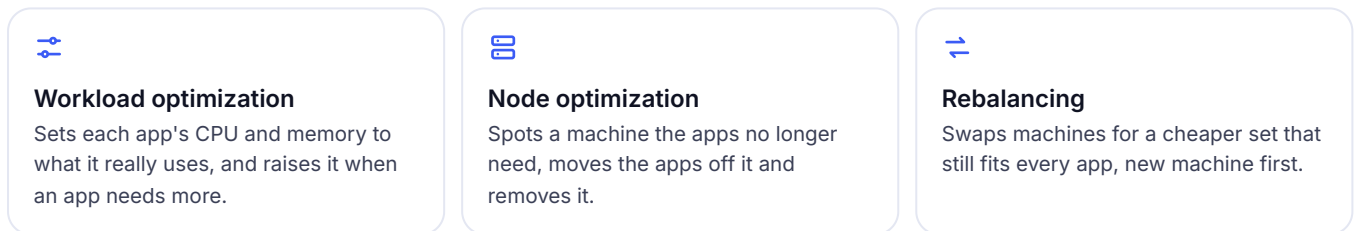


On 4 October 2026, when Coral was connected

3 machines · 10 CPU · \$368.65 a month

2 Connecting Coral

One **zoorikctl** command installed Coral's agent in the cluster. Two days later we gave Coral permission to remove machines, in this cluster only. From then on Coral worked on its own, and we watched.



3 What Coral did

Node optimization: a spare worker removed

6 Oct

Coral worked out that the apps would fit on fewer machines. It moved them to the remaining machines and removed one 4-CPU worker, **on its own, in under 10 minutes**. Nobody switched anything on. Apps that run a single copy restarted once on another machine, and every app was running again afterwards.

System machine


\$73.73 a month

STAYS

Worker


\$147.46 a month

STAYS

Worker


\$147.46 a month

-**\$147.46** a month

REMOVED

-**\$147.46** a month

Workload optimization: apps sized to what they use

from 8 Oct

We switched workload optimization to automatic. By 12 October Coral had right-sized **14 of the cluster's apps**. Most were reserving more than they used; one app needed more memory, and Coral gave it more.

CPU the apps reserve

Before


45% less

Now



0.78 cores

0.43 cores

Memory the apps reserve

Before


16% less

Now



6.4 GiB

5.4 GiB

Less reserved, more room on every machine

Rebalancing: one click, a cheaper machine

12 Oct

With one click, Coral swapped the remaining worker for a **cheaper machine of the same size**, chosen from Azure's prices. Azure built the new machine in under 2 minutes; Coral moved the apps one copy at a time and then removed the old machine. **8 minutes 36 seconds** from start to finish.

Worker

4 CPU, Intel


\$147.46 a month

BEFORE



New worker

4 CPU, AMD, same size


\$81.03 a month

AFTER

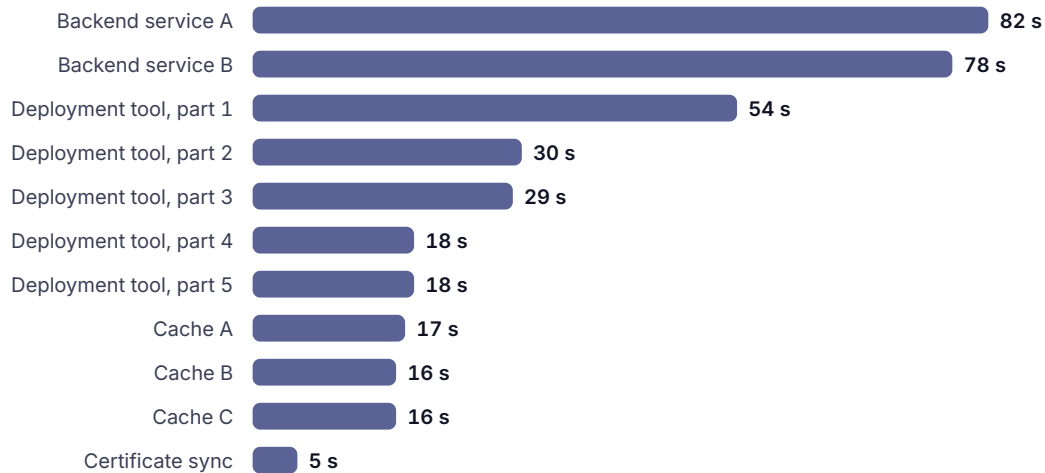
-**\$66.43** a month

4 How the apps came through

Moving an app to another machine means stopping it in one place and starting it in another. Here is what that meant for our apps during the rebalance on 12 October, measured from the moment the old copy stopped to the moment the new one was ready.

Apps that run a single copy: each restarted once

seconds until back up



0 seconds for the 6 apps that run two copies, including Coral's API and both of our consoles: one copy was always running while the other moved.

What this means: an app that runs a single copy is down for a short time whenever it moves, with Coral or without it. Here that was 5 to 82 seconds. Apps that must never pause should run two copies, and Coral keeps one of them running. When the spare worker was removed on 6 October, single-copy apps also restarted once; those times were not measured.

5 How the apps stayed safe



New machine first

Nothing was removed until its replacement was ready and its apps had moved.



One copy at a time

Every move kept within each app's own limits, so apps that run two copies always kept one running.



Checks before every move

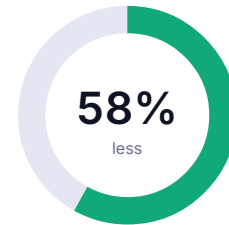
Where a faster move could have put an app at risk, Coral chose the safe one.

6 The result

Before and after

	4 Oct	12 Oct	Difference
Machines	3	2	1 fewer
CPU	10	6	4 fewer
Per month	\$368.65	\$154.76	\$213.89 less
Per year	\$4,423.80	\$1,857.12	\$2,566.68 less

Share of the machine bill removed



\$213.89 of \$368.65 a month

See what Coral would save on your cluster

Run the free assessment at zoorik.com/assessment, or talk to us at zoorik.com.

Zoorik's own development cluster on Azure Kubernetes Service, 4 to 12 October 2026. Costs are Azure pay-as-you-go list prices for the cluster's machines at 730 hours a month. App figures cover the cluster's own apps. Results depend on each cluster's apps and machines.