Pepperdata for Amazon EMR

Reduce your Amazon EMR costs by up to 47% with zero code changes

Enterprises grappling with controlling cloud costs in these uncertain economic times rely on Pepperdata to optimize their cloud spend.

> *"Pepperdata allowed us to significantly increase capacity for our Amazon EMR workloads and reduce our costs by over 50%. We can focus on our business, while they optimize for costs and performance."*
>
> —Mark Kidwell, Chief Data Architect, Data Platforms and Services, Autodesk, Inc.

Pepperdata for Amazon EMR autonomously optimizes your Spark jobs and reduces your Amazon EMR costs by up to 47 percent with no application code changes, no recommendations, and no manual tuning.

Instance Hour Reduction

Continuously optimize your unused node resources to reduce instance hours and costs without the need for recommendations

Enhanced Autoscaling

Gain additional savings on top of your autoscaler by automatically utilizing instances fully before adding more

Real-Time Optimization

Automatically tune your environment to its "sweet spot" without manual intervention

Driving Savings for Global Enterprises

AUTODESK
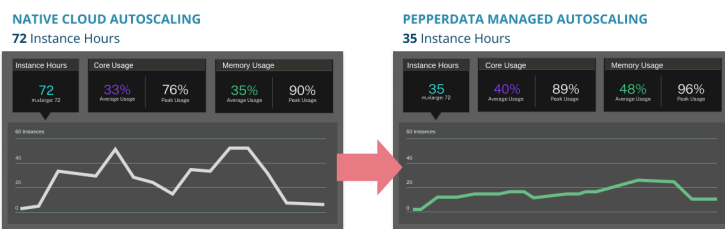
citi

RBC

EXTOLE

IQVIA™

Magnite

Instance Hour Reduction

Pepperdata Capacity Optimizer reduces costs for Apache Spark workloads on Amazon EMR by optimizing CPU and memory usage in real time using machine learning. When installed, Pepperdata instructs the native EMR scheduler to focus on *actual usage* rather than *allocated usage*, enabling it to determine where there is unused capacity in each node.

Capacity Optimizer then instructs the scheduler to add resources only to nodes that are underutilized, and to launch new instances only when resource capacity in existing instances is full—thus reducing instance hours. This reduction in instance hours directly correlates to reduced cloud costs, with some companies seeing tens of thousands of dollars in savings per day.

Pepperdata Capacity Optimizer Slashes Instance Hours By 51%

NATIVE CLOUD AUTOSCALING — 72 Instance Hours

Instance Hours	Core Usage		Memory Usage	
72 (Average 72)	33% Average Usage	76% Peak Usage	35% Average Usage	90% Peak Usage

PEPPERDATA MANAGED AUTOSCALING — 35 Instance Hours

Instance Hours	Core Usage		Memory Usage	
35 (Average 72)	40% Average Usage	89% Peak Usage	48% Average Usage	96% Peak Usage

Enhanced Autoscaling

While the native autoscaler provisions resources to accommodate peak workloads, most workloads achieve peak usage only on occasion. Capacity Optimizer ensures that all nodes are fully utilized before additional nodes are created and directs the Amazon EMR autoscaler to downscale as fewer resources are required. Pepperdata also supports your cluster to automatically downsize the number of autoscaling instances in a time-sensitive and robust manner. By automatically reducing the number of instances to match actual usage, Pepperdata allows you to achieve up to 47 percent cost savings above the default configuration with no code changes.

Real-Time Optimization

Most organizations provision applications for peak usage, a provisioning level that remains in place even when applications scale down.

Pepperdata understands that workloads are dynamic: workloads operate at peak and also scale down below peak levels for significant amounts of time. Pepperdata Capacity Optimizer is the only Real-Time Cost Optimization solution that calibrates your workloads to their "sweet spot" without needing to apply recommendations, make application changes, or manually tune your clusters.

"I've enjoyed the Pepperdata relationship because it was an easy way to save money... If you can get that money back and maybe save some time too with this amount of investment, it's really a no-brainer."

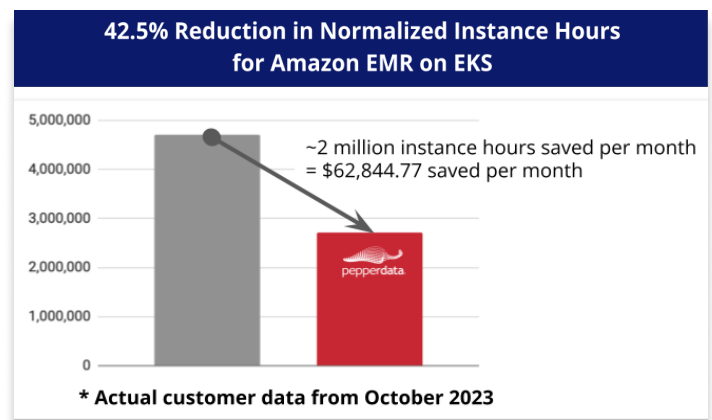—Ben Smith, VP of Technical Operations, Extole

With Pepperdata Capacity Optimizer, no one in the operations or engineering teams needs to manually optimize their cluster resources or update their application code to help control costs. Using machine learning, Pepperdata Capacity Optimizer autonomously examines all clusters in a cloud environment to identify where more work can be done within a cluster.

There is also no need to apply optimizations based on recommendations since Pepperdata optimizations are automatically put in place in real time. Because Pepperdata acts in real-time to continuously ensure node capacity is being fully utilized, your costs are immediately reduced and you are further enabled to launch additional applications at no extra cost in nodes where unused capacity is identified.

Pepperdata for Amazon EMR on EC2 and Amazon EMR on Amazon EKS

Pepperdata can immediately reduce costs for workloads running on Amazon EMR on EC2 as well as on Amazon EMR on Amazon EKS. Pepperdata conducted a benchmark study of Capacity Optimizer for Amazon EMR on EC2 modeled after the industry standard TPC-DS framework. Running a set of 103 queries, the benchmark showed a **38 percent reduction of instance hours** in addition to the savings offered by the standard AWS Custom Auto Scaler. The benchmark found that Pepperdata excelled in the most complicated queries that reflected the most demanding real-world environments.

In an informal benchmarking of a customer's Amazon EMR on EKS environment, Capacity Optimizer reduced instance hours costs by 42.5 percent, resulting in about two million instance hours saved per month—**equating to ongoing monthly savings of $62,844.77.**

42.5% Reduction in Normalized Instance Hours for Amazon EMR on EKS

~2 million instance hours saved per month = $62,844.77 saved per month

* Actual customer data from October 2023

Your Easy Button to Cost Savings

As a result of its ability to work with the native system scheduler to identify where more work can be done, and to find and add tasks to nodes with available resources —continuously and in real time—Pepperdata Capacity Optimizer can reduce your instance hours by up to 47 percent automatically with no manual tuning, no recommendations, and no application changes. Pepperdata installs in under an hour in most environments with typical ROI of 100 percent to over 650 percent.