PRODUCT SHEET

Avesha’s technologies are the **next generation innovation** abstracting the network and infrastructure for application needs.

Avesha

KubeTally

Security
Multi-Cluster Observability
Load Balancing
Global Load Balancing
Service Layer Connectivity
Intelligent Predictive Remediation
Resiliency
Multi-Cluster Observability
Device Connectivity Fabric
Security

Tracking Chargeback Amounts per Team in Kubernetes Environments

Introduction

In Kubernetes (K8s) management, accurately attributing resource consumption to the correct business unit or team is crucial for operational efficiency and cost management. A large Pharma company is leveraging KubeSlice to address chargeback issues that contemporary Kubernetes Cost Management solutions are unable to solve. Tools like Kubecost offer visibility into namespace usage but fall short in isolating and attributing these costs effectively across business units. This is where KubeSlice, a sophisticated Kubernetes management tool, steps in to fill the gap, offering chargeback and resource management capabilities across multi-cluster environments.

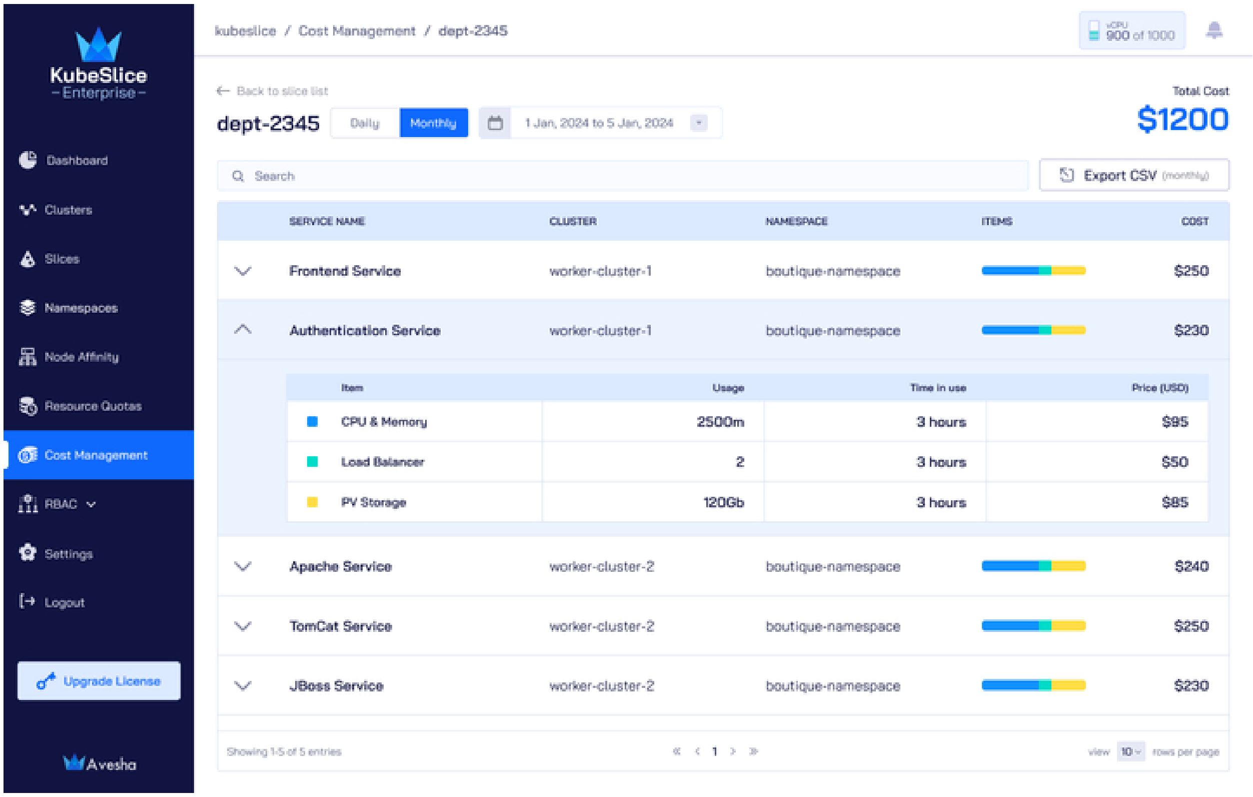
The Challenge of Chargeback in Kubernetes

Kubernetes environments, especially those spanning multiple clusters, present a complex challenge in tracking and attributing resource usage. Business units or application teams within an organization often share these resources, making it difficult to accurately charge back the costs incurred by each unit. The inability to isolate and track usage for namespaces per business unit complicates this process further, leading to inefficiencies and conflicts over resource consumption charges.

Chargeback Per Team Example

To illustrate how KubeSlice enables precise chargeback functionality, let's consider a hypothetical example of a “Boutique Team”, that has created a "Boutique Slice" using the KubeSlice product. "Boutique Slice" consists of the namespaces that the Boutique Team works on. This example shows how costs can be tracked across various services within the Boutique Slice, providing a clear and transparent way to attribute costs to specific components of the application.

Chargeback Table for Boutique Slice (Boutique Team’s Services)



dept-2345

1 Jan, 2024 to 5 Jan, 2024

Total Cost
$1200

	SERVICE NAME	CLUSTER	NAMESPACE	ITEMS	COST
⌄	Frontend Service	worker-cluster-1	boutique-namespace		$250
⌃	Authentication Service	worker-cluster-1	boutique-namespace		$230
⌄	Apache Service	worker-cluster-2	boutique-namespace		$240
⌄	TomCat Service	worker-cluster-2	boutique-namespace		$250
⌄	JBoss Service	worker-cluster-2	boutique-namespace		$230

Item	Usage	Time in use	Price (USD)
🟦 CPU & Memory	2500m	3 hours	$95
🟩 Load Balancer	2	3 hours	$50
🟨 PV Storage	120Gb	3 hours	$85

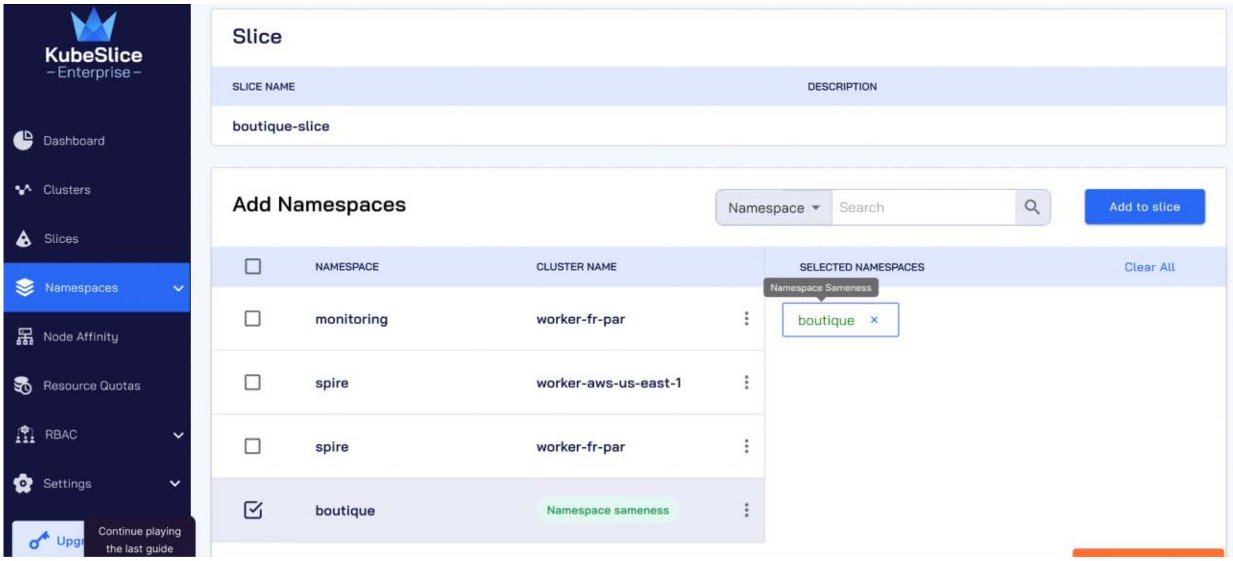
Showing 1-5 of 5 entries « ‹ 1 › » view 10 rows per page

KubeSlice: A K8s Management Solution for Tracking Kubernetes Chargeback for Multi-Cluster Environments

KubeSlice is designed to address these challenges, with features that not only provide visibility into resource usage but also enable precise chargeback mechanisms. Here's how:

Isolation Across Clusters

One of KubeSlice's key features is its ability to maintain “namespace sameness” across multiple clusters. This means that different namespaces, potentially in different clusters, can be equated and connected, allowing for aggregated usage tracking. This capability is crucial for organizations operating in multi-cluster environments, ensuring that application teams are accurately charged for their resource consumption, regardless of where those resources are deployed. Current solutions that offer a chargeback solution, do not aggregate costs across multiple clusters for team namespaces.



Slice

SLICE NAME	DESCRIPTION
boutique-slice	

Add Namespaces

□	NAMESPACE	CLUSTER NAME
□	monitoring	worker-fr-par
□	spire	worker-aws-us-east-1
□	spire	worker-fr-par
☑	boutique	Namespace sameness

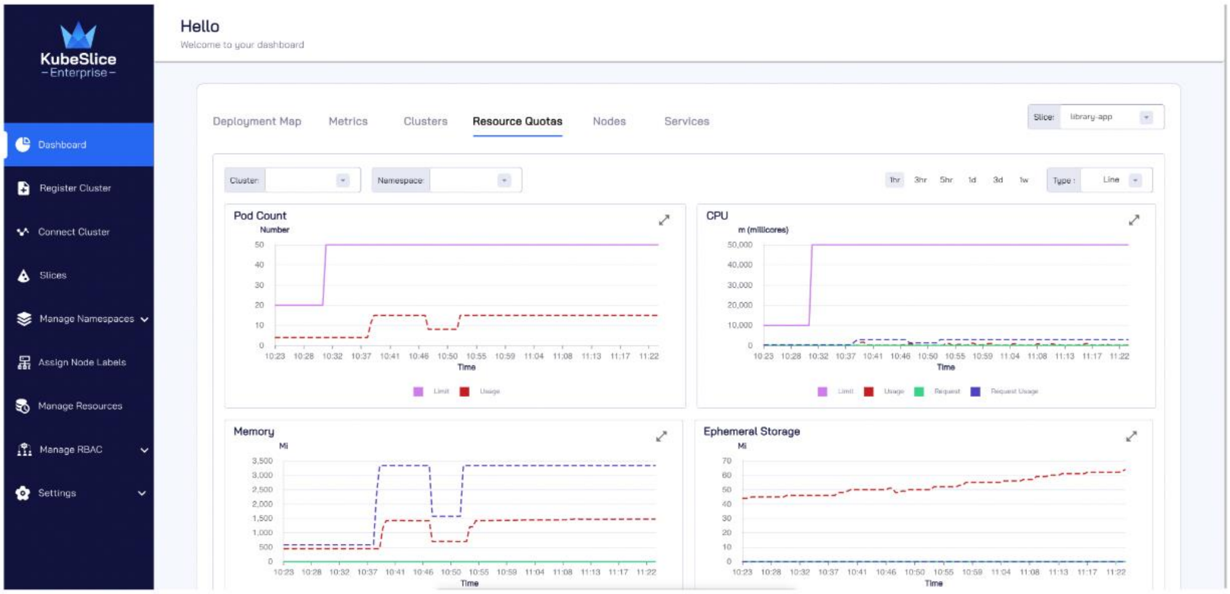
SELECTED NAMESPACES

boutique ×

Fig1: Namespace Sameness in KubeSlice

Dashboard - Resource Utilization

The KubeSlice dashboard provides a view of resource utilization across a multi-cluster Slice over time. With the ability to view resource usage by the number of pods, CPU, memory, and ephemeral storage, application and platform teams can gain insights into how resources are being consumed and by whom (which team)



Hello

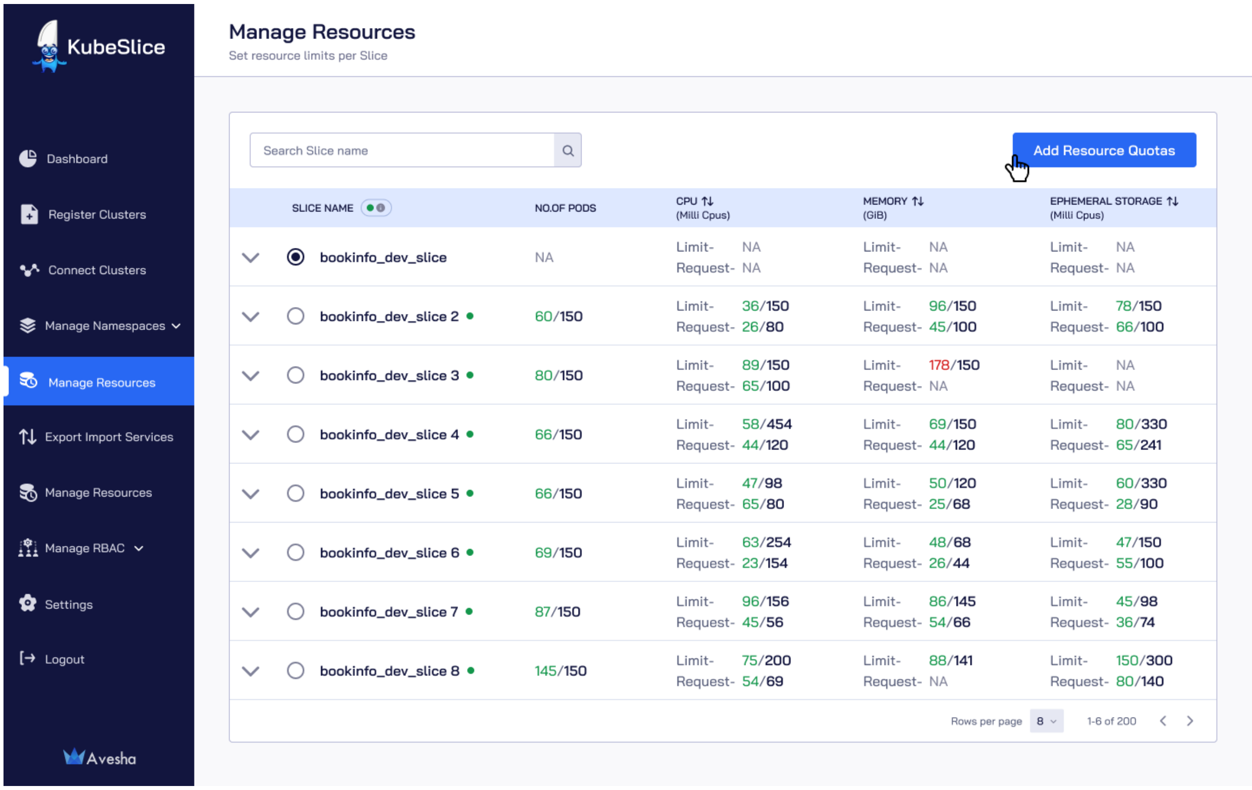
Welcome to your dashboard

Fig2: Resource Utilization Per Namespace in KubeSlice

Manage Resources Effectively

The "Manage Resources" section of KubeSlice allows teams to set resource limits per Slice, ensuring that each business unit or application team has access to the resources they need without exceeding their allocated budget. The ability to add resource quotas and monitor them through a clear and organized interface helps in maintaining efficiency and preventing resource misuse by certain teams.

Specifically, it incorporates mechanisms to safeguard the integrity of chargeback information. This includes the use of immutable timestamps and secure logging to ensure that data related to resource usage cannot be tampered with. Such features are crucial for maintaining accurate accounting records



Manage Resources

Set resource limits per Slice

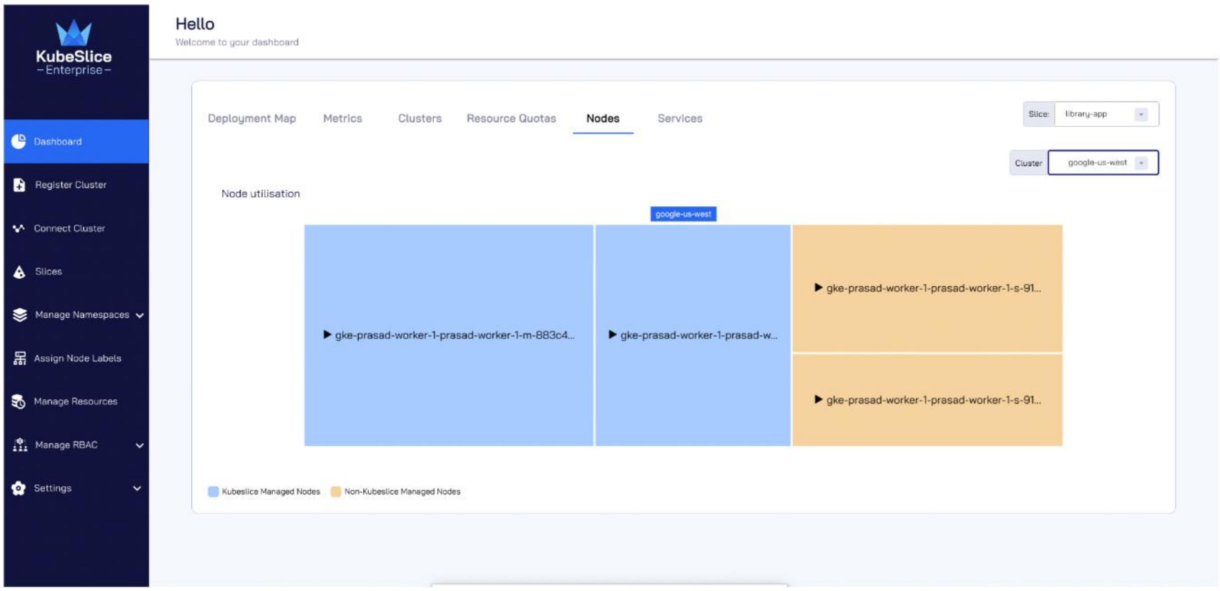
	SLICE NAME	NO.OF PODS	CPU ⇅ (Milli Cpus)	MEMORY ⇅ (GiB)	EPHEMERAL STORAGE ⇅ (Milli Cpus)
●	bookinfo_dev_slice	NA	Limit- NA Request- NA	Limit- NA Request- NA	Limit- NA Request- NA
○	bookinfo_dev_slice 2 ●	60/150	Limit- 36/150 Request- 26/80	Limit- 96/150 Request- 45/100	Limit- 78/150 Request- 66/100
○	bookinfo_dev_slice 3 ●	80/150	Limit- 89/150 Request- 65/100	Limit- 178/150 Request- NA	Limit- NA Request- NA
○	bookinfo_dev_slice 4 ●	66/150	Limit- 58/454 Request- 44/120	Limit- 69/150 Request- 44/120	Limit- 80/330 Request- 65/241
○	bookinfo_dev_slice 5 ●	66/150	Limit- 47/98 Request- 65/80	Limit- 50/120 Request- 25/68	Limit- 60/330 Request- 28/90
○	bookinfo_dev_slice 6 ●	69/150	Limit- 63/254 Request- 23/154	Limit- 48/68 Request- 26/44	Limit- 47/150 Request- 55/100
○	bookinfo_dev_slice 7 ●	87/150	Limit- 96/156 Request- 45/56	Limit- 86/145 Request- 54/66	Limit- 45/98 Request- 36/74
○	bookinfo_dev_slice 8 ●	145/150	Limit- 75/200 Request- 54/69	Limit- 88/141 Request- NA	Limit- 150/300 Request- 80/140

Rows per page 8 1-6 of 200

Fig3: Setting Limits and Requests Per Namespace in KubeSlice

Node Utilization Insights

KubeSlice's node utilization feature gives a deeper view into how resources are utilized at the node level for multi-cluster environments. When you see node utilization by namespace, platform teams can identify trouble areas and optimize resource allocation to ensure smooth operations. This level of granularity is invaluable for maintaining high performance and availability in Kubernetes environments.



Hello

Welcome to your dashboard

Slice: library-app

Cluster google-us-west

Fig4: Node Affinity Per Namespace in KubeSlice

Conclusion

KubeSlice simplifies managing of Kubernetes resources and implementing effective chargeback mechanisms. Its ability to aggregate usage per namespace across multiple clusters, combined with detailed insights into resource and node utilization, enables organizations to accurately attribute costs to the right business units. This not only ensures fairness and transparency in chargeback practices but also enhances overall operational efficiency in managing Kubernetes environments.