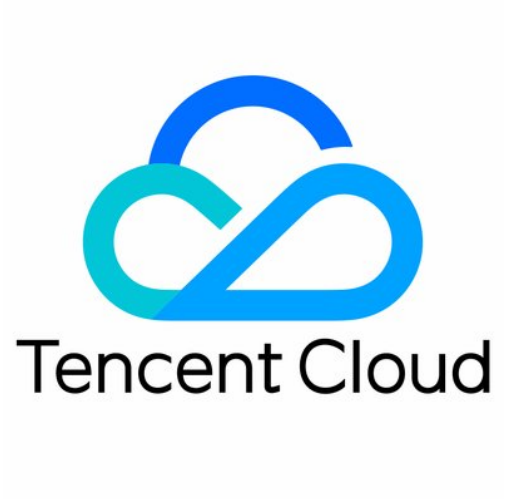
Tencent Kubernetes Engine

TKE Insight

Product Documentation



Contents

TKE Insight
Cost Insight

Last updated：2024-05-24 16:00:58

Feature Description

Traditional Kubernetes clusters usually focus solely on resource-level data, such as the Allocatable indicator for nodes representing resources that can be allocated, and Metrics-Server that can monitor the utilization of nodes and pods. TKE Insight has introduced Cost Insight, a cost visualization dashboard that helps users better understand, monitor, and control resource usage and cost situations in a containerized environment. It mainly includes the following objectives:

Transparency: Users can intuitively understand the resource usage of applications and services in a containerized environment, including the cost consumption in terms of CPU and memory across different clusters, namespaces, and workloads. This helps users comprehend the overall cost structure and improve decision-making efficiency.

Cost control and optimization: Through Cost Insight, users can identify applications or services with high costs. Users can optimize such issues, for example, through intelligent request recommendation, preemptible jobs, the native node dedicated scheduler, and QoSAgent, to reduce unnecessary expenses.

Budget management: Users can formulate more reasonable budgets based on actual cost situations and monitor budget implementation. When resource usage approaches or exceeds the budget, users can adjust policies in time to avoid extra expenditures.

Directions

1. Log in to the Tencent Kubernetes Engine console.
2. On the left, select **TKE Insight > Cost Insight**.

Instructions

The cost value in Cost Insight is a calculated value for reference only. For accurate costs, log in to Tencent Cloud Billing Center to check.

Super nodes cannot distinguish whether the pay-as-you-go billing is for general pay-as-you-go or bidding pods. Therefore, it is currently not supported to use reserved coupons to offset the cost of pods of corresponding specifications. This is because reserved coupons require API calls to query coupon information during cost calculation, and placing the call in the monitoring component would put significant pressure on the API.

The percentage change in costs for the current month compared with the previous month will not be displayed if there is no cost data for the previous month.
In cost calculation, the GPU usage of pods is not accounted for. Instead, the price of the GPU is amortized over the CPU and memory. As a result, the unit price of CPU and memory on GPU nodes is increased.
For the list price indicators of native nodes and general nodes, data is reported based on the unit price of the pay-as-you-go billing mode as the billing mode of nodes cannot be obtained.
For pods on super nodes billed on a pay-as-you-go basis, when multiple GPU types are specified in the annotation, the price is calculated based on the first GPU type.

Pricing

Billing Mode

Billing items include general nodes (CVM), native nodes, and super nodes.

General Node Billing Mode

Based on the node model and billing mode (pay-as-you-go), find the node's list price.
Note:
The model is indicated in the node's annotation of node.kubernetes.io/instance-type.

Native Node Billing Mode

Based on the node model and billing mode (pay-as-you-go), find the node's list price.
Note:
The model is indicated in the node's annotation of node.kubernetes.io/instance-type.

Super Node Billing Mode

Pay-as-you-go

List price of pay-as-you-go super node = Sum of list prices of nodes of the super node where pods are located
The list price of a node where a pod is located is calculated as follows:
Determine the CPU and GPU types of the node. If multiple CPU types are specified, use the first type for price calculation. If multiple GPU types are specified, use the first type for price calculation.
If the GPU type is not empty, look up the CPU unit price, memory unit price, GPU unit price, and license unit price (which is 0 CNY license is free) corresponding to the GPU type. If the GPU type is empty, look up the CPU unit price and memory unit price corresponding to the CPU type.
Determine the specification of the node where the pod is located (i.e., CPU specification, memory specification, and GPU specification). For more details, see Resource Specification Calculation Method and Resource Specifications.
Calculate the list price of the pod: GPU specification × GPU unit price + CPU specification × CPU unit price + Memory specification × Memory unit price + License unit price.

Note:
The CPU type is indicated in the pod's annotation of eks.tke.cloud.tencent.com/cpu-type.
The GPU type is indicated in the pod's annotation of eks.tke.cloud.tencent.com/gpu-type.

Reporting Node List Price to Tencent Cloud Observability Platform

1. For nodes, the list price of each node will be reported to Tencent Cloud Observability Platform (TCOP). Currently supported granularities include 5min and 1h. The 5min granularity represents the price of the node per 5 minutes, and the 1h granularity represents the price of the node per 1 hour.
2. For pods on pay-as-you-go super nodes, the list price of each pod will be reported. Currently supported granularities include 5min and 1h. The 5min granularity represents the price of the sun-machine where the pod is located per 5 minutes, and the 1h granularity represents the price of the sun-machine where the pod is located per 1 hour.

1. Method of Calculating the Cost of the Current Month

Fetch the last month's prices of general nodes, native nodes, and super nodes from TCOP and add them together to get a total price.

2. Method of Estimating the Cost of the Next Month

Fetch the last hour's prices of general nodes, native nodes, and super nodes at the 1h granularity from TCOP. Assume that the sum of the last hour's prices of general nodes, native nodes, and super nodes is x CNY, the estimated cost for the next month is x × 24 × 30 CNY.

3. Cost Trend

Today's cost: Fetch the last day's prices of general nodes, native nodes, and super nodes from the monitor, and add the prices together.
Current month's cost trend: Fetch the last month's prices at the 1h granularity from Cloud Monitor. The frontend displays the cost trend of the last month at the 1h granularity.
Note:
The data point of each hour represents the sum of list prices of general nodes, native nodes, and super nodes at the corresponding time point.

4. Resource Cost Distribution

Fetch the last hour's costs of general nodes, native nodes, and super nodes. Multiply each cost by 24 and 30 to calculate the last month's cost of each general node, native node, and super node.

5. Namespace Cost Distribution

Estimate the monthly cost of each namespace load in the cluster based on the cost of the last hour.
1. Retrieve all nodes in the current cluster (general nodes, native nodes, and super nodes).

2. Fetch the CPU usage and memory usage of each pod on general nodes and native nodes from TCOP.
3. Fetch the model information of CVM nodes and native nodes in the current cluster, and calculate the CPU unit price and memory unit price of each model (the CPU unit price and memory unit price of super nodes are fixed, so no calculation is needed). The calculation method is as follows:
3.1 Based on the price estimates for all models across the network, the CPU unit price is x CNY/core/hour, and the memory unit price is y CNY/GB/hour.
3.2 Based on step 3.1, determine the ratio of CPU unit price/Memory unit price. Then, convert the price of the entire machine to a memory-based price to find the memory unit price of the machine. Next, multiply the memory unit price by the ratio to find the CPU unit price. For example, if CPU unit price/Memory unit price = r, the machine has x cores and y GB, and the machine price is z, then the calculation method for CPU and memory unit prices is:
Memory unit price of the machine: z/(xr+y)
CPU unit price: r(z/(xr+y))
4. Based on the CPU usage and memory usage of each pod obtained in step 2, and the CPU unit price and memory unit price of the node where the pod is located, calculate the price of each pod.
5. Based on the pay-as-you-go super nodes obtained in step 2, obtain the list price of pods on the pay-as-you-go super nodes from TCOP.
6. Based on the pod cost information calculated in steps 4 and 5, aggregate the cost information for each namespace.
7. Idle cost: The total cost of all current nodes minus the sum of the namespace costs.

6. Workload Cost Distribution

Estimate the monthly cost of each workload in the cluster based on the cost of the last hour.
1. Retrieve all nodes in the current cluster (general nodes, native nodes, and super nodes).
2. Fetch the CPU usage and memory usage of each pod on general nodes and native nodes from the monitor.
3. Fetch the model information of general nodes and native nodes in the current cluster, and calculate the CPU unit price and memory unit price of each model (the CPU unit price and memory unit price of super nodes are fixed, so no calculation is needed. CPU unit price: 36 CNY/core/hour, memory unit price: 18 CNY/GB/hour). The calculation method is as follows:
3.1 Based on the estimates from TKE FinOps (based on the prices of all models across the network), the CPU unit price is x CNY/core/hour, and the memory unit price is y CNY/GB/hour.
3.2 Based on step 3.1, determine the ratio of CPU unit price/Memory unit price. Then, convert the price of the entire machine to a memory-based price to find the memory unit price of the machine. Next, multiply the memory unit price by the ratio to find the CPU unit price. For example, if CPU unit price/Memory unit price = r, the machine has x cores and y GB, and the machine price is z, then the calculation method for CPU and memory unit prices is:
Memory unit price of the machine: z/(xr+y)
CPU unit price: r(z/(xr+y))
4. Based on the CPU usage and memory usage of each pod obtained in step 2, and the CPU unit price and memory unit price of the node where the pod is located, calculate the price of each pod.

5. Based on the pay-as-you-go super nodes obtained in step 2, obtain the list price of pods on the pay-as-you-go super nodes from TCOP.
6. Based on the pod cost information calculated in steps 4 and 5, aggregate the cost information for each workload.
7. Idle cost: The total cost of all current nodes minus the sum of the workload costs.

Node Map

Last updated：2024-05-24 15:27:45

Overview

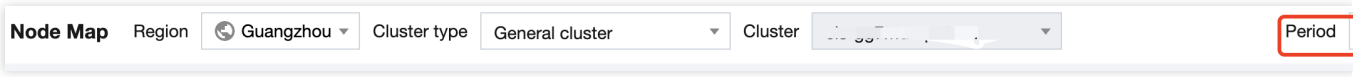
Traditional object management webpages typically use a list-based format, as exemplified by the existing resource object list for TKE clusters in the TKE console. However, this approach suffers from several limits, including low readability due to abstract numerical values, lack of clear object identification, and the inability to support certain sorting methods. To break these limits, **TKE Insight** has introduced Node Map, a platform that displays all resource objects of users in a visualized way. This platform provides rich query filtering, type-based aggregation, and status display features that enable users to rapidly locate their required objects.
Node Map displays the status and metrics of nodes by using visualized charts and diagrams on webpage. This assists users in comprehending the resource utilization and load rate of their current cluster nodes, as well as in analyzing any potential issues on the nodes.

Precaution

Node Map does not support the statistical analysis of metrics on super nodes.

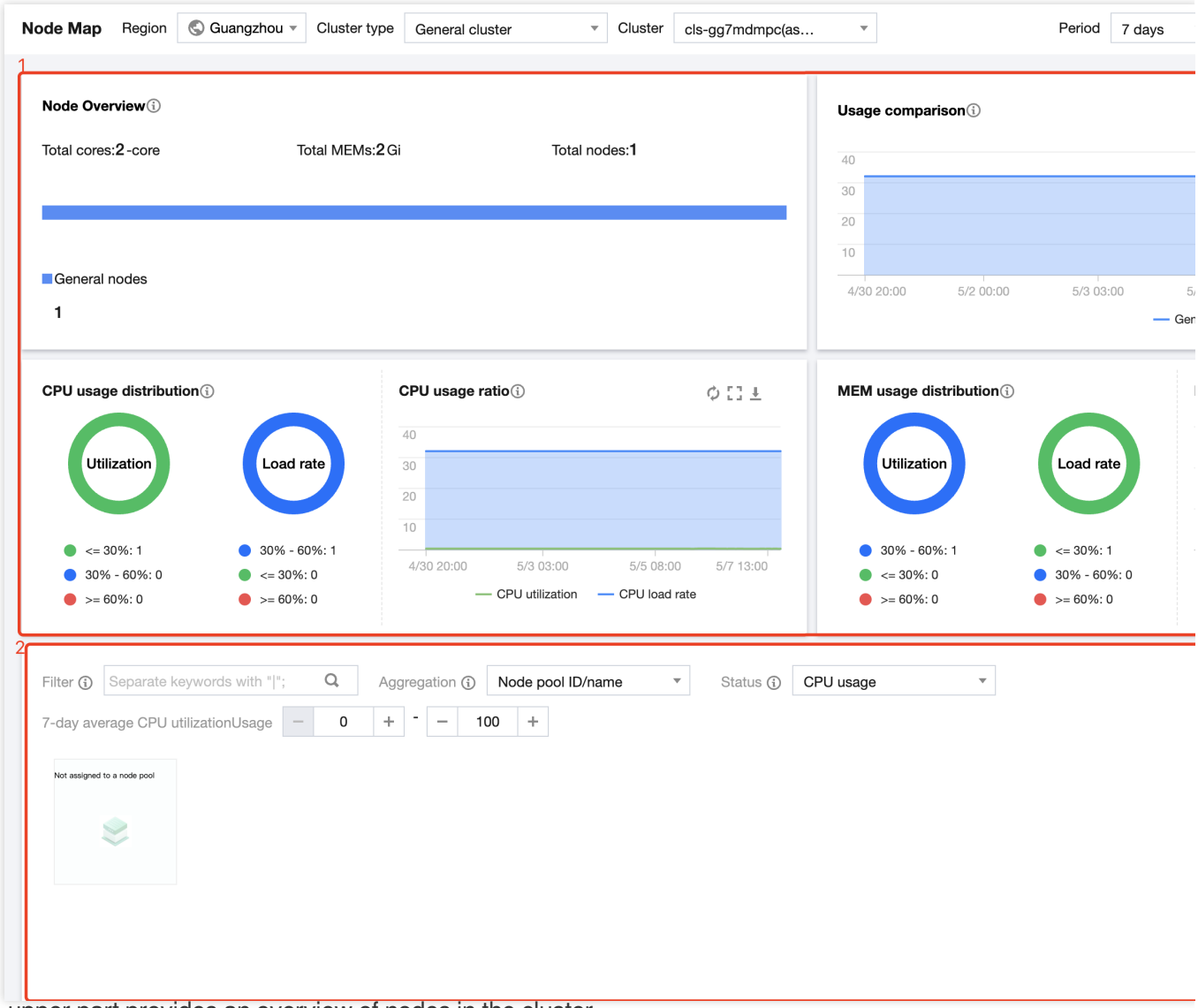
Directions

1. Log in to the Tencent Kubernetes Engine console.
2. On the left, select **TKE Insight > Node Map**.
3. On the Node Map page, set the **Region**, **Cluster Type**, and **Cluster** to view data.
4. In the upper right corner, set the **Period** (24 hours, 7 days, or 30 days), **Granularity**, and **Value type** (Average or Peak) to filter data, as shown below:



Node Map Feature Description

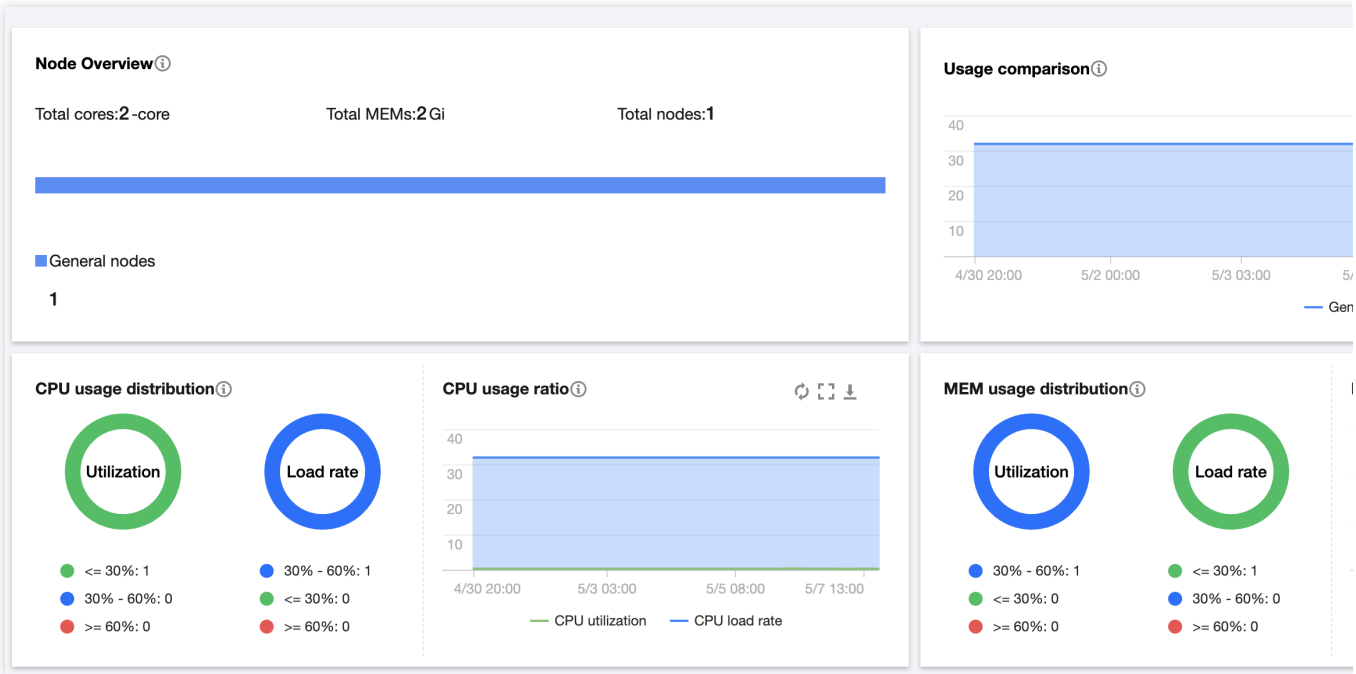
The Node Map page is divided into two parts: Overview and Resource Object Heat Map.



1. The upper part provides an overview of nodes in the cluster.
2. The lower part displays individual node resource objects in the cluster.

Cluster/Node Overview

The Cluster/Node Overview is as shown below:



Node Overview

Displays the total cores and total memory of nodes in the current cluster.

Displays the number of different types of nodes (native nodes, general nodes, and registered nodes) in the current cluster.

Note:

Super nodes do not conceptually count as nodes, so they are not covered in the statistics.

Node Usage Comparison

Compares the CPU load rate, memory load rate, CPU utilization, and memory utilization among different types of nodes (native nodes, general nodes, and registered nodes).

Node Resource Usage Distribution

Displays the distribution of CPU/memory usage among nodes (native nodes, general nodes, and registered nodes) in the cluster.

Note:

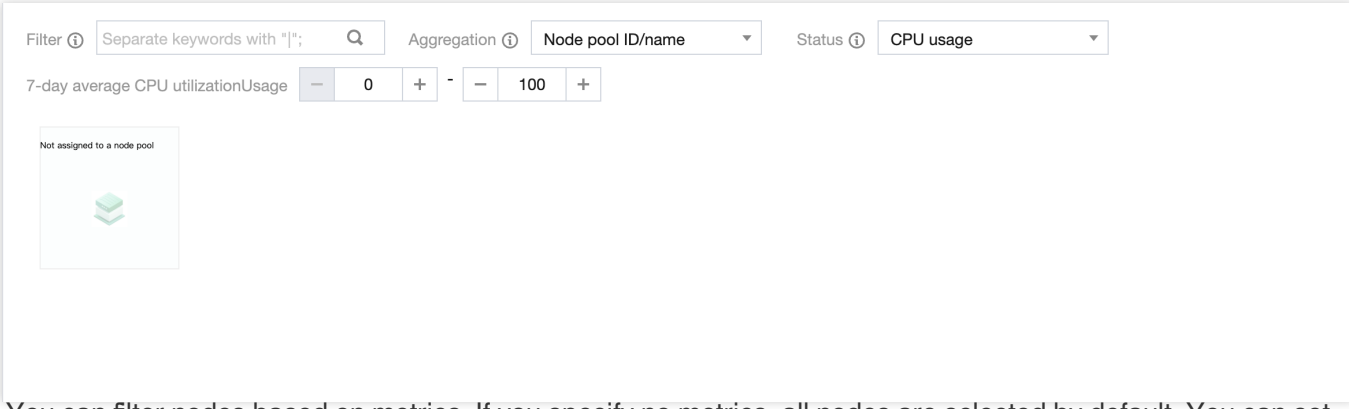
Super nodes do not conceptually count as nodes, so they are not covered in the statistics.

Metric	Description	Remarks
CPU utilization distribution	Shows the distribution of average or peak CPU utilization among nodes (native nodes, general nodes, and registered nodes) in the cluster in the last 24 hours, 7 days, or 30 days (as set in the upper right corner). Displays the number of nodes by the utilization range.	The CPU utilization of a node is defined as the actual CPU resource usage of the node divided by the node specification. Super nodes do not conceptually count as nodes, so they are not covered in the statistics.

CPU load rate distribution	Shows the distribution of average or peak CPU load rates among nodes in the cluster in the last 24 hours, 7 days, or 30 days (as set in the upper right corner). Displays the number of nodes by the load rate range.	The CPU load rate of a node is defined as the sum of CPU requests for all pods on the node divided by the node specification.
Memory utilization distribution	Shows the distribution of average or peak memory utilization among nodes in the cluster in the last 24 hours, 7 days, or 30 days (as set in the upper right corner). Displays the number of nodes by the utilization range.	The memory utilization of a node is defined as the actual memory resource usage of the node divided by the node specification.
Memory load rate distribution	Shows the distribution of average or peak memory load rates among nodes in the cluster in the last 24 hours, 7 days, or 30 days (as set in the upper right corner). Displays the number of nodes by the load rate range.	The memory load rate of a node is defined as the sum of memory requests for all pods on the node divided by the node specification.
CPU usage ratio	Shows the overall trend of CPU load rate and utilization in the cluster based on the period set in the upper right corner.	The utilization statistics covers native nodes, general nodes, registered nodes, and super nodes in the cluster. The load rate statistics covers native nodes, general nodes, and registered nodes in the cluster. Super nodes are not covered as they do not conceptually count as nodes.
Memory usage ratio	Shows the overall trend of memory load rate and utilization in the cluster based on the period set in the upper right corner.	The utilization statistics covers native nodes, general nodes, registered nodes, and super nodes in the cluster. The load rate statistics covers native nodes, general nodes, and registered nodes in the cluster. Super nodes are not covered as they do not conceptually count as nodes.

Viewing Node Objects

As shown below, you can filter nodes and aggregate node display by setting the **Filter**, **Aggregation**, and **Status** fields.



Filter: You can filter nodes based on metrics. If you specify no metrics, all nodes are selected by default. You can set multiple attributes, and the nodes meeting all the filter criteria are selected.

Aggregation: Filtered nodes are grouped. Nodes in the same group are shown in the same light-colored box and have the same attribute values.

Status: You can filter nodes by setting **Status** to **CPU utilization**, **Memory utilization**, **CPU load rate**, or **Memory load rate**.

CPU utilization: The average or peak node CPU utilization in the last 24 hours, 7 days, or 30 days.

Memory utilization: The average or peak node memory utilization in the last 24 hours, 7 days, or 30 days.

CPU load rate: The average or peak node CPU load rate in the last 24 hours, 7 days, or 30 days.

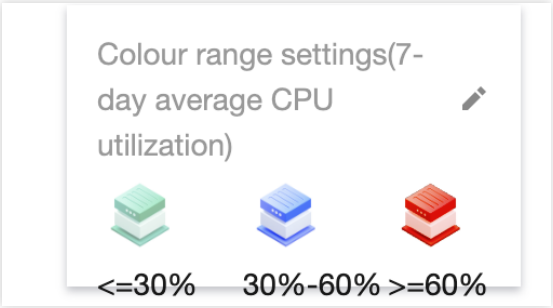
Memory load rate: The average or peak node memory load rate in the last 24 hours, 7 days, or 30 days.

Note:

In the node object heat map, nodes are by default **sorted in ascending order** according to the current **Status** setting. For example, if you set **Status** to **CPU utilization**, the nodes in the node heat map are by default arranged in ascending order of their average or peak CPU utilization in a time range (the **Period** set in the upper right corner).

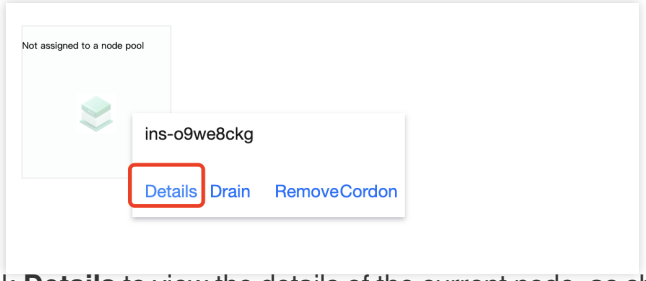
Filtered nodes are distinguished by three colors: **green**, **blue**, and **red**. You can click

in the lower right corner of the page to adjust the threshold ranges for the three statuses, as shown below:

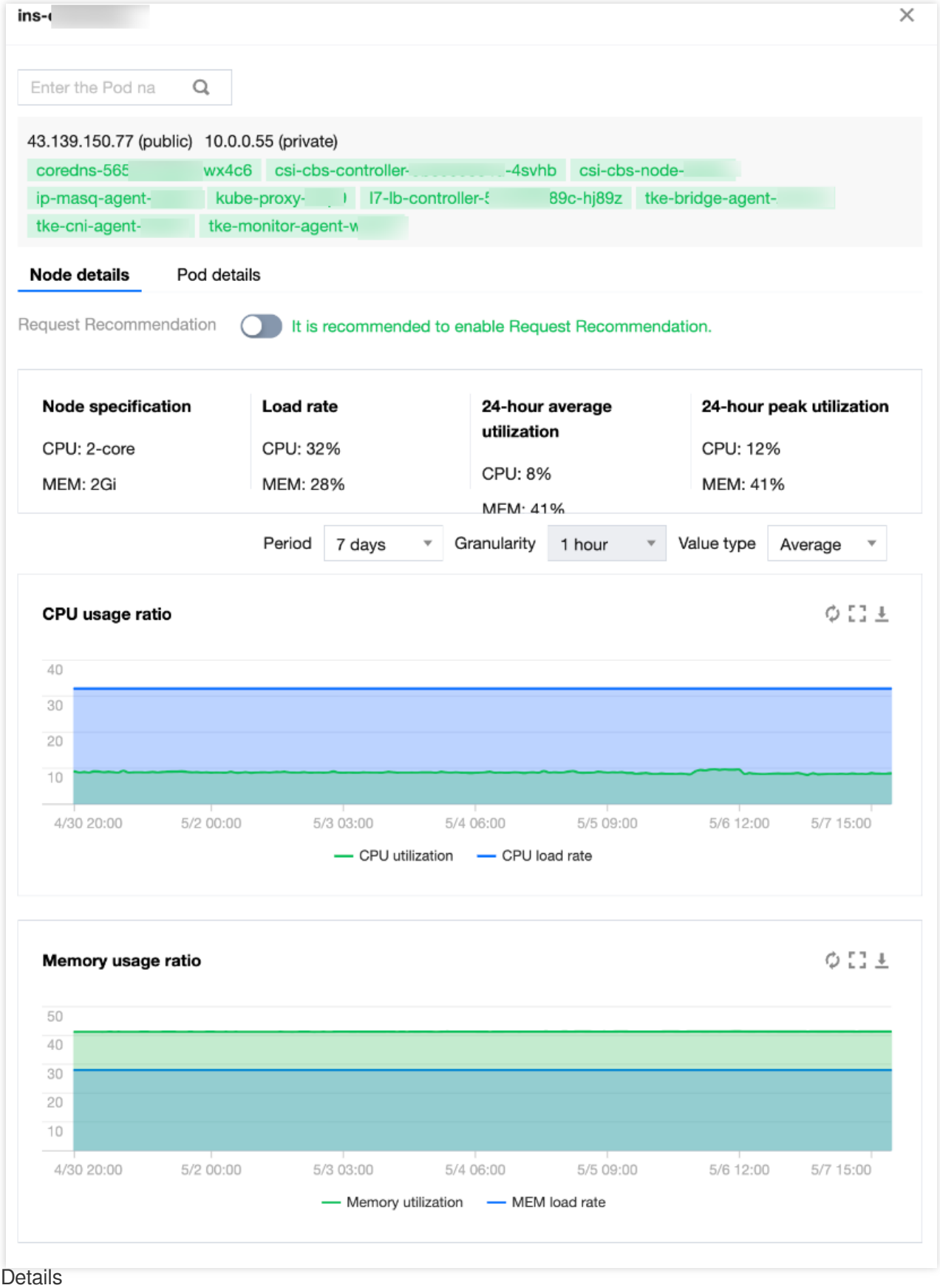


Node Operation

When you hover your mouse over a node object, you can view the details of the node and perform the following operations:



1. Click **Details** to view the details of the current node, as shown below:



Node Details

Pod Details

Feature Switches

Request Recommendation: A cluster-level feature switch. If the switch is enabled, suitable requests are recommended for workloads based on workload usage. For details, refer to Request Recommendation.
Native Node Dedicated Scheduler: A cluster-level feature switch. If the switch is enabled, the specifications of native nodes can be virtually increased to schedule more pods. For details, refer to Native Node Dedicated Scheduler.
Node Information
Node specification: The CPU and memory capacity of the node.
Load rate: The sum of requests for all pods on the node divided by the node specification (based on the setting in the upper right corner of the page).
24-hour average utilization: Average CPU and memory utilization of the node in the last 24 hours.
24-hour peak utilization: Peak CPU and memory utilization of the node in the last 24 hours.
Node Trend Chart Information
CPU usage ratio:
The trend of the CPU usage ratio of the node in the cluster in the specified period (as set in the upper right corner).
Memory usage ratio:
The trend of the memory usage ratio of the node in the cluster in the specified period (as set in the upper right corner).
CPU Usage Trend Chart
Request: The requested usage of the pod.
Usage: The actual usage of the pod.
Recommendation: The recommended usage of the pod. To receive recommendation, you need to enable Request Recommendation in advance.
Pod Information
Pod name: The name of the pod.
Pod status: The status of the pod.
Workload: The workload to which the current pod belongs.
Request: The text in black indicates the request configuration for the current pod. The text in blue indicates the request recommendation for the current pod, which requires enabling the Request Recommendation Component.
Note:
1. By clicking the request recommendation in blue, you can update the request value of the deployment to which the pod belongs.
2. This one-click update capability is exclusive to native nodes. If some pods of the workload are not on native nodes, you need to migrate the pods to native nodes first. For details, refer to Request Recommendation.
2. Click **Drain** to drain the pod on the node to another node in the cluster.
3. Click **Remove** to remove the node from the node pool.
If the node is in a node pool, the current node is managed by the node pool and needs to be processed on the node pool page.
If the node is not in a node pool, the node can be removed from the cluster. If the node is a pay-as-you-go node, it can be canceled when you remove it.
4. Click **Decordon** to schedule the node for a new pod.

Note:

A node is automatically cordoned once drained. After you click **Decordon**, the node can continue to be scheduled for a new pod.

Workload Map

Last updated：2023-05-06 17:36:46

Overview

Traditional object management webpages typically use a list-based format, as exemplified by the resource object list for TKE clusters in the TKE console. However, this approach suffers from several limitations, including low readability due to abstract numerical values, a lack of clear object identification, and the inability to support certain sorting methods. To break these limits, TKE has introduced Workload Map, a cloud-native asset management platform that displays all resource objects of users in a visualized way. This platform provides rich filtering, aggregation, and status display features that enable users to rapidly locate their required objects.
Workload Map displays the status and metrics of workloads by using visualized charts and diagrams on webpage. This assists users in comprehending the configuration and usage of their current workloads, as well as in identifying any potential issues.

Overview

Workload Map does not offer monitoring data for recommended Pod values on super nodes. When Workload Map aggregates monitoring metrics, if any Pod in a workload resides on a super node, statistical data for recommended values at the workload and cluster levels will be incomplete.

Directions

1. Log in to the TKE console.
2. Choose **Cloud native asset management** > **Workload Map** in the left sidebar.
3. On the Workload Map page, select **Region**, **Cluster type** and **Cluster** options as needed.

Workload Map Features

The Workload Map page is divided into two parts: Overview and Resource Object Heat Map.
The upper part provides an overview of the workloads in the selected cluster.
1. The lower part displays individual workload resource objects in the current cluster.

Workload overview

Workload Overview: Displays the number and percentage of workloads in the current cluster that have recommended request values, as well as the total number of CPU cores and memory in nodes in the current cluster.

Note

You need to enable the Request Recommendation feature in order to receive recommended request values for adjusting workloads.

CPU Usage: Shows the CPU utilization of workloads in the cluster.

Request: The sum of CPU requests for all Pods in all workloads in the cluster.

Usage: The sum of actual CPU utilization for all Pods in all workloads in the cluster.

Recommendation: The sum of recommended CPU utilization for all Pods in all workloads in the cluster. To receive this value, Request Recommendation needs to be enabled beforehand.

Memory Usage: Shows the memory usage of workloads in the cluster.

Request: The sum of memory requests for all Pods in all workloads in the cluster.

Usage: The sum of actual memory usage for all Pods in all workloads in the cluster.

Recommendation: The sum of recommended memory usage for all Pods in all workloads in the cluster. To receive this value, Request Recommendation needs to be enabled beforehand.

Viewing workloads

You can filter workloads by metric or by status and aggregate workloads.

Filter by metric: You can filter workloads by relevant metrics. If you specify no metrics, all workloads are selected by default. You can filter workloads by multiple metrics, and the intersection of those filters is selected.

Aggregate: The filtered workloads are grouped. Workloads in the same group are shown in the same light-colored box and have the same metric value.

Status: You can filter workloads based on their status, such as the **peak CPU utilization**, **average CPU utilization**, and **CPU packing density**.

Peak CPU utilization: Shows the peak CPU utilization of a workload, with metrics available for the last 24 hours, 7 days, and 30 days.

Average CPU utilization: Shows the average CPU utilization of a node, with metrics available for the last 24 hours, 7 days, and 30 days.

CPU packing density: Shows the average CPU packing density of a node, with metrics available for the last 24 hours, 7 days, and 30 days.

Note

In the Workload Object Heat Map, nodes are sorted in ascending order by default, based on the currently selected "status" attribute value. For instance, if you choose "24-hour average CPU utilization" as the status attribute, workloads are displayed in ascending order based on their 24-hour average CPU utilization.

Filtered workloads are distinguished by three colors: green, blue, and red. To adjust the threshold ranges for these three display colors, you can click


in the bottom-right corner of the page.

Download: You can click the


icon on the right of Workload Overview to download the workload information list on this page.

Description of fields in the list:

Name: Name of the workload.

Namespace: Namespace to which the workload belongs.

Workload type: Kubernetes type to which the workload belongs.

24-hour average CPU utilization: Average CPU utilization (%) of the workload in the last 24 hours.

7-day average CPU utilization: Average CPU utilization (%) of the workload in the last 7 days.

30-day average CPU utilization: Average CPU utilization (%) of the workload in the last 30 days.

24-hour peak CPU utilization: Peak CPU utilization (%) of the workload in the last 24 hours.

7-day peak CPU utilization: Peak CPU utilization (%) of the workload in the last 7 days.

30-day peak CPU utilization: Peak CPU utilization (%) of the workload in the last 30 days.

24-hour average memory usage: Average memory usage (%) of the workload in the last 24 hours.

7-day average memory usage: Average memory usage (%) of the workload in the last 7 days.

30-day average memory usage: Average memory usage (%) of the workload in the last 30 days.

24-hour peak memory usage: Peak memory usage (%) of the workload in the last 24 hours.

7-day peak memory usage: Peak memory usage (%) of the workload in the last 7 days.

30-day peak memory usage: Peak memory usage (%) of the workload in the last 30 days.

Workload operations

By hovering over a workload, you can view the details of the current workload and perform the following operations:

1. Click **Details** to view the details about the current workload.

Workload details

Pod details

Feature switches

Request Recommendation: Cluster-level feature switch. If the switch is turned on, suitable request values are recommended for workloads based on their resource usage.

Recommended Value (in blue font): Indicates the recommended request value for the current workload. You can update this value by clicking the switch. The value is displayed only when Request Recommendation is enabled and there is a recommended value.

Workload information

24-hour Average Utilization: Average CPU and memory utilization of the workload in the last 24 hours.

24-hour Peak Utilization: Peak CPU and memory utilization of the workload in the last 24 hours.

Workload chart information

CPU Usage: CPU utilization of the current workload.

Request: Sum of requested CPU for all Pods in the current workload.

Usage: Sum of actual CPU utilization for all Pods in the current workload.

Recommendation: Sum of recommended CPU utilization for all Pods in the current workload. To receive this value, Request Recommendation needs to be enabled beforehand.

Memory Usage: Memory usage of the current workload.

Request: Sum of requested memory for all Pods in the current workload.

Usage: Sum of actual memory usage for all Pods in the current workload.

Recommendation: Sum of recommended memory usage for all Pods in the current workload. To receive this value, Request Recommendation needs to be enabled beforehand.

CPU usage chart

Request: Number of requested Pods.

Usage: Number of Pods actually used.

Recommendation: Recommended usage of Pods.

Pod details

Pod name: Name of the Pod.

Pod status: Status of the Pod.

Node: Node to which the current Pod belongs.

Node type: Type of the node to which the current Pod belongs.

2. Click **Recommend** to show the recommended request value for the current workload. This button is available only when Request Recommendation is enabled and there is a recommended value.

3. Click **Delete** to delete the workload.