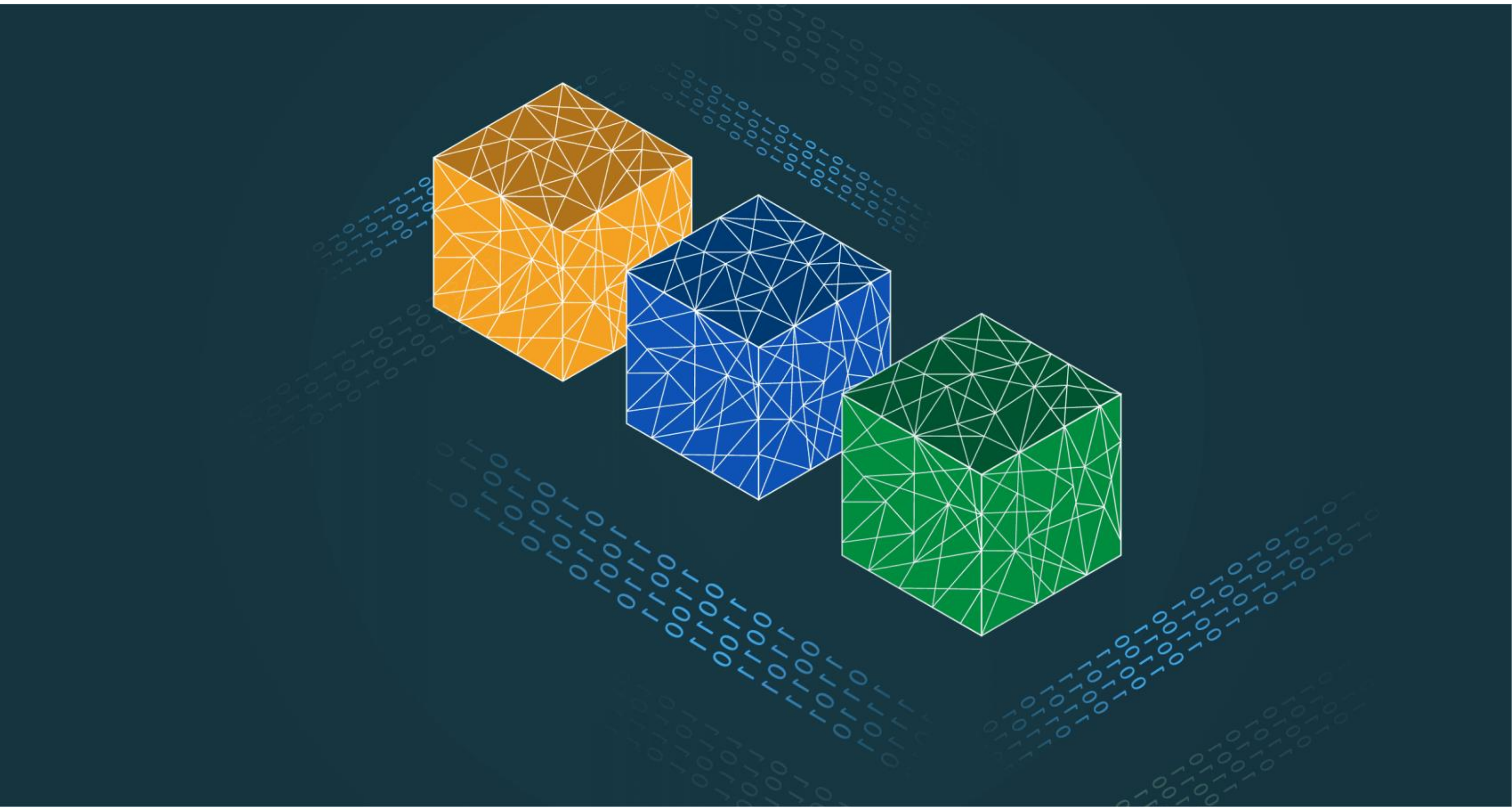
ProphetStor



Federator.ai User Guide

Federator.ai version 5.0 User Guide

ProphetStor Data Services, Inc.
830 Hillview Court, Suite 100
Milpitas, CA 95035 USA
Phone: 1.408.508.6255
Website: www.prophetstor.com

1.19.2022

Contents

Overview

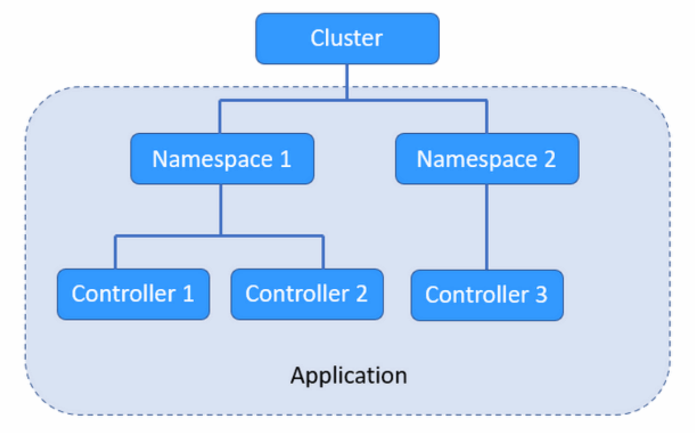
ProphetStor Federator.ai is an AI-based solution that helps enterprises manage and optimize resources for applications on Kubernetes and virtual machines (VMs) in VMware clusters. Using advanced machine learning algorithms to predict application workloads, Federator.ai offers:

- AI-based workload prediction for containerized applications in Kubernetes clusters as well as VMs in VMware clusters and Amazon Web Services (AWS) Elastic Compute Cloud (EC2)
- Resource recommendations based on workload prediction, application, Kubernetes, and other related metrics
- Automatic provisioning of CPU/memory for generic Kubernetes application controllers/namespaces
- Correlation and causality analysis of microservices/controllers of Kubernetes applications
- Automatic scaling of Kubernetes application containers, Kafka consumer groups, and Ingress upstream services
- Multicloud cost analysis and recommendations based on workload predictions for Kubernetes clusters and VM clusters
- Actual cost and potential savings based on recommendations for clusters, Kubernetes applications, VMs, and Kubernetes namespaces

If you have not installed Federator.ai yet, refer to your *Federator.ai Installation Guide* for information.

Terminology

Application – Defined by Federator.ai as a group of Kubernetes controllers that work together to serve tasks from the view of the end user. For example, an e-commerce web application consists of controllers for frontend and backend and can be considered as an application. An application is not a Kubernetes object.



Auto Provisioning – The ability to automatically deploy CPU and memory resource recommendations to controllers and namespaces of generic applications in Kubernetes clusters based on pre-defined profiles.

Autoscaling – See HPA below.

Auto Scaling group – (ASG) A collection of Amazon EC2 instances that are treated as a logical grouping for automatic scaling and management.

Container – An object that contains a software module with everything needed to run an application.

Controller – In Kubernetes, controllers are control loops that watch the state of your cluster, then make or request changes where needed. Each controller tries to move the current cluster state closer to the desired state. The types of controllers supported by Federator.ai are *Deployment* and *StatefulSet*. Additionally, Federator.ai supports *DeploymentConfig* controllers for OpenShift.

Cluster – A Kubernetes cluster with one or more nodes or a VM cluster with one or more VMs.

Deployment – A Deployment provides declarative updates for Pods and ReplicaSets. The user describes a desired state in a deployment and the deployment controller changes the actual state to the desired state at a controlled rate.

HPA – Horizontal Pod Autoscaling – In Kubernetes, the system automatically increases or decreases the number of containers/pods (replicas) based on the workload. For generic applications, autoscaling is based on CPU/memory usage. For Kafka consumers or Ingress upstream services, application-aware autoscaling uses metrics provided by the application.

Namespace – Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces.

Microservice – Also known as controllers, microservices are independent, modular Kubernetes components that work together as a single application.

Node – In Kubernetes, nodes are server-like machines, such as a virtual machine running complete systems and multiple applications. There can be master nodes and worker nodes.

Pod – A group of one or more containers with shared storage/network resources and a specification for how to run the containers. Typically, one container runs in each pod.

Replica – A copy of a pod running for an application.

StatefulSet – A Kubernetes object that manages stateful applications. Unlike a Deployment, a StatefulSet maintains a sticky identity for each of its pods that remains the same across any rescheduling.

VM – A VMware virtual machine running on a physical *host* machine.

VM Cluster – A cluster with one or more VMs.

Related topics:

Federator.ai Administration Portal

Configure Kubernetes Clusters

Configure VMWare Clusters

Configure Applications

Auto Provisioning

Getting Started

After installation of Federator.ai, you must access the Federator.ai portal in order to configure your system.

Access the Federator.ai Portal

To access the Federator.ai administration portal, use the URL that is displayed at the end of the installation process.

You can also find the URL for the Federator.ai administration portal via the following methods:

Kubernetes

In a Kubernetes environment, use the `kubectl` command to find the administration portal service port number and node IP address.

```
# kubectl get svc -n federatorai |grep federatorai-dashboard-frontend-node-port
```

The output will look something like this:

```
federatorai-dashboard-frontend-node-port NodePort 10.103.181.133 <none> 9001:31012/TCP
```

Get the node's IP to access (INTERNAL-IP).

```
$kubectl get nodes -o wide
```

For example:

```
# kubectl get nodes -o wide
NAME      STATUS    ROLES     AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION          CONTAINER-RUNTIME
h7-130    Ready     master    35d   v1.18.5   172.31.7.130   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
h7-131    Ready     <none>    35d   v1.18.5   172.31.7.131   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
h7-132    Ready     <none>    35d   v1.18.5   172.31.7.132   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
h7-133    Ready     <none>    35d   v1.18.5   172.31.7.133   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
```

The URL will be `https://172.31.7.130:31012`.

OpenShift

In an OpenShift environment, use the `oc get route` command to find the URL.

```
# oc get route -n federatorai|grep federatorai-dashboard-frontend
```

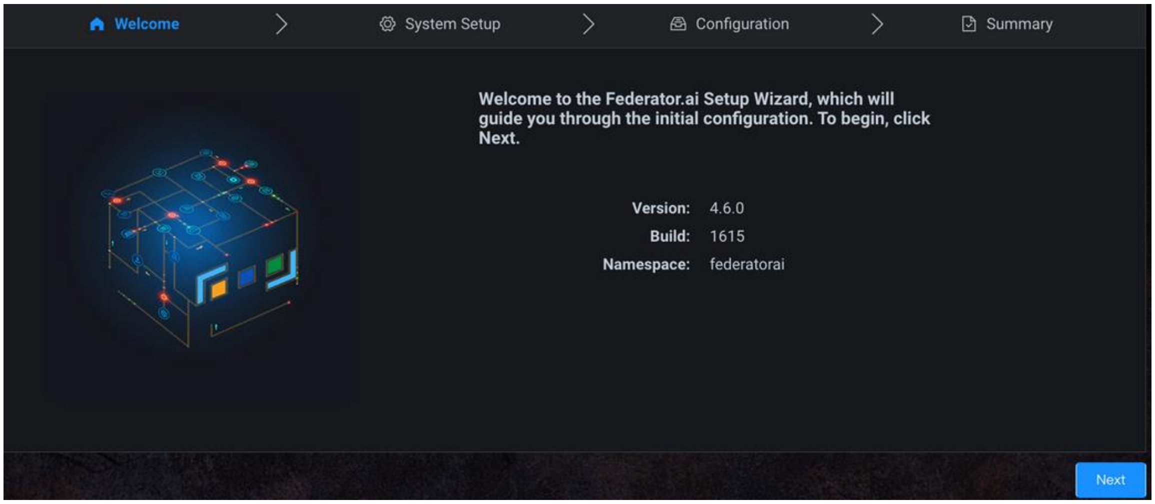
The output will look something like this:

```
federatorai-dashboard-frontend “federatorai-dashboard-frontend-federatorai.apps.ocp4.172-31-11-30.nip.io”
```

The URL will be `https://federatorai-dashboard-frontend-federatorai.apps.ocp4.172-31-11-30.nip.io`

Setup Wizard

The first time you log in after installation of Federator.ai, a setup wizard launches that allows you to configure a cluster that should be monitored by Federator.ai. You can add more clusters from the *Configuration* section of the portal after completing the setup wizard. Click *Next* to begin configuration.

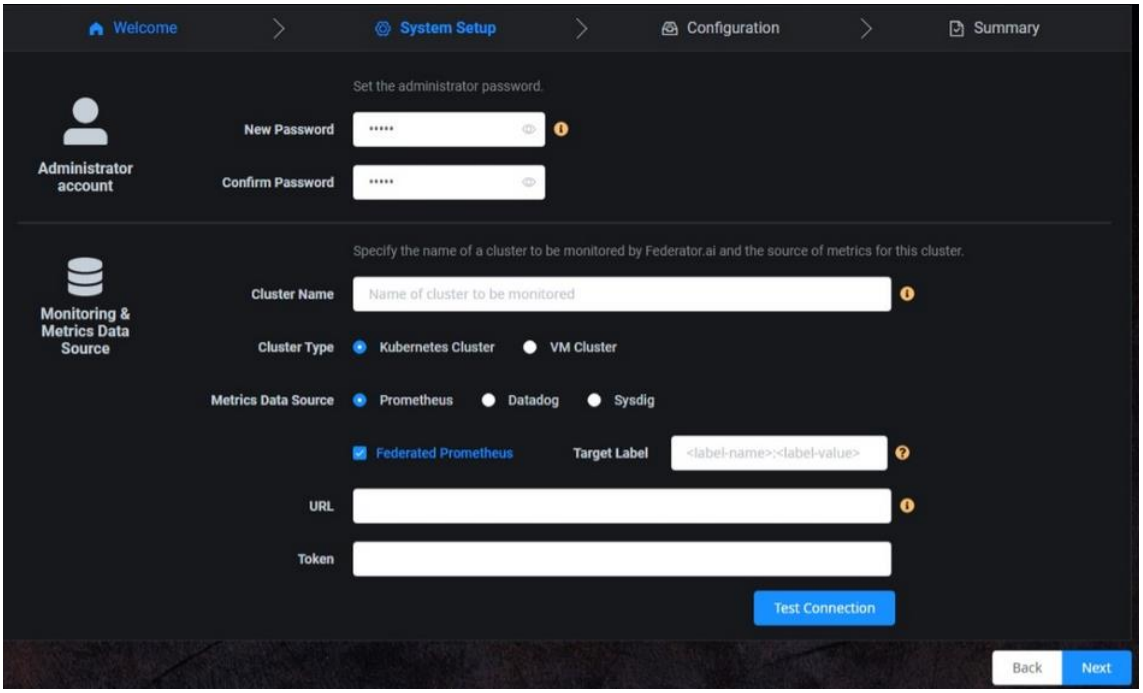


Welcome to the Federator.ai Setup Wizard, which will guide you through the initial configuration. To begin, click Next.

Version:	4.6.0
Build:	1615
Namespace:	federatorai

Kubernetes Cluster

1. Set the administrator password and the name of a cluster to be monitored by Federator.ai, select *Kubernetes Cluster* and the source of metrics for this cluster, and specify authentication information. Then, click *Test Connection* to confirm that all information is correct. Click *Next* when the system can connect to the cluster.



Set the administrator password.

New Password: •••••

Confirm Password: •••••

Specify the name of a cluster to be monitored by Federator.ai and the source of metrics for this cluster.

Cluster Name: Name of cluster to be monitored

Cluster Type: ● Kubernetes Cluster ○ VM Cluster

Metrics Data Source: ● Prometheus ○ Datadog ○ Sysdig

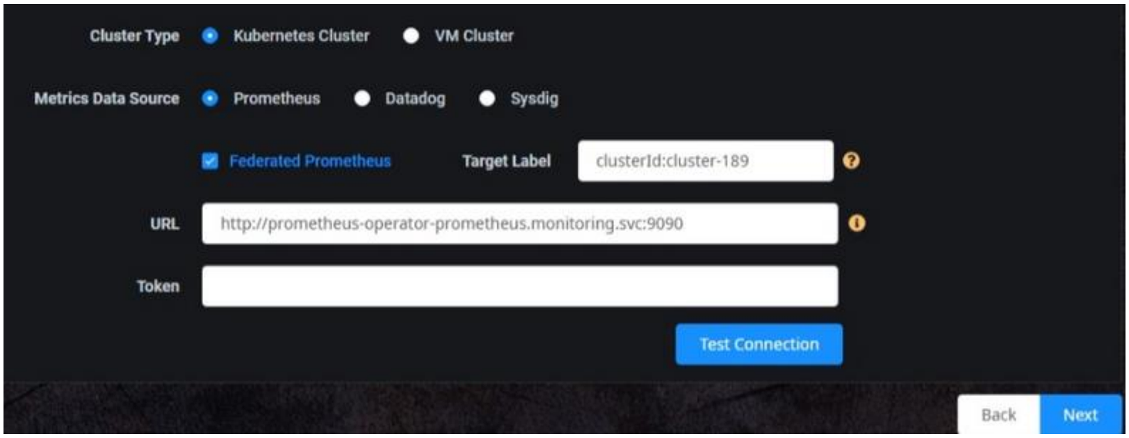
☑ Federated Prometheus Target Label: <label-name>:<label-value>

URL:

Token:

You must set the administrator password in order to continue but your cluster can be configured later from the *Configuration* section of the portal after completing the setup wizard.

For the Prometheus open-source monitoring system, the URL is required but the token is optional for authentication. Specify if you are using Federation, which is a group of Prometheus servers that send metrics to a centralized Prometheus server. You will need to specify the target label of the centralized Prometheus server. The format is: `<label-name>:<label-value>` (e.g., clusterID:host-1).



Cluster Type ● Kubernetes Cluster ○ VM Cluster

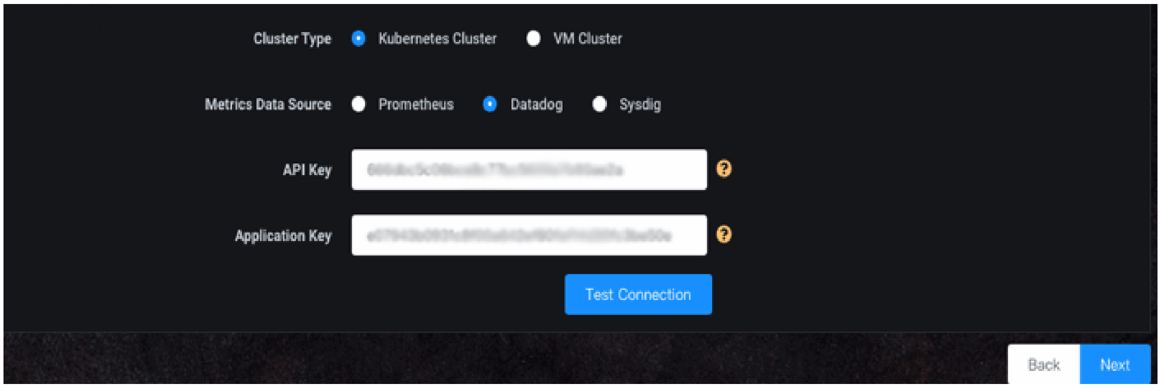
Metrics Data Source ● Prometheus ○ Datadog ○ Sysdig

☑ Federated Prometheus Target Label: clusterId:cluster-189

URL: http://prometheus-operator-prometheus.monitoring.svc:9090

Token:

For Datadog, the API key and application key are required for authentication. If needed, you can click on the link to the Datadog website, which is included in the popup help text.



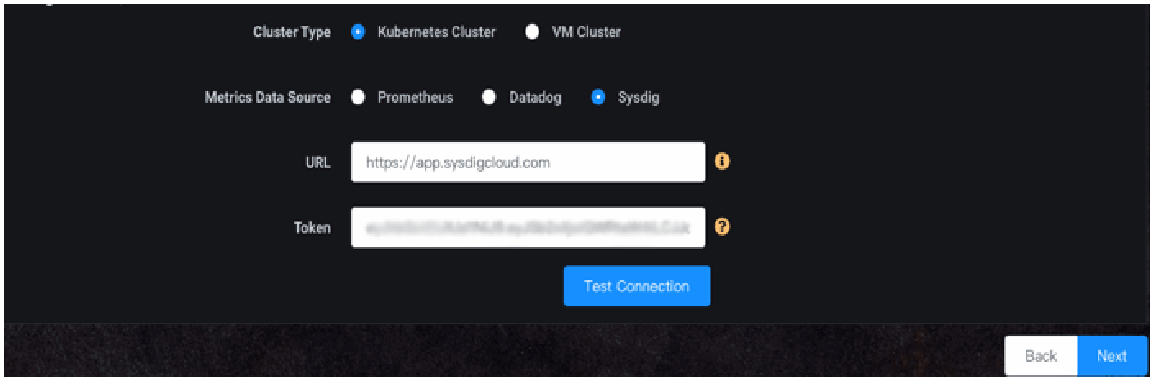
Cluster Type ● Kubernetes Cluster ○ VM Cluster

Metrics Data Source ○ Prometheus ● Datadog ○ Sysdig

API Key: [redacted]

Application Key: [redacted]

For Sysdig, a URL and token are required for authentication. If needed, you can click on the link to the Sysdig website, which is included in the popup help text.



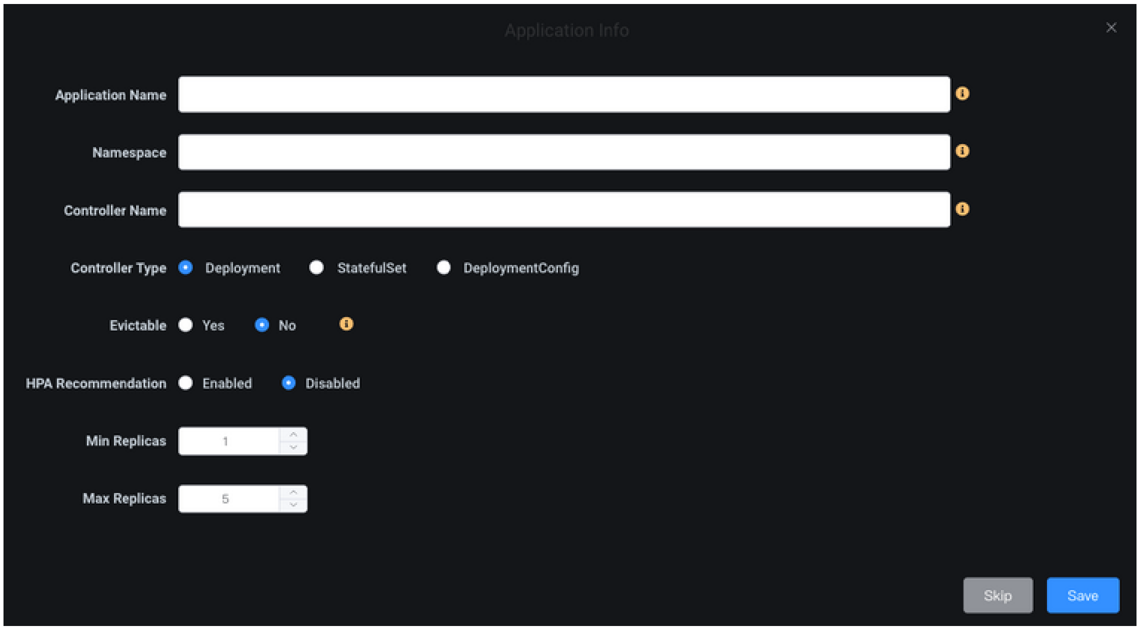
Cluster Type ● Kubernetes Cluster ○ VM Cluster

Metrics Data Source ○ Prometheus ○ Datadog ● Sysdig

URL: https://app.sysdigcloud.com

Token: [redacted]

2. Enter information about the first application you want to monitor. You can skip this step and configure applications to be monitored from the *Configuration* section of the portal.



Application Info

Application Name

Namespace

Controller Name

Controller Type ● Deployment ○ StatefulSet ○ DeploymentConfig

Evictable ○ Yes ● No

HPA Recommendation ○ Enabled ● Disabled

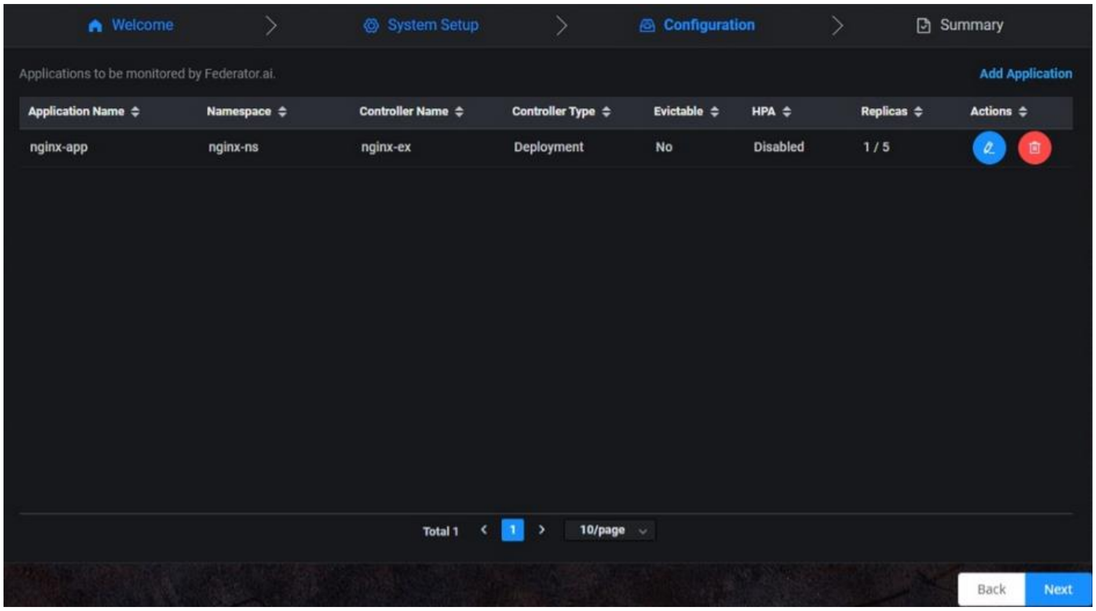
Min Replicas 1

Max Replicas 5

You can add applications now or from the *Configuration* section of the portal after completing the setup wizard.

- *Application Name* - The name of your application to be monitored by Federator.ai. An application is a group of one or more Kubernetes controllers that work together to serve tasks from the view of the end user; an application is not a Kubernetes object.
- *Namespace* – The Kubernetes namespace where the controller is deployed.
- *Controller Name* - The name of controller to be monitored.
- *Controller Type* – Supported controller types are *Deployment, StatefulSet,* and *DeploymentConfig* (OpenShift only).
- *Evictable* – Indicate if the controller can be interrupted if the node is shut down. Evictable controllers are good candidates to be deployed in Spot instances.
- *HPA Recommendation* – Indicate if you want to enable Horizontal Pod Autoscaling (HPA). When enabled, CPU and memory usage is monitored, and the number of pods is automatically increased/decreased based on the CPU/memory usage workload. HPA and Auto Provisioning are mutually exclusive; you can use HPA or auto provisioning, but not both.
- *Mix/Max Replicas* – Specify the minimum and maximum number of pods when HPA is enabled.

3. Click *Save* to view a list of applications that will be monitored by Federator.ai.



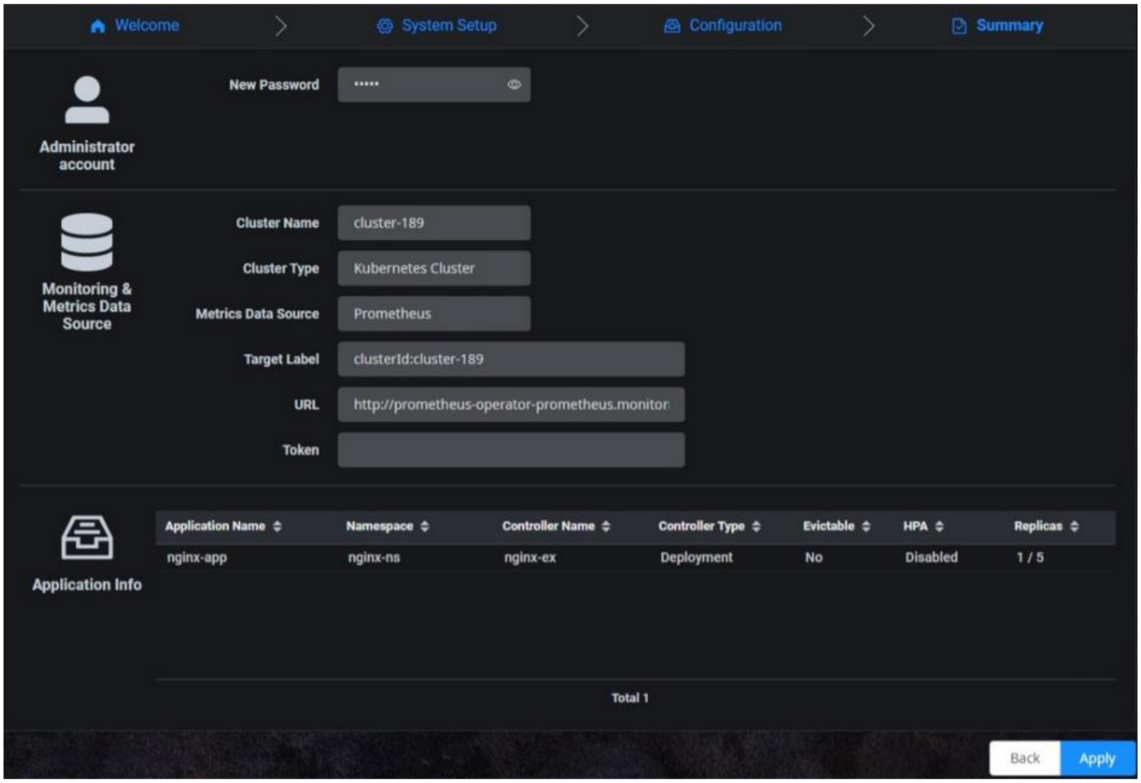
Applications to be monitored by Federator.ai.

Application Name	Namespace	Controller Name	Controller Type	Evictable	HPA	Replicas	Actions
nginx-app	nginx-ns	nginx-ex	Deployment	No	Disabled	1 / 5	

Click the + key to add an application.

Click the blue *Edit* icon to update application information or click the red *Delete* icon to remove an application.

4. Click *Next* to view the *Summary* screen.



New Password: •••••

Cluster Name: cluster-189

Cluster Type: Kubernetes Cluster

Metrics Data Source: Prometheus

Target Label: clusterId:cluster-189

URL: http://prometheus-operator-prometheus.monitor

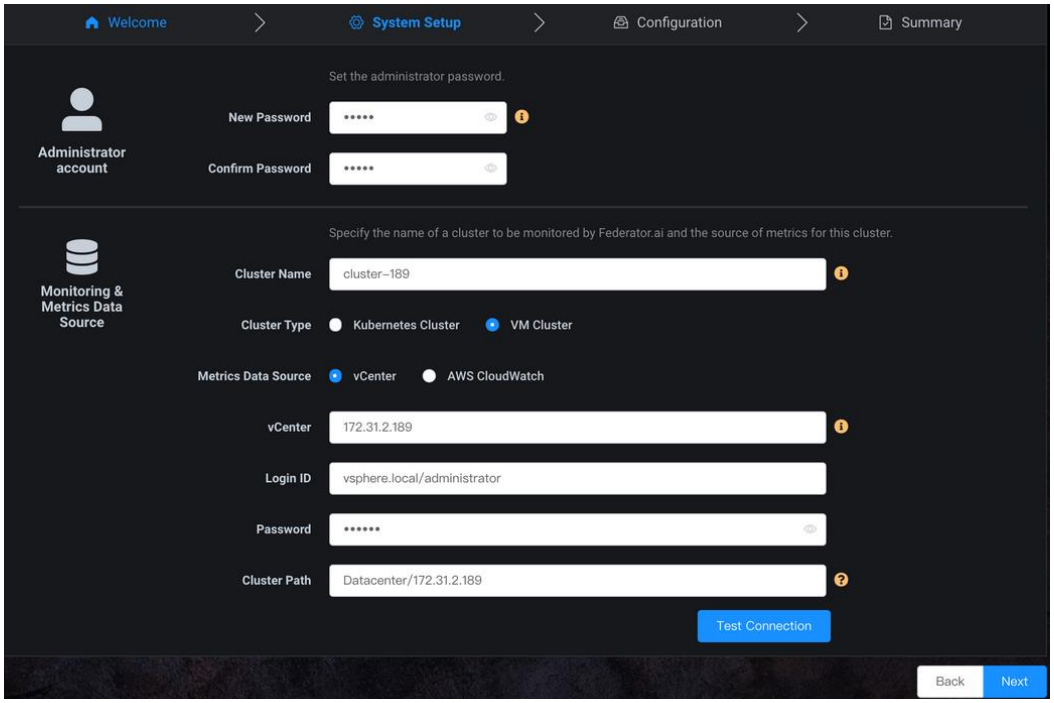
Token:

Application Name	Namespace	Controller Name	Controller Type	Evictable	HPA	Replicas
nginx-app	nginx-ns	nginx-ex	Deployment	No	Disabled	1 / 5

5. Click *Apply* to apply all changes or click *Back* to edit information.

VM Cluster

1. Set the administrator password and the name of a cluster to be monitored by Federator.ai, select *VM Cluster* and *vCenter* or *AWS CloudWatch* as the source of metrics for this cluster, and specify connection information.



Set the administrator password.

New Password •••••

Confirm Password •••••

Specify the name of a cluster to be monitored by Federator.ai and the source of metrics for this cluster.

Cluster Name cluster–189

Cluster Type ○ Kubernetes Cluster ● VM Cluster

Metrics Data Source ● vCenter ○ AWS CloudWatch

vCenter 172.31.2.189

Login ID vsphere.local/administrator

Password ••••••

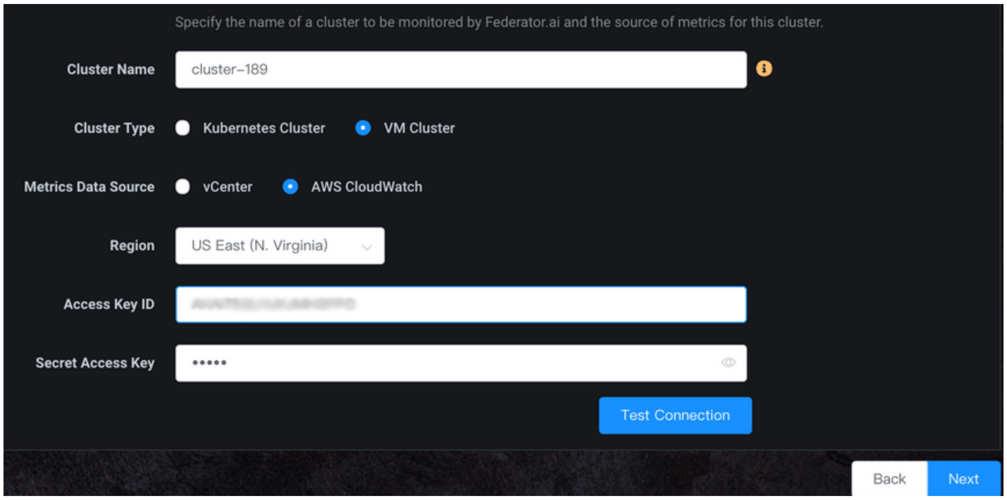
Cluster Path Datacenter/172.31.2.189

You must set the administrator password in order to continue but your cluster can be configured later from the *Configuration* section of the portal after completing the setup wizard.

The cluster name must have a maximum of 253 lowercase characters,"-", or "." allowed. The name must start and end with an alphanumeric character.

For vCenter, specify the vCenter IP address (you can have multiple vCenters in your system), login ID, password, and the path to the cluster, within vCenter. If needed, you can click on the link to the vCenter website, which is included in the popup help text.

For AWS CloudWatch, specify the region of Amazon AWS S3 service, the AWS Identity and Access Management key ID (16 to 128 bytes), and the secret access key of the key ID that is used for access. Note: The CloudWatch agent must be installed on the EC2 node in order to use this data source.



Specify the name of a cluster to be monitored by Federator.ai and the source of metrics for this cluster.

Cluster Name: cluster–189

Cluster Type: ○ Kubernetes Cluster ● VM Cluster

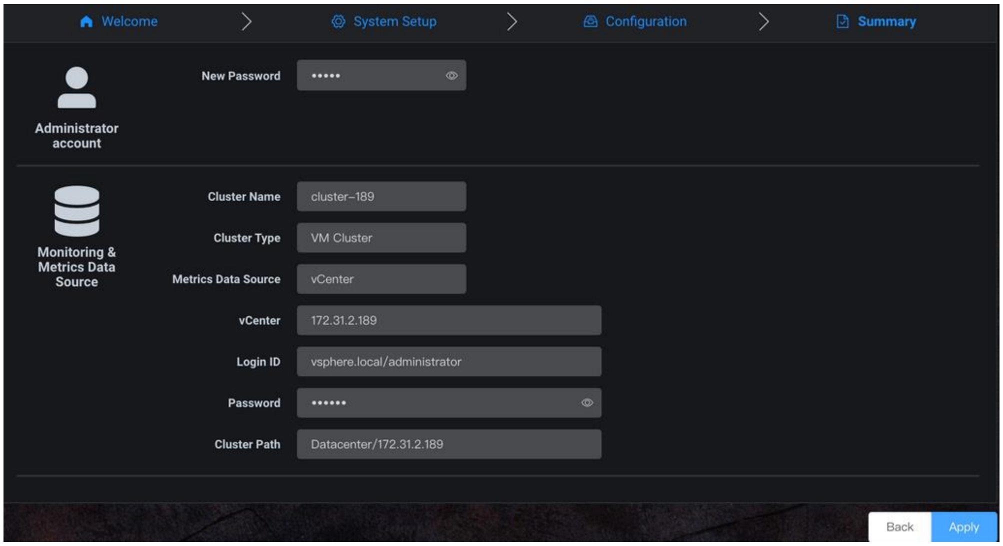
Metrics Data Source: ○ vCenter ● AWS CloudWatch

Region: US East (N. Virginia)

Access Key ID: [redacted]

Secret Access Key: •••••

2. Click *Test Connection* to confirm that all information is correct.
3. Click *Next* to view the *Summary* screen when the system can connect to the cluster.



New Password: •••••

Cluster Name: cluster–189

Cluster Type: VM Cluster

Metrics Data Source: vCenter

vCenter: 172.31.2.189

Login ID: vsphere.local/administrator

Password: ••••••

Cluster Path: Datacenter/172.31.2.189

The system will discover the Virtual Machines (VMs) in this cluster, which may take a few minutes to complete. Use the *Edit Cluster* function on the *Configuration / Clusters* page to select which VMs that you want to monitor.

Related topics:

Terminology

Federator.ai Administration Portal

Configure Applications

Configure Kubernetes Clusters

Configure VM Clusters

Federator.ai Administration Portal

The Federator.ai administration portal displays the overall health of each cluster, as well as application workload and resource recommendations. Information is presented in tables and charts.

Portal Sections

The portal is separated into the following sections:

- Dashboard – Overall system information, including the number of monitored resources, as well as cluster and Kubernetes application workload predictions and recommendations.
- Application Analysis - Provides statistical analysis and predictions based on the correlation between resource usage and application workload for Kubernetes.
- Cluster Overview - Cluster and Kubernetes node health information, including CPU utilization, memory utilization, disk capacity, and VM network throughput.
- Predictions and Planning – Forecasting tools, including actual CPU and memory usage observations, predicted workload usage, utilization analysis, and recommendations for Kubernetes and VM resources.
- Autoscaling - (Kubernetes) Displays HPA recommendations for controllers enabled with autoscaling and autoscaling predictions for each Kafka individual topic/consumer group and Ingress upstream service
- Cost – Cost analysis, cost trends, and cost optimization for resources at different levels, including Kubernetes and VM clusters and nodes, as well as Kubernetes namespaces and applications.
- Configuration – Configuration of clusters, Kubernetes applications and controllers/consumer groups, as well as system configuration, including resetting the admin password, metrics data source, system notifications, licensing, and price books.
- Events – System events that have occurred.

Portal Icons

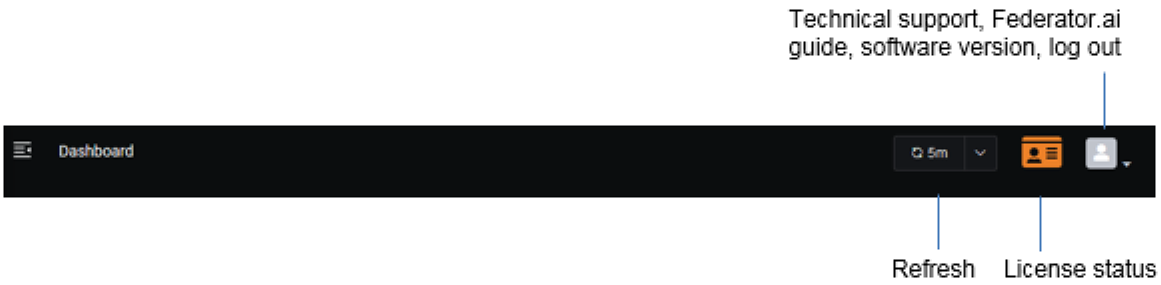
To make it easy to distinguish between cluster types and providers, the following icons are used throughout the portal:

<table>
<tr><th>Icon</th><th>Meaning</th></tr>
<tr><td>▦</td><td>On-premises provider</td></tr>
<tr><td>☸</td><td>Kubernetes clusters</td></tr>
<tr><td>▤</td><td>VM clusters. The cluster type is:
<table>
<tr><td>✥</td><td>AWS CloudWatch VM cluster configured with AWS Auto Scaling groups.</td></tr>
<tr><td>aws</td><td>AWS CloudWatch VM cluster with individual VMs.</td></tr>
<tr><td>▤</td><td>vCenter cluster.</td></tr>
</table>
</td></tr>
</table>

Common Administration Portal Functions

The administration portal presents information in tables and charts. At the top right of each portal page, you can do the following:

- Refresh statistics
- Check license status
- Get technical support contact information
- View Federator.ai product documentation
- Display the product software version
- Log out


Technical support, Federator.ai guide, software version, log out
Dashboard
5m
Refresh
License status

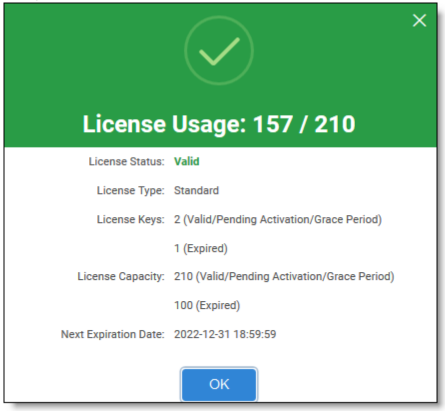
Refresh Statistics

By default, Federator.ai information is refreshed every five minutes. To change the interval, click the drop-down at the top right and select a 1, 5, 15, or 30-minute interval.

To force a refresh, click *Refresh Now* where the current interval is displayed.

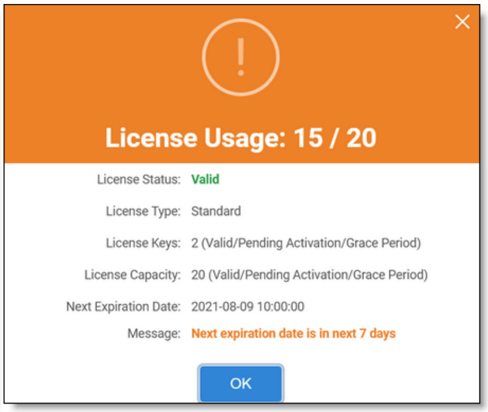
License Status

Click the *License* icon at the top right of the dashboard to see Federator.ai license information, including license status and type, number and type of license keys, licensed capacity and usage, as well as license expiration. When the icon is green, all licenses are valid. Orange indicates a situation that requires attentions, such as a license is near expiration, a license is in a grace period, or the number of monitored resources exceeds the license limit. Red requires immediate attention because it indicates a license has expired.



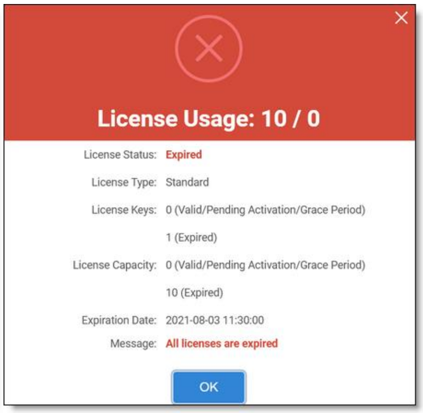
License Usage: 157 / 210

License Status: Valid
License Type: Standard
License Keys: 2 (Valid/Pending Activation/Grace Period)
1 (Expired)
License Capacity: 210 (Valid/Pending Activation/Grace Period)
100 (Expired)
Next Expiration Date: 2022-12-31 18:59:59



License Usage: 15 / 20

License Status: Valid
License Type: Standard
License Keys: 2 (Valid/Pending Activation/Grace Period)
License Capacity: 20 (Valid/Pending Activation/Grace Period)
Next Expiration Date: 2021-08-09 10:00:00
Message: Next expiration date is in next 7 days



License Usage: 10 / 0

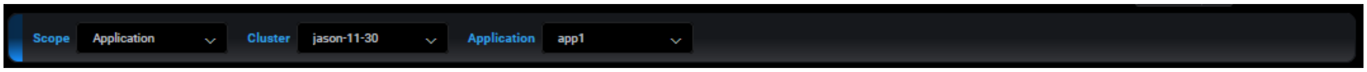
License Status: Expired
License Type: Standard
License Keys: 0 (Valid/Pending Activation/Grace Period)
1 (Expired)
License Capacity: 0 (Valid/Pending Activation/Grace Period)
10 (Expired)
Expiration Date: 2021-08-03 11:30:00
Message: All licenses are expired

User Functions

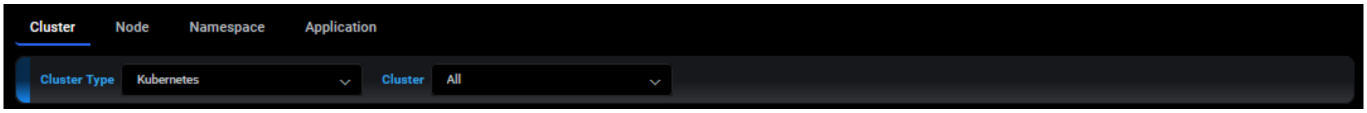
Click the *User* icon to contact technical support, view the Federator.ai product documentation, display the product software version, or log out from the system.

Filters

A panel appears on most pages that allows you to filter the display of information. Depending upon the page and cluster type, you may be able to select a scope/resource type, cluster, application, namespace, controller, time range, etc.



Scope: Application
Cluster: jason-11-30
Application: app1

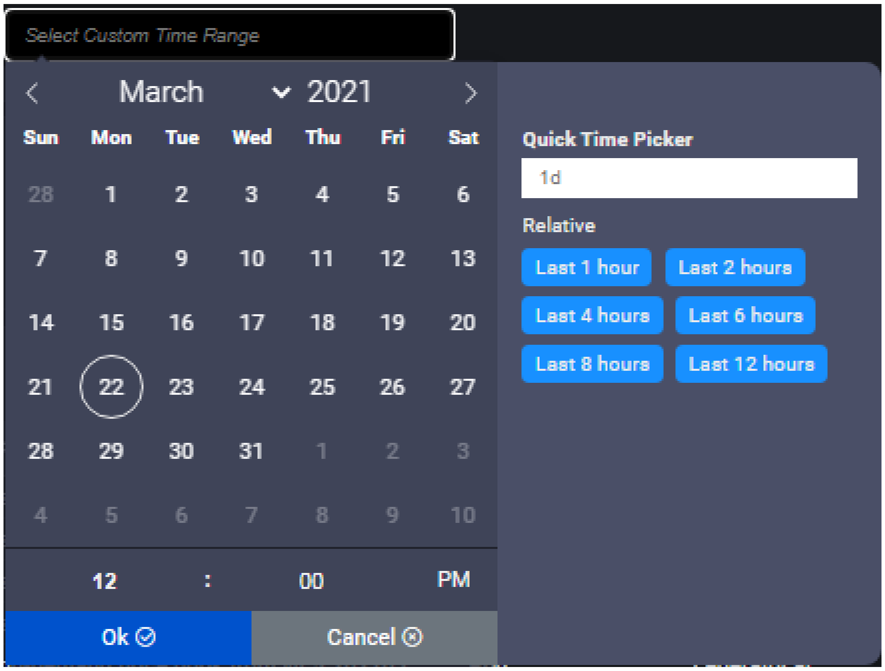


Cluster Type: Kubernetes
Cluster: All

Specify Time Range

When a chart allows you to specify a time range to display data, you can select a predefined time frame (e.g., last 1 hour, last 24 hours) from the drop-down box or you can specify a custom time range via one of the following methods:

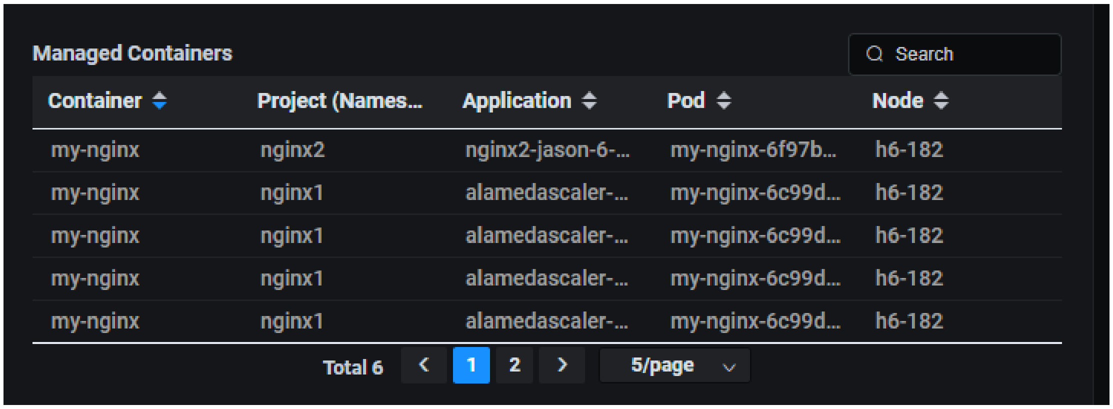
- Use the calendar to select the start and end dates.
- Specify the number of hours (e.g., 5h), days (e.g., 5d), weeks (e.g., 3w), or months (e.g., 6m) in the *Quick Time Picker* box.
- Click a predefined relative time range.



Quick Time Picker: 1d

Search/Sort Information in Tables

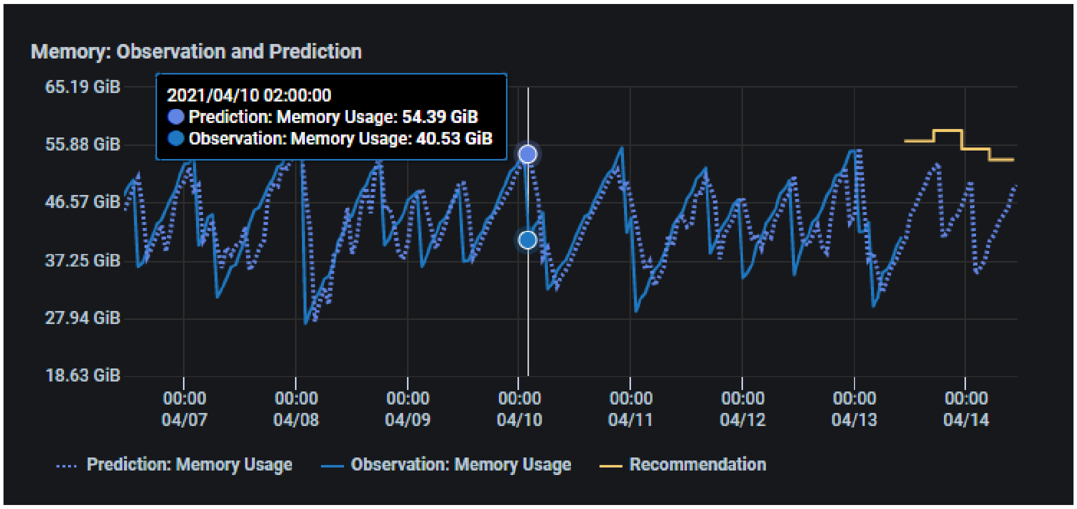
Click a column heading to sort the entire list based upon values in that column. The blue highlighted triangle or inverted triangle on the column indicates the direction of the sort. To search, type a name or value in the *Search* box; clear the search field to return to the full view of the list. As soon as you start typing, only those items that have matching text are displayed. You can also determine how many rows to show per page (5, 10, or 20).



Container	Project (Names...	Application	Pod	Node
my-nginx	nginx2	nginx2-jason-6-...	my-nginx-6f97b...	h6-182
my-nginx	nginx1	alamedascaler-...	my-nginx-6c99d...	h6-182
my-nginx	nginx1	alamedascaler-...	my-nginx-6c99d...	h6-182
my-nginx	nginx1	alamedascaler-...	my-nginx-6c99d...	h6-182
my-nginx	nginx1	alamedascaler-...	my-nginx-6c99d...	h6-182

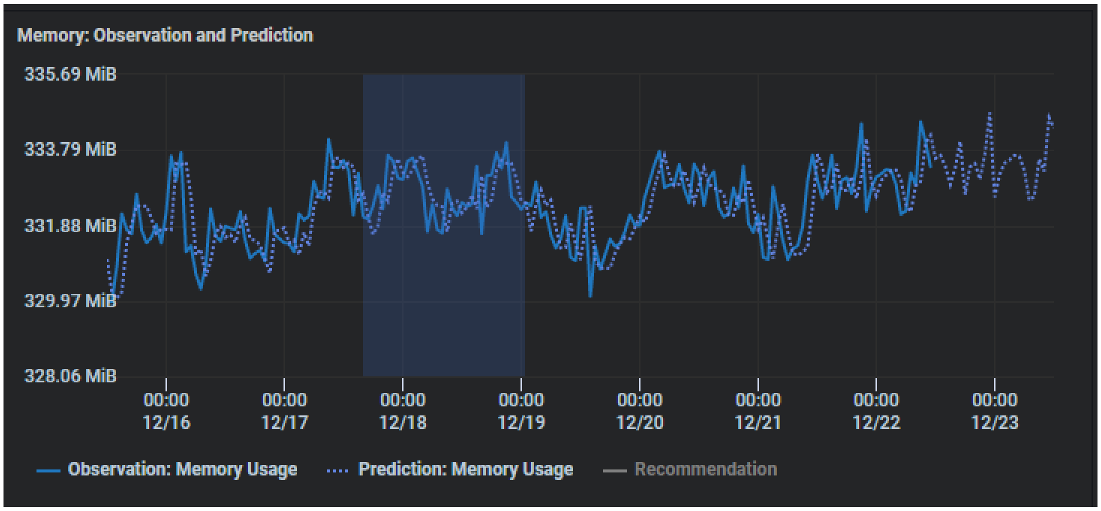
Show/Hide Metrics in Charts

Click anywhere on a chart to see values for a specific point in time. Highlight or click on the key at the bottom of the chart to show/hide individual metrics.

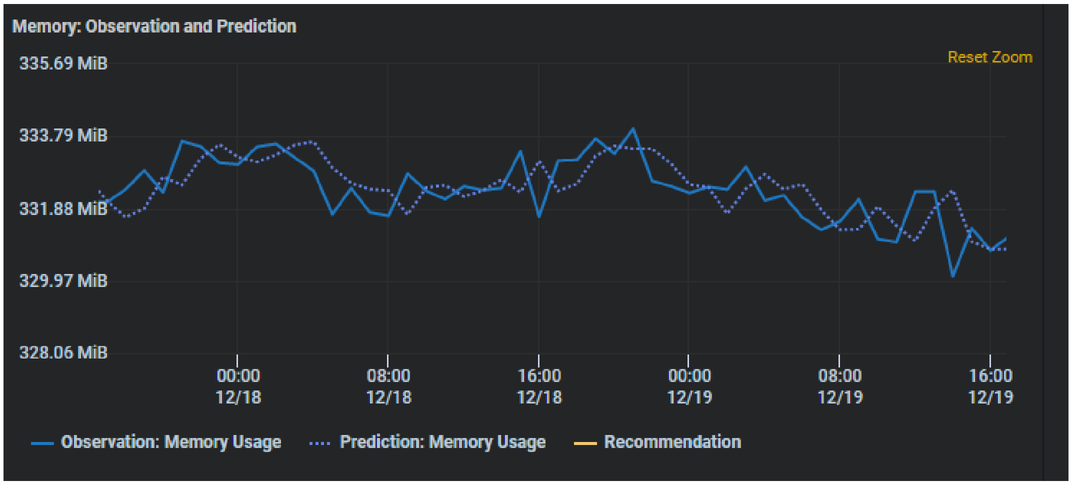


Zoom In/Out of Charts

Click and drag a pointer in a chart to zoom into a specific time frame of interest.



Click *Reset Zoom* to return to the original time frame.



Related topics:

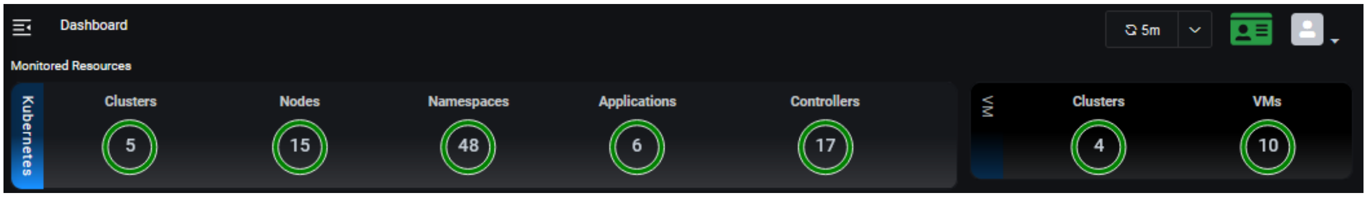
Common Administration Portal Functions

Dashboard

Licenses

Dashboard

The *Dashboard* displays the number of monitored resources in Kubernetes clusters and VM clusters, as well as cluster and Kubernetes application workload predictions and recommendations.



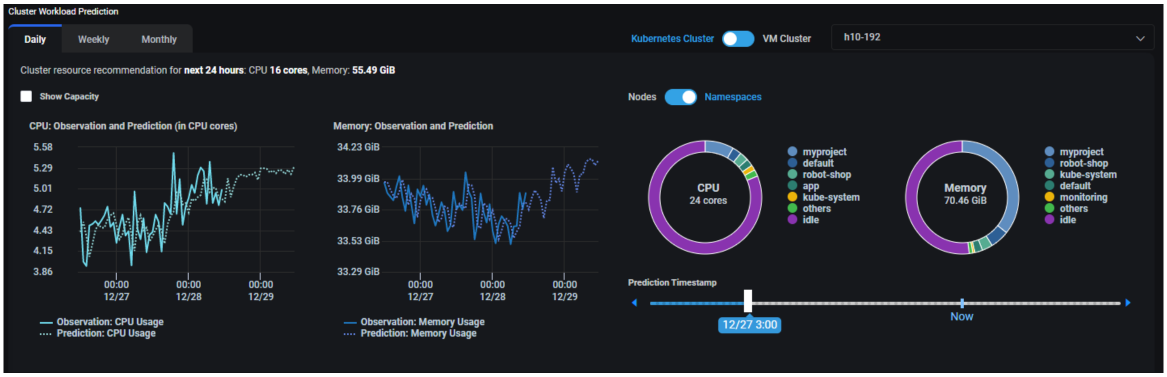
Monitored Resources

Kubernetes: Clusters 5, Nodes 15, Namespaces 48, Applications 6, Controllers 17

VM: Clusters 4, VMs 10

Cluster Workload Prediction

Toggle between *Kubernetes Cluster* and *VM Cluster,* select a cluster from the drop-down list, and select the time frame (daily, weekly, or monthly) to display CPU and memory observations, predictions, and recommendations.



Cluster Workload Prediction

Cluster resource recommendation for **next 24 hours**: CPU **16 cores**, Memory: **55.49 GiB**

□ Show Capacity

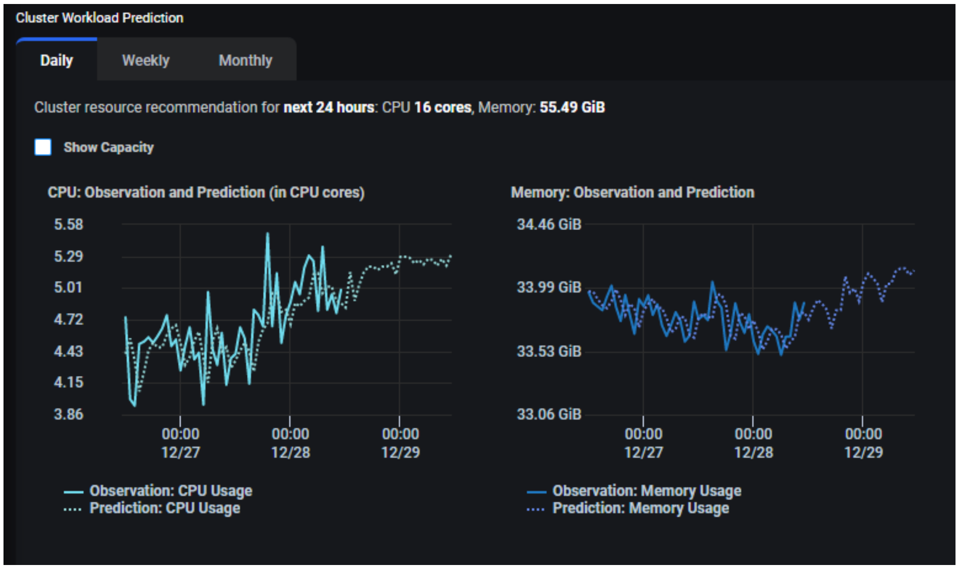
The first two charts display information for the selected cluster; the chart on the right toggles the display of information for nodes and namespaces of the selected Kubernetes cluster. For a VM cluster, the chart on the right displays information for the VMs in the selected cluster.

The text above the charts summarizes the CPU and memory recommendations for the next 24 hours (daily), 7 days (weekly), or 30 days (monthly).

Cluster Charts

The cluster charts display CPU and memory observations and predictions for the cluster.

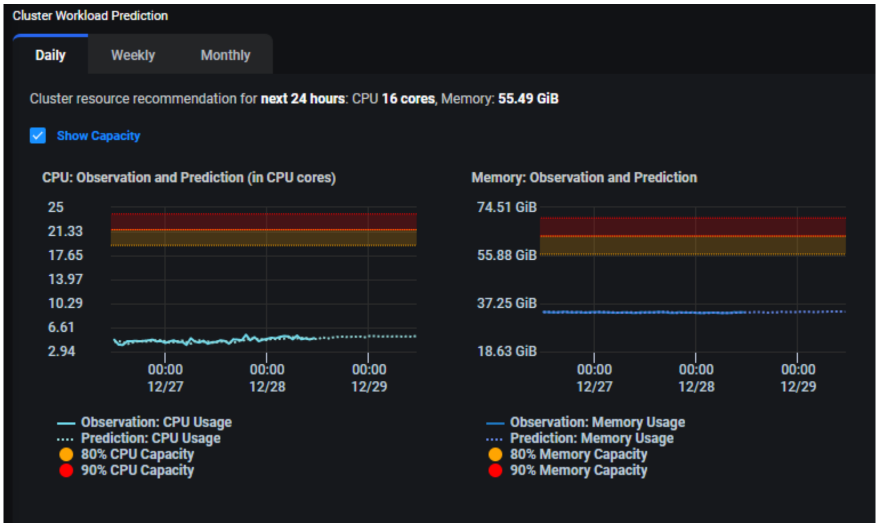
The solid lines represent the observed actual usage while the dotted lines show the historical and future predicted usage. Click anywhere on the charts to see values for a specific point in time. This will adjust the slider in the Nodes/Namespaces chart accordingly.



Cluster resource recommendation for **next 24 hours**: CPU **16 cores**, Memory: **55.49 GiB**

□ Show Capacity

Check *Show Capacity* to see the maximum CPU and memory usage limits for the cluster. Orange represents 80-90% and red represents 90-100%. This is a useful way to see if the utilization of resources is approaching the overall cluster capacity.



Cluster resource recommendation for **next 24 hours**: CPU **16 cores**, Memory: **55.49 GiB**

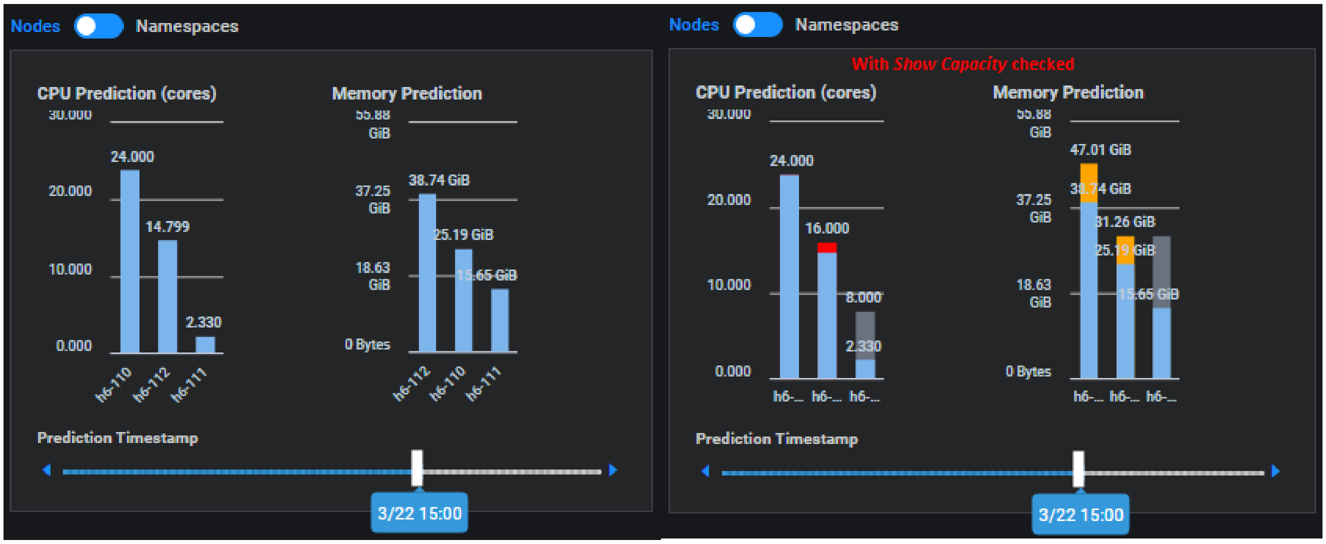
☑ Show Capacity

Node/Virtual Machine Chart

This chart displays CPU and memory observations and predictions for each member of the cluster. For Kubernetes, toggle to *Nodes* to display this chart.

The slider always starts at the current time but allows you to select any day/hour. Slide to the left of *Now* for historical usage; slide right for future predictions.

If *Show Capacity* was selected for the cluster, the chart will show if the node's utilization of resources is approaching the maximum CPU and memory usage limits. Orange represents less than 20% availability and red represents less than 10% availability.

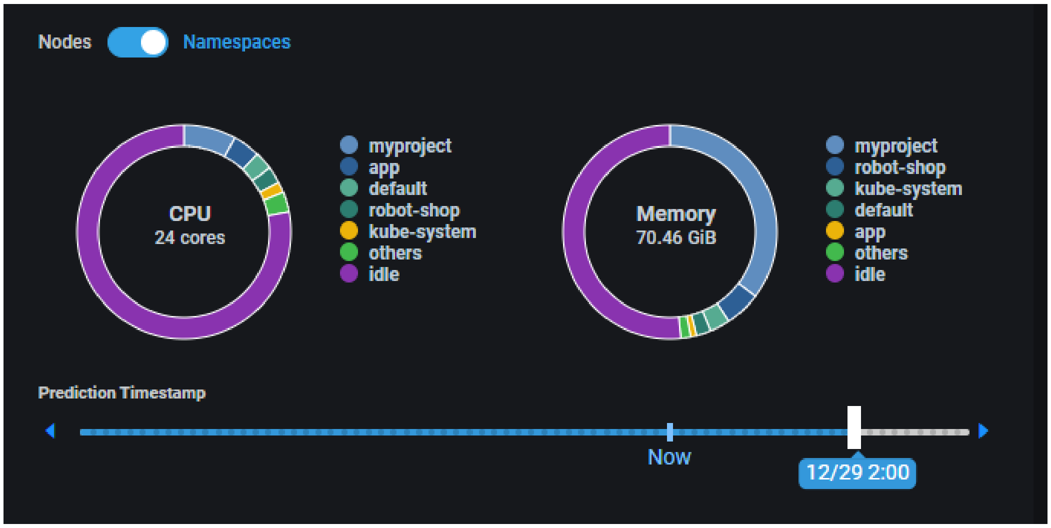

With *Show Capacity* checked

Namespace Chart (Kubernetes)

Toggle to *Namespaces* to display CPU and memory predictions for each namespace and for when the system is idle. Move your cursor over each section to show the usage or predicted usage by a specific namespace and the overall percentage used or predicted to use by the namespace or when the resource is idle.

Highlighting the amount of predicted idle (unused) resources provides a useful way to determine where your cluster is over-provisioned and can help you balance the resource allocation within the cluster.

The slider always starts at the current time but allows you to select any day/hour. Slide to the left of *Now* for historical predictions; slide right for future predictions.


Nodes
Namespaces
CPU
24 cores
myproject
app
default
robot-shop
kube-system
others
idle
Memory
70.46 GiB
myproject
robot-shop
kube-system
default
app
others
idle
Prediction Timestamp
Now
12/29 2:00

Application Workload Prediction (Kubernetes)

Select an application from the drop-down list and select the time frame (daily, weekly, or monthly) to display CPU and memory observations and predictions.

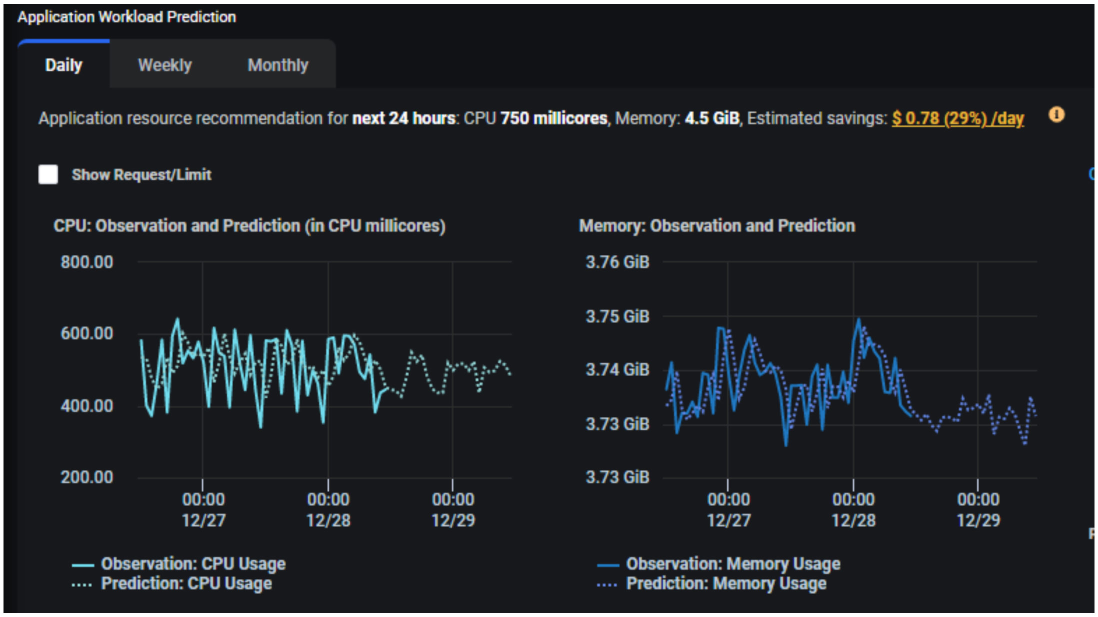
The first two charts display information for the selected application; the chart on the right displays information for controllers of the application.

The text above the charts summarizes the CPU and memory recommendations for the next 24 hours (daily), 7 days (weekly), or 30 days (monthly). It may also show estimated savings based on the system recommendations. You can link to the *Cost Optimization/Application* page for more details.

Application Charts

The application charts display CPU and memory observations, predictions, and recommendations for the application.

The solid lines represent the observed actual usage while the dotted lines show the historical and future predicted usage. Click anywhere on the charts to see values for a specific point in time. This will adjust the slider in the Controllers chart accordingly.

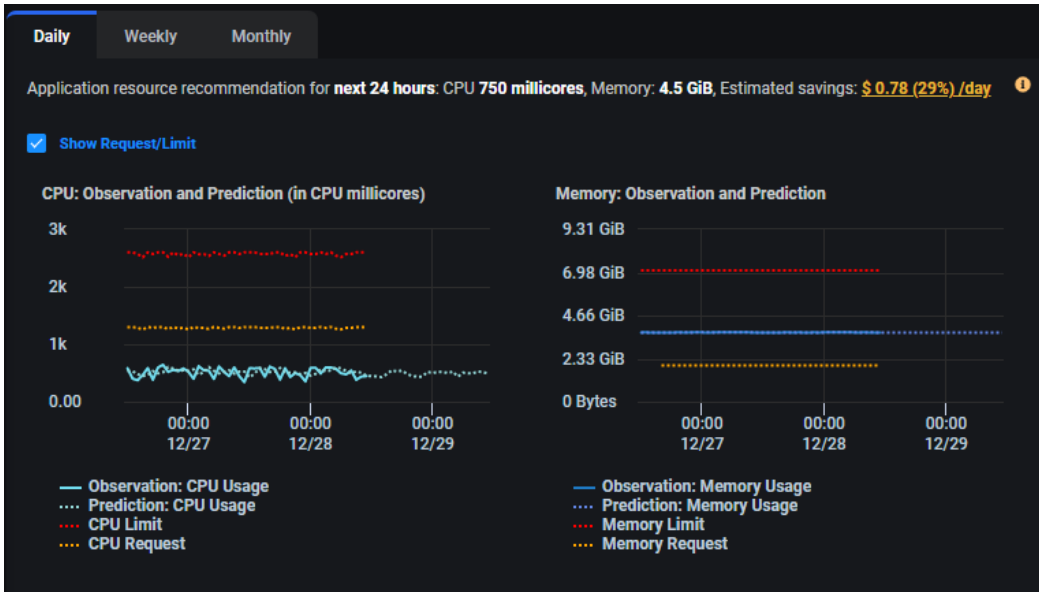


Application resource recommendation for **next 24 hours**: CPU **750 millicores**, Memory: **4.5 GiB**, Estimated savings: $ 0.78 (29%) /day

□ Show Request/Limit

Select *Show Request/Limit* to see your application's CPU and memory usage limits in Kubernetes. Orange represents your resource request and red represents the resource limit.

Generally, memory is a hard limit, but CPU is more *stretchable*. This is a useful way to see if you are over-provisioned or under-provisioned for your application.



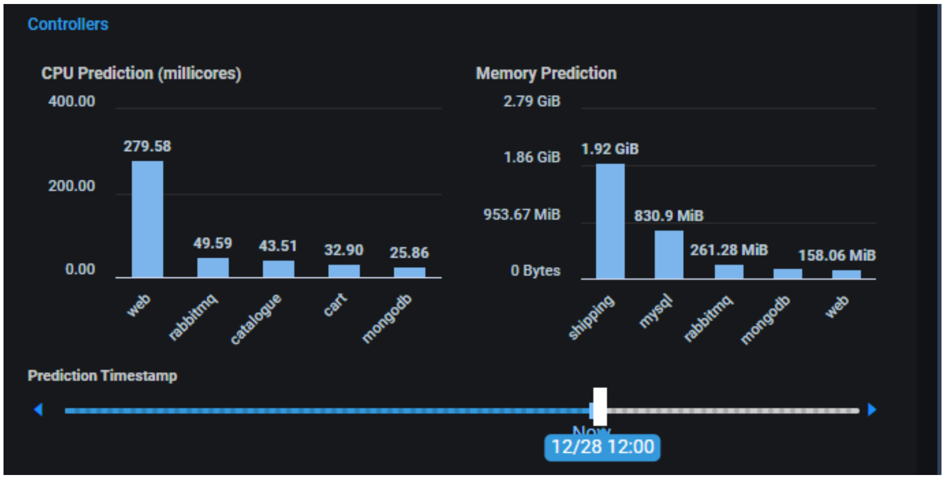
Application resource recommendation for **next 24 hours**: CPU **750 millicores**, Memory: **4.5 GiB**, Estimated savings: $ 0.78 (29%) /day

☑ Show Request/Limit

Controllers Chart

The chart displays CPU and memory observations and predictions for each controller defined in a generic application.

The slider always starts at the current time but allows you to select any day/hour. Slide to the left of *Now* for historical predictions; slide right for future predictions.



Related topics:

Common Administration Portal Functions

Refresh Statistics

License Status

User Functions

Search/Sort Information in Tables

Show/ Hide Information in Charts

Zoom In/Out of Charts

Filters

Application Analysis

Each Kubernetes application typically consists of multiple microservices. Also known as controllers, microservices are independent, modular components that work together as a single application. For example, a web shopping application might include microservices for a catalog, database, shopping cart, payment, shipping, etc.

The overall application workload can impact each of the microservices differently. Using our example, a highly advertised sale during the holiday season may drive a larger than usual number of customers to the shopping website, greatly impacting the shopping cart, payment, and shipping services as well as the catalog and database.

Understanding how individual microservices/controllers are impacted by external factors that affect the primary workload is very important in order to prevent over- or under-provisioning of resources. For example, how will a 50% increase in primary workload increase the CPU or memory usage for each microservice/controller?

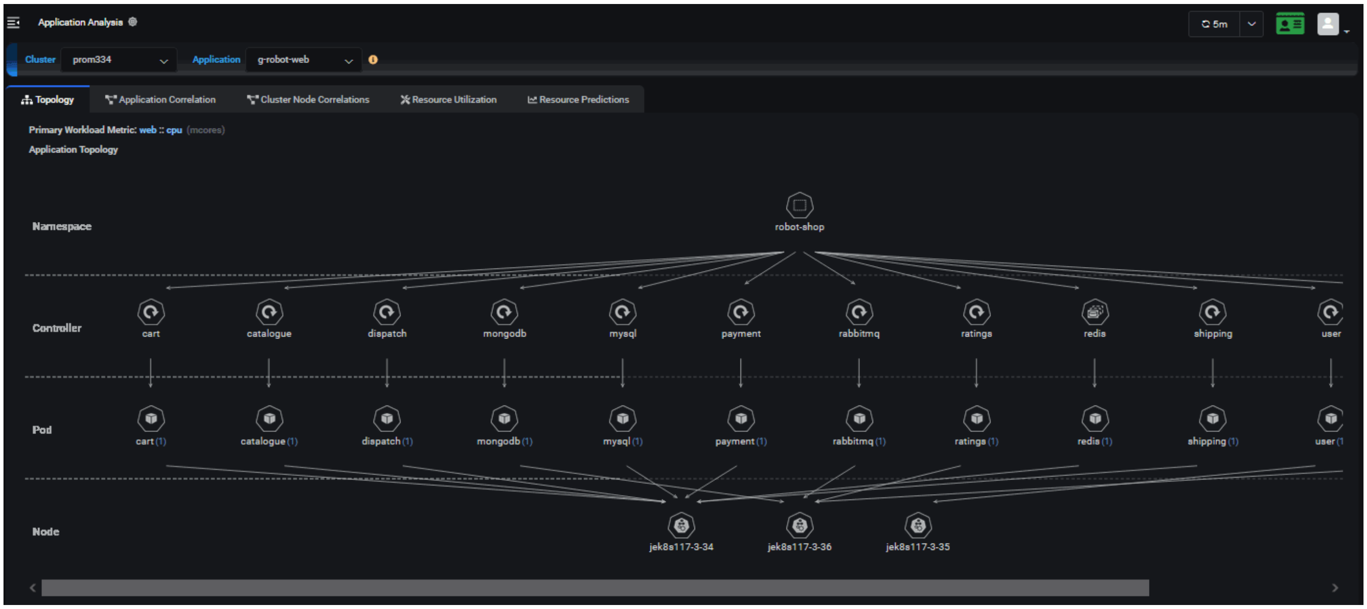
Federator.ai provides insight to individual controllers and cluster nodes to predict the resource usage of each microservice and make system recommendations based on application workload and the impact to each microservice.

The *Application Analysis* section includes pages for:

- Topology – Displays how microservices/controllers interact and the top five CPU and memory users.
- Application correlation – Displays the correlation between controllers of an application.
- Cluster node correlation – Displays the correlation between controllers and nodes on which an application runs of an application.
- Resource utilization – Displays the resource breakdown for each metric for controllers and nodes.
- Resource predictions – Displays resource predictions and recommendations for the primary workload for each controller and node.
- Application KPI Metrics – Displays metrics from each controller of the application, including common metrics, such as CPU, memory, and network traffic, as well as application-specific KPI metrics, if configured.

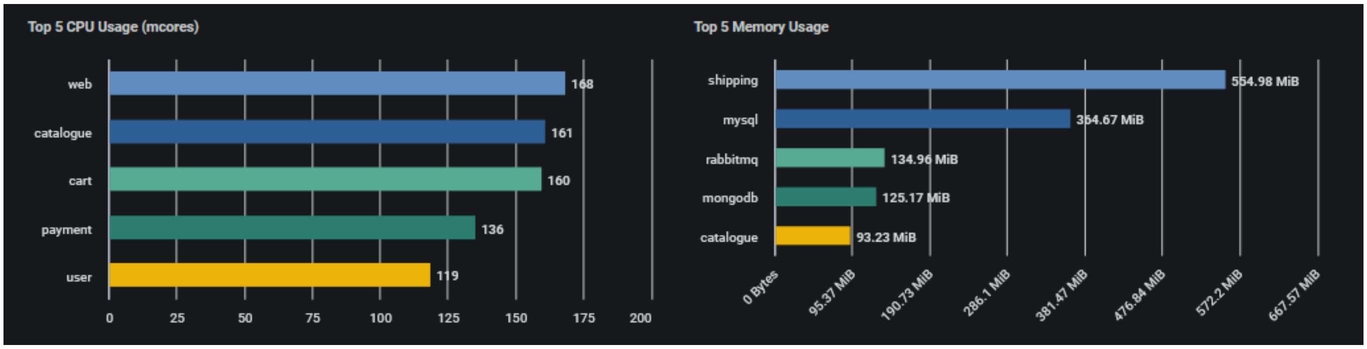
At the top of each page, you will see the controller considered the primary indicator of application load and the workload metric being used. The metric can be CPU, memory, number of network bytes received, number of network bytes transmitted, or an application-specific metric for a controller. Information for metrics from each controller is compared against the primary. The primary controller and the workload metric are set from the *Configuration* - *Applications* page.

Topology

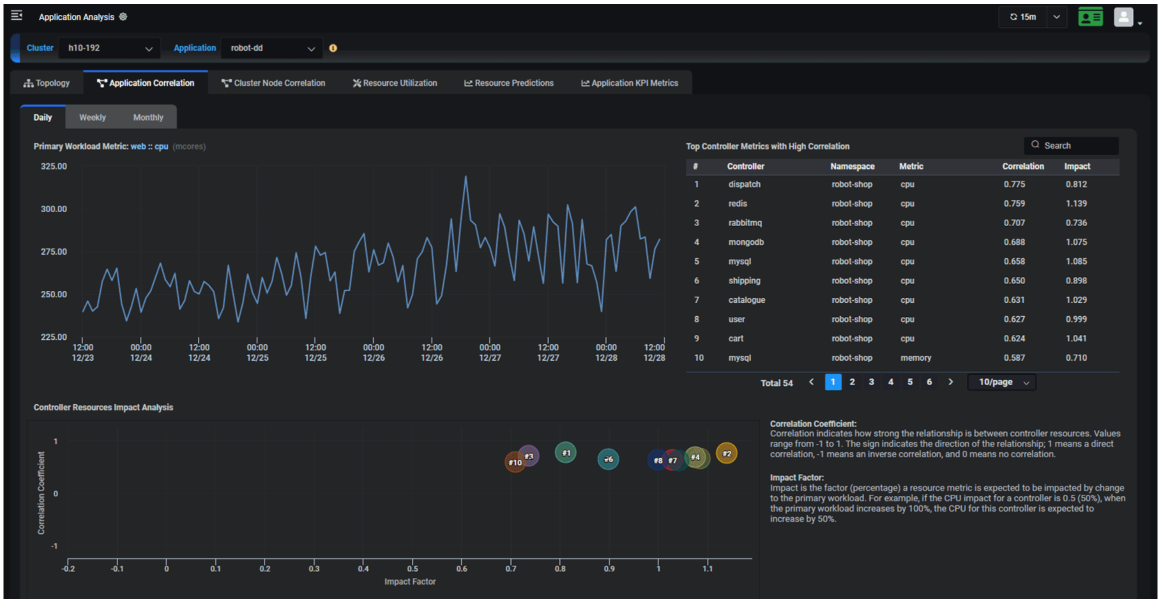


The *Topology* page shows the microservices/controllers of an application and which nodes they are running on. Here, you will see the namespace, controllers, pods, and nodes for a specific application. The number next to each pod tells you how many pods are used for that function.

The *Top 5 CPU/Memory Usage* charts display the current top five controllers for CPU and memory usage. Data is calculated every hour.



Application Correlation



Primary Workload Metric: web :: cpu (mcores)

Top Controller Metrics with High Correlation

#	Controller	Namespace	Metric	Correlation	Impact
1	dispatch	robot-shop	cpu	0.775	0.812
2	redis	robot-shop	cpu	0.759	1.139
3	rabbitmq	robot-shop	cpu	0.707	0.736
4	mongodb	robot-shop	cpu	0.688	1.075
5	mysql	robot-shop	cpu	0.658	1.085
6	shipping	robot-shop	cpu	0.650	0.898
7	catalogue	robot-shop	cpu	0.631	1.029
8	user	robot-shop	cpu	0.627	0.999
9	cart	robot-shop	cpu	0.624	1.041
10	mysql	robot-shop	memory	0.587	0.710

Total 54 < 1 2 3 4 5 6 > 10/page

Correlation Coefficient:
Correlation indicates how strong the relationship is between controller resources. Values range from -1 to 1. The sign indicates the direction of the relationship; 1 means a direct correlation, -1 means an inverse correlation, and 0 means no correlation.

Impact Factor:
Impact is the factor (percentage) a resource metric is expected to be impacted by change to the primary workload. For example, if the CPU impact for a controller is 0.5 (50%), when the primary workload increases by 100%, the CPU for this controller is expected to increase by 50%.

The *Application Correlation* page shows the resource usage correlation between controllers and the primary workload metric of an application. There can be a large number of controller/metrics combinations to analyze because there are at least four metrics (CPU, memory, number of network bytes received, number of network bytes transmitted), and *n* number of controllers. Therefore, there are 4 x *n* possible controller/metric combinations.

The top chart displays values of the primary workload metric during the specified time frame:

- Daily – Displays the last five days.
- Weekly – Displays the last 30 days.
- Monthly – Displays the last four months.

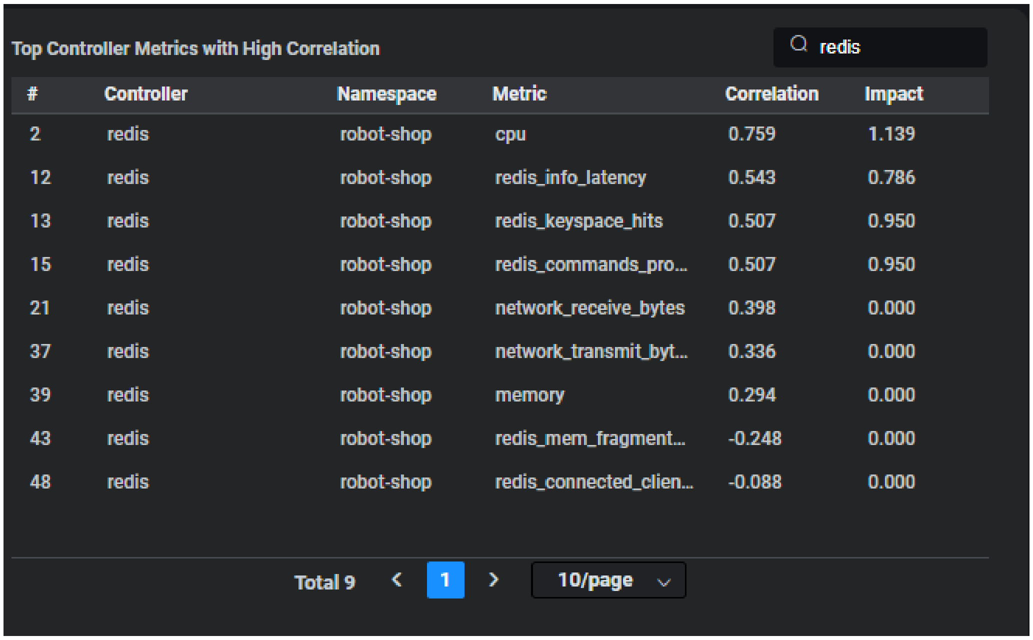
Click anywhere on the chart to see values for a specific point in time.

The *Top Controller Correlation* table displays the controller metrics in the order of the correlation to the primary workload metric. This chart identifies which resource metrics of controllers have the highest correlation to the primary workload and how much of an impact there is to these metrics when the primary workload metrics increase or decrease. For each controller in the table, you will see the associated namespace, metric, correlation, and impact. Correlation indicates how strong the relationship is between controller resource metrics or other performance metrics and the primary workload metric. It is a measure of whether resource usage or performance of a controller go up/down similarly to the primary workload metric. Values range from -1 to 1, where 0 means no correlation. The sign indicates the direction of the relationship:

- 1 = Direct correlation. If the primary controller/metric usage goes up, this controller/metric usage is predicted to go up.
- -1 = Inverse correlation. If the primary controller/metric usage goes up, this controller/metric usage is predicted to go down.

Impact is the factor (percentage) a resource metric is expected to be impacted by change to the primary workload. It measures how much the increase/decrease of the workload metric impacts the increase/decrease of resource usage or a performance metric for a controller. For example, if the CPU impact for a controller is 0.5 (50%), when the primary workload increases by 100%, the CPU for this controller is expected to increase by 50%. Note that the impact factor of a metric of a controller is only meaningful when there is a stronger correlation between the metric and the primary workload metric. The impact factor of a metric is set to 0 if the correlation coefficient between this metric and the primary workload metric is between 0.5 and -0.5.

For a quick view on the primary workload's impact to a specific controller (e.g., *redis*), type the name of the controller in the search box. You will see the correlation coefficient and impact factor of each controller metric versus the primary workload metric.



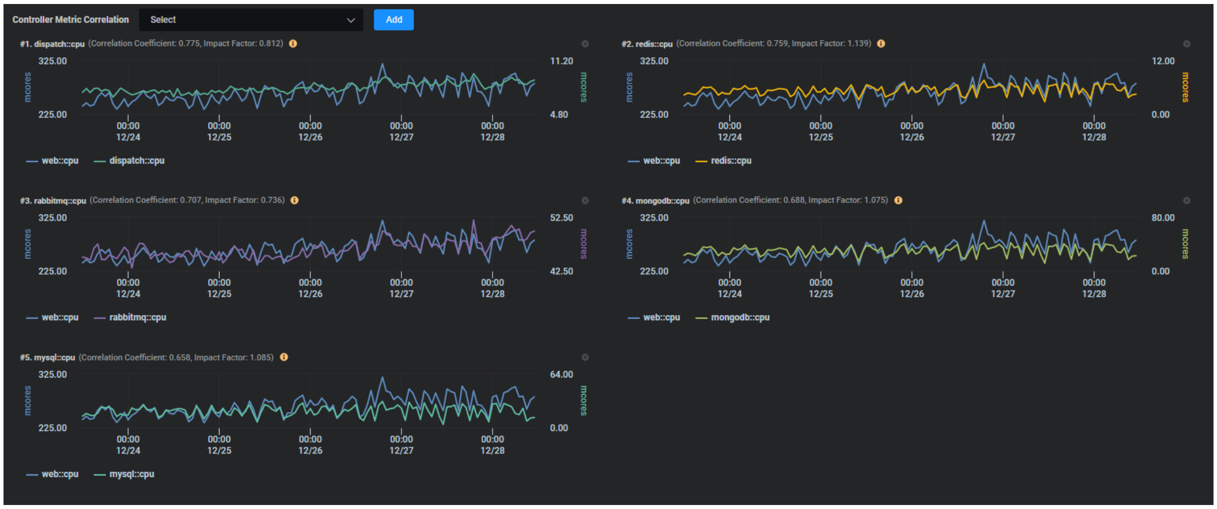
Top Controller Metrics with High Correlation

#	Controller	Namespace	Metric	Correlation	Impact
2	redis	robot-shop	cpu	0.759	1.139
12	redis	robot-shop	redis_info_latency	0.543	0.786
13	redis	robot-shop	redis_keyspace_hits	0.507	0.950
15	redis	robot-shop	redis_commands_pro...	0.507	0.950
21	redis	robot-shop	network_receive_bytes	0.398	0.000
37	redis	robot-shop	network_transmit_byt...	0.336	0.000
39	redis	robot-shop	memory	0.294	0.000
43	redis	robot-shop	redis_mem_fragment...	-0.248	0.000
48	redis	robot-shop	redis_connected_clien...	-0.088	0.000

The *Controller Resources Impact Analysis* chart is a visualization of the data that appears in the *Top Controller Metrics with High Correlation* table. This chart displays the metrics shown in the current page of *Top Controller Correlation Table*. If there are more than 10 controller metrics in the table, click a new page number to view the next 10 controller metrics in the chart.

The *Controller Metric Correlation* charts provide a visualization of the top five controllers metrics with the highest correlation to the primary workload metric. The blue line is the primary workload metric while a different color is used for the controller/metric being compared. While the distance between lines may not be very pronounced, the relationship between lines (one goes up, the other goes up/down) reveals how the resource usage or performance metric of a controller is tied together. You can show additional comparisons of controller metrics with the primary workload metric by selecting a different controller metric from the drop-down menu and adding it to the list.

The range for each metric is displayed on the sides of the chart and may be different for each controller metric.



Controller Metric Correlation Select Add

#1. dispatch::cpu (Correlation Coefficient: 0.775, Impact Factor: 0.812)

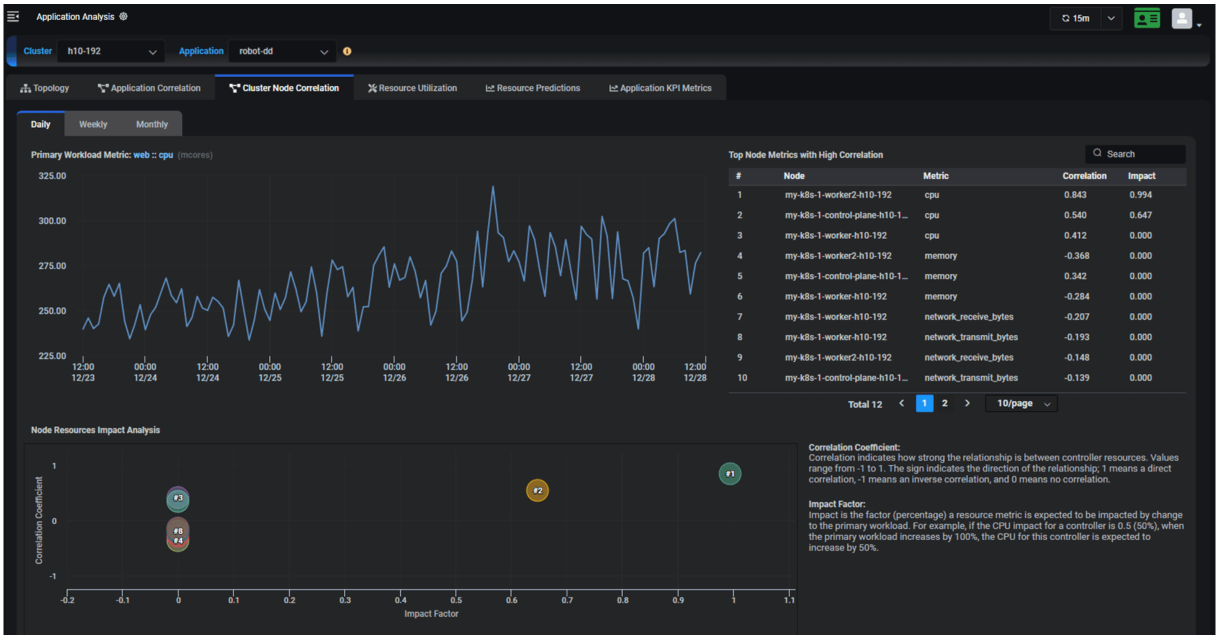
#2. redis::cpu (Correlation Coefficient: 0.759, Impact Factor: 1.139)

#3. rabbitmq::cpu (Correlation Coefficient: 0.707, Impact Factor: 0.736)

#4. mongodb::cpu (Correlation Coefficient: 0.688, Impact Factor: 1.075)

#5. mysql::cpu (Correlation Coefficient: 0.658, Impact Factor: 1.085)

Cluster Node Correlation



Primary Workload Metric: web :: cpu (mcores)

Top Node Metrics with High Correlation

#	Node	Metric	Correlation	Impact
1	my-k8s-1-worker2-h10-192	cpu	0.843	0.994
2	my-k8s-1-control-plane-h10-1...	cpu	0.540	0.647
3	my-k8s-1-worker-h10-192	cpu	0.412	0.000
4	my-k8s-1-worker2-h10-192	memory	-0.368	0.000
5	my-k8s-1-control-plane-h10-1...	memory	0.342	0.000
6	my-k8s-1-worker-h10-192	memory	-0.284	0.000
7	my-k8s-1-worker-h10-192	network_receive_bytes	-0.207	0.000
8	my-k8s-1-worker-h10-192	network_transmit_bytes	-0.193	0.000
9	my-k8s-1-worker2-h10-192	network_receive_bytes	-0.148	0.000
10	my-k8s-1-control-plane-h10-1...	network_transmit_bytes	-0.139	0.000

Total 12 < 1 2 > 10/page

Correlation Coefficient:
Correlation indicates how strong the relationship is between controller resources. Values range from -1 to 1. The sign indicates the direction of the relationship; 1 means a direct correlation, -1 means an inverse correlation, and 0 means no correlation.

Impact Factor:
Impact is the factor (percentage) a resource metric is expected to be impacted by change to the primary workload. For example, if the CPU impact for a controller is 0.5 (50%), when the primary workload increases by 100%, the CPU for this controller is expected to increase by 50%.

The *Cluster Node Correlation* page shows the resource usage correlation between cluster node resource usage and the primary workload metric of an application. There can be a large number of node/metrics combinations to analyze because there are at least four metrics (CPU, memory, number of network bytes received, number of network bytes transmitted), and *n* number of nodes. Therefore, there are 4 x *n* possible node/metric combinations.

The top chart displays actual usage for the primary controller during the specified time frame:

- Daily – Displays the last five days.
- Weekly – Displays the last 30 days.
- Monthly – Displays the last four months.

Click anywhere on the chart to see values for a specific point in time.

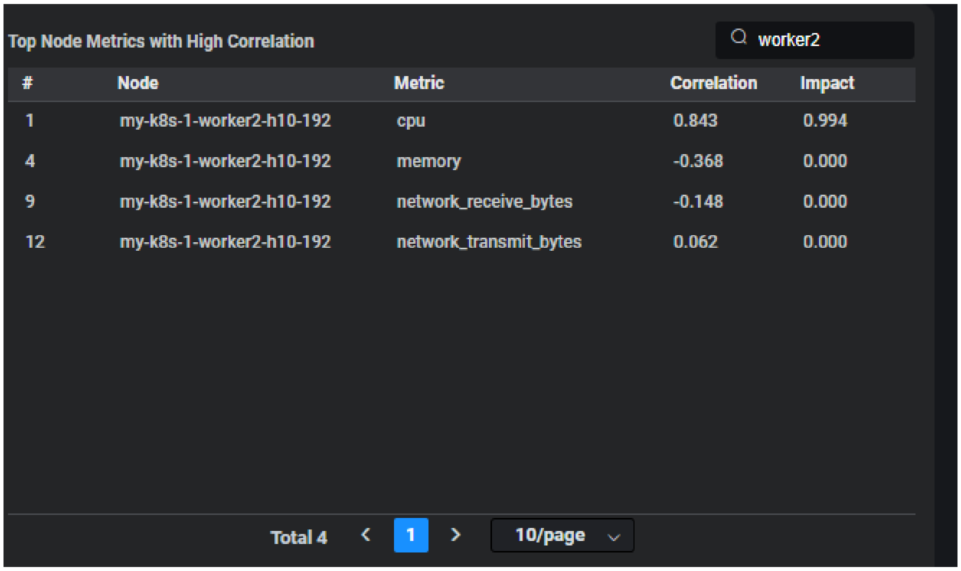
The *Top Node Correlation* table displays the top node resource usage metrics with the highest correlation to the primary workload metric. This chart identifies which node resources get impacted the most when primary workload metric is higher/lower. For each node in the table, you will see the associated metric, correlation, and impact.

Correlation indicates how strong the relationship is between node resources and the primary workload metric of an application. It is a measure of whether resource usage of a node goes up/down similarly to the primary workload metric. Values range from -1 to 1, where 0 means no correlation. The sign indicates the direction of the relationship:

- 1 = Direct correlation. If the primary workload metric usage goes up, this node’s resource usage metric is predicted to go up.
- -1 = Inverse correlation. If the primary workload metric usage goes up, this node’s resource usage metric is predicted to go down.

Impact is the factor (percentage) a resource metric is expected to be impacted by change to the primary workload metric. It measures how much the increase/decrease of the workload metric impacts the increase/decrease of resource usage for a node. For example, if the CPU impact for a node is 1.05 (105%), when the primary workload increases by 100%, the CPU for this node is expected to increase by 105%. Note that the impact factor of a resource metric of a node is only meaningful when there is a stronger correlation between the metric and the primary workload metric. The impact factor of a metric is set to 0 if the correlation coefficient between this metric and the primary workload metric is between 0.5 and -0.5.

For a quick view on the primary workload's impact to a specific cluster node, type the name of the cluster node in the search box. You will see the correlation coefficient and impact factor of each cluster node resource metric versus the primary workload metric.



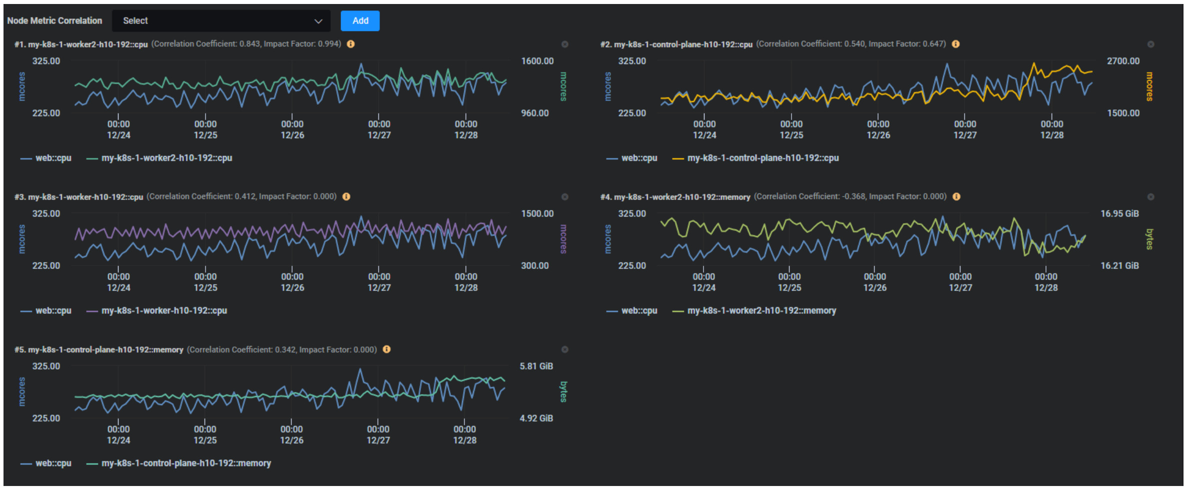
Top Node Metrics with High Correlation

#	Node	Metric	Correlation	Impact
1	my-k8s-1-worker2-h10-192	cpu	0.843	0.994
4	my-k8s-1-worker2-h10-192	memory	-0.368	0.000
9	my-k8s-1-worker2-h10-192	network_receive_bytes	-0.148	0.000
12	my-k8s-1-worker2-h10-192	network_transmit_bytes	0.062	0.000

The *Node Resources Impact Analysis* chart is a visualization of the data that appears in the *Top Node Correlation* table.

The *Node Metric Correlation* charts provide a visualization of the top five node metrics with the highest correlation to the primary workload metric of an application. The blue line is the primary workload metric while a different color is used for the node/metric being compared. While the distance between lines may not be very pronounced, the relationship between lines (one goes up, the other goes up/down) reveals how the node resource usage is tied together.

The range for each metric is displayed on the sides of the chart and may be different for each node metric.



Node Metric Correlation

#1. my-k8s-1-worker2-h10-192::cpu (Correlation Coefficient: 0.843, Impact Factor: 0.994)

#2. my-k8s-1-control-plane-h10-192::cpu (Correlation Coefficient: 0.540, Impact Factor: 0.647)

#3. my-k8s-1-worker-h10-192::cpu (Correlation Coefficient: 0.412, Impact Factor: 0.000)

#4. my-k8s-1-worker2-h10-192::memory (Correlation Coefficient: -0.368, Impact Factor: 0.000)

#5. my-k8s-1-control-plane-h10-192::memory (Correlation Coefficient: 0.342, Impact Factor: 0.000)

Resource Utilization

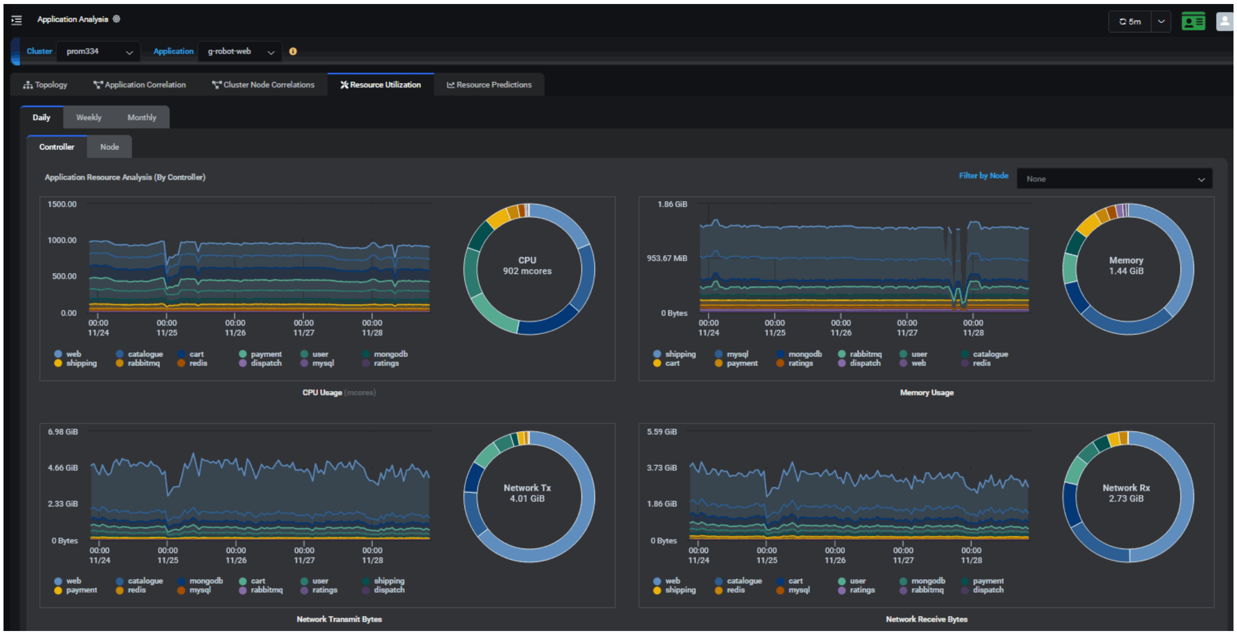
The *Resource Utilization* page displays the application resource breakdown for each metric (CPU, memory, number of network bytes received, number of network bytes transmitted) by each controller or by each cluster node for the specified time frame:

- Daily – Displays the last five days.
- Weekly – Displays the last 30 days.
- Monthly – Displays the last four months.

Click anywhere on a chart to see values for a specific point in time.

Resource Utilization - Controllers

These tables show the breakdown of resource usage by each controller. Data can be filtered by cluster node to show the breakdown of resource usage by controllers running on that node.



Cluster: prom334
Application: g-robot-web
Filter by Node: None

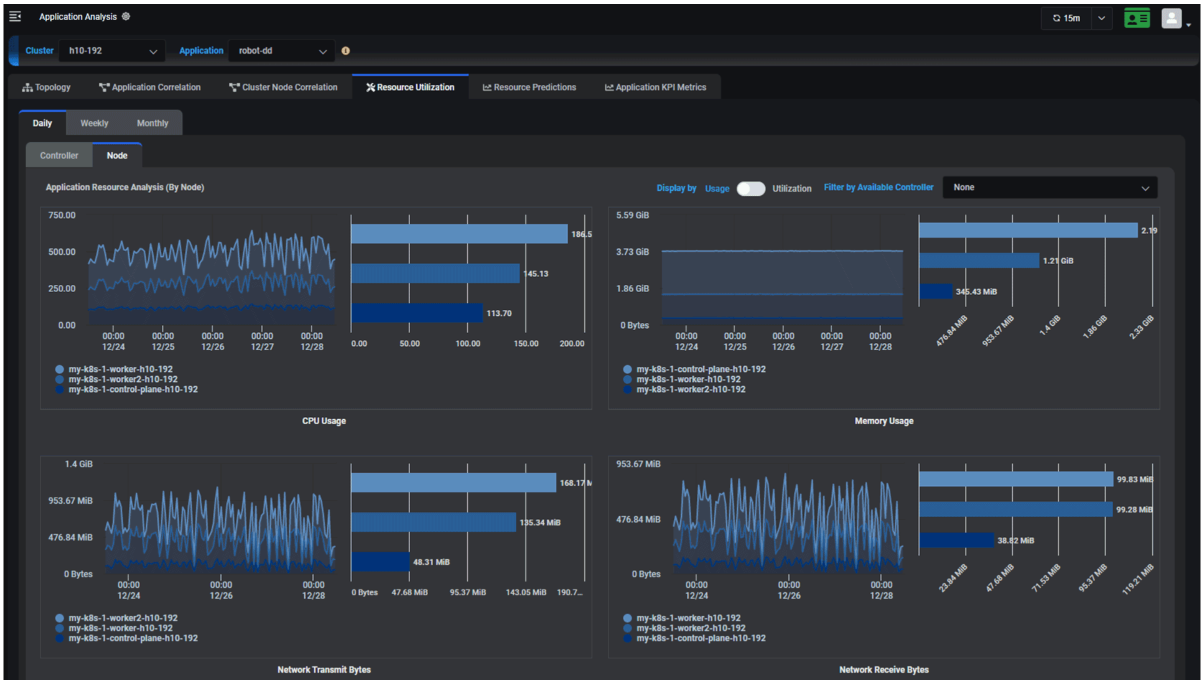
For each metric, there are two charts. The chart on the left displays the usage per controller over time. Click anywhere to see the values for all controllers at that time. You can click on the key at the bottom of the chart to show/hide individual controllers.

The donut chart on the right displays the total amount used in the last hour, six hours, or 24 hours in Daily/Weekly/Monthly view, respectively. Move your cursor over each section to show the usage (value and percentage) by a specific controller.

Resource Utilization - Nodes

These tables show the breakdown of resource usage by each node where an application runs. Data can be filtered by controller to show the resource usage of the controller on each node. You can also toggle to display data by actual usage or utilization (compared to capacity).

Usage



Cluster: h10-192

Application: robot-dd

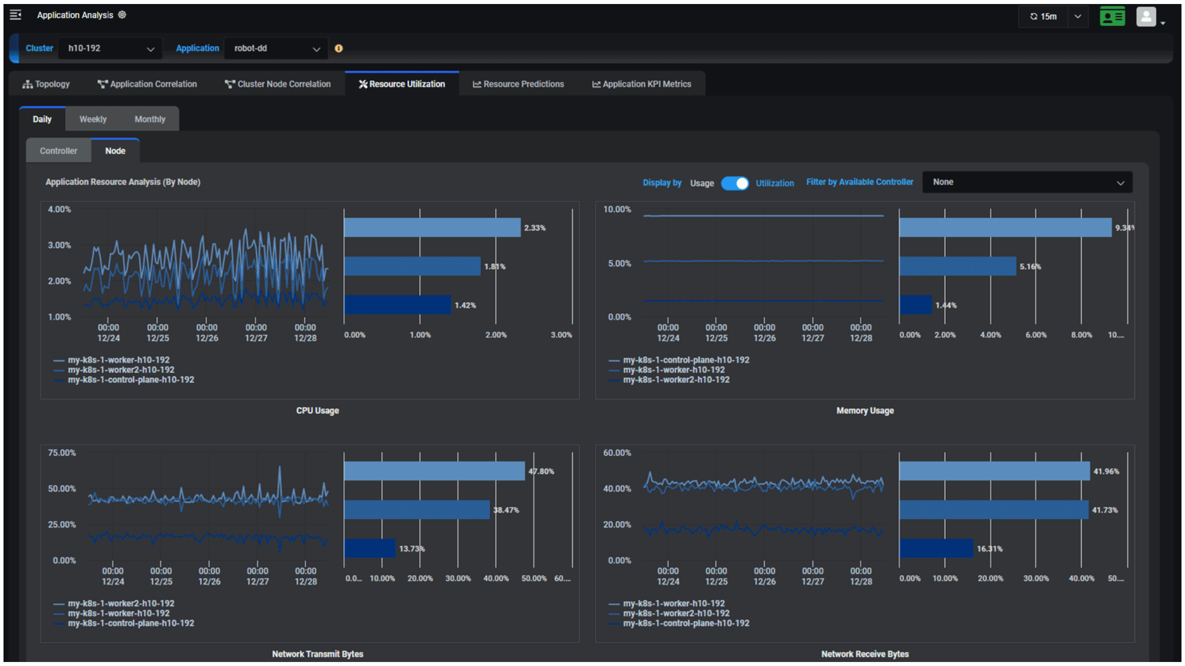
Display by Usage / Utilization

Filter by Available Controller: None

For each metric, there are two charts. The chart on the left displays the resource usage per node over time. Click anywhere to see the values for all nodes at that time. You can click on the key at the bottom of the chart to show/hide individual nodes.

The chart on the right displays the total amount used by the application on each node in the last hour, six hours, or 24 hours in Daily/Weekly/Monthly view, respectively.

Utilization



Cluster h10-192

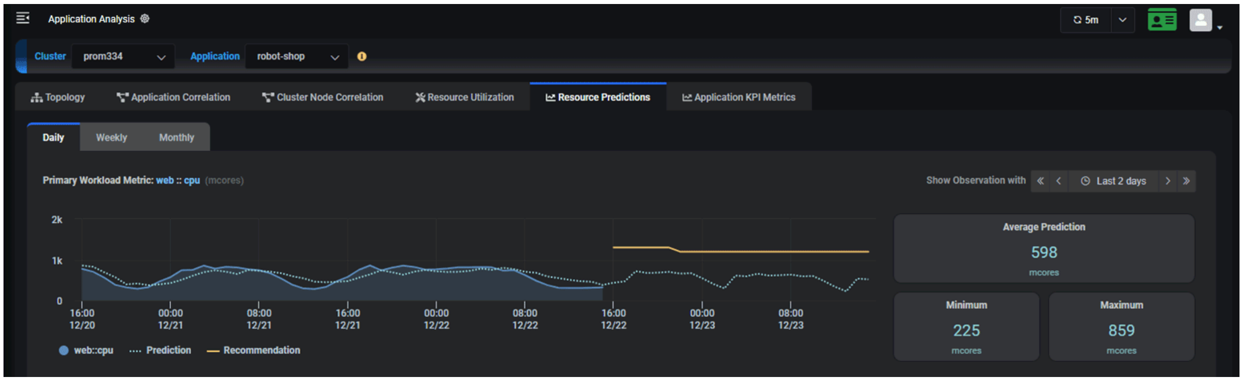
Application robot-dd

Filter by Available Controller None

For each metric, there are two charts. The chart on the left displays the utilization per node by the application over time. Click anywhere to see the values for all nodes at that time. You can click on the key at the bottom of the chart to show/hide individual nodes.

The chart on the right displays the percentage used by the application on each node in the last hour, six hours, or 24 hours in Daily/Weekly/Monthly view, respectively. For example, if the *Network Received Bytes* metric shows 55%, then 55% of traffic for this application went through this node.

Resource Predictions



The *Resource Predictions* page displays resource predictions and recommendations, along with average, minimum, and maximums for the primary workload metric for the specified time frame:

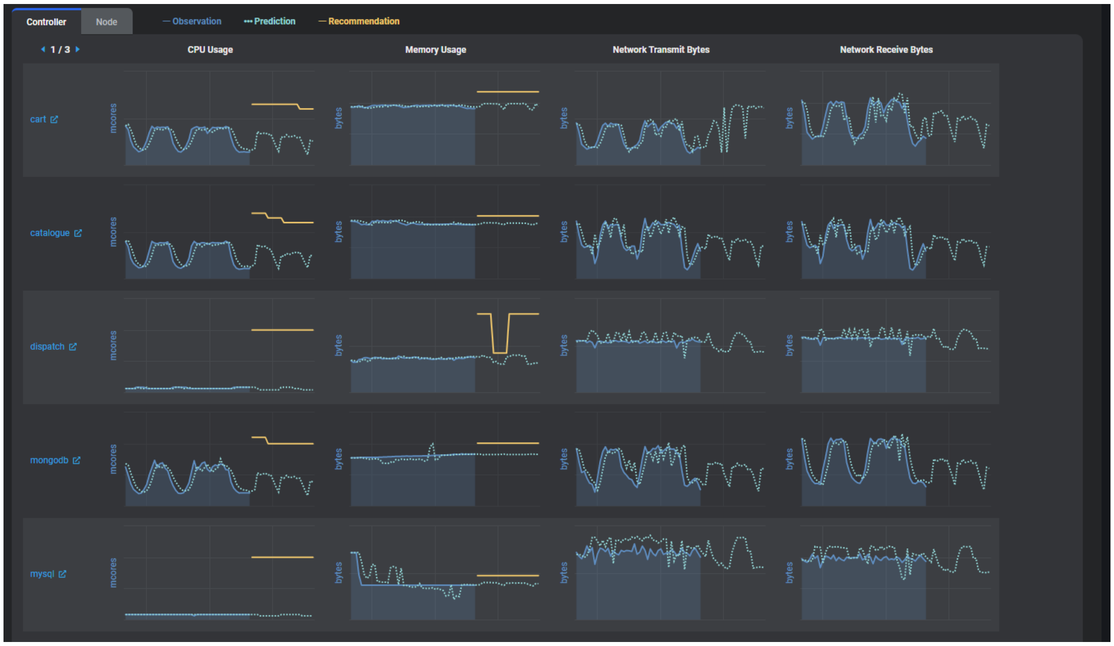
- Daily – Displays data and predictions for seven days but can be changed to display one day, two days, seven days, 14 days, 28 days, or a custom time frame.
- Weekly – Displays data and predictions for four weeks but can be changed to display one week, two weeks, four weeks, eight weeks, 12 weeks, or a custom time frame.
- Monthly – Displays data and predictions for three months but can be changed to display one month, two months, three months, six months, nine months, or a custom time frame.

Click anywhere on the chart to see values for a specific point in time.

Resource Predictions - Controllers

You can display a birds-eye view of usage, predictions, and recommendations for each metric (CPU, memory, number of network bytes received, number of network bytes transmitted) per controller for the specified time frame.

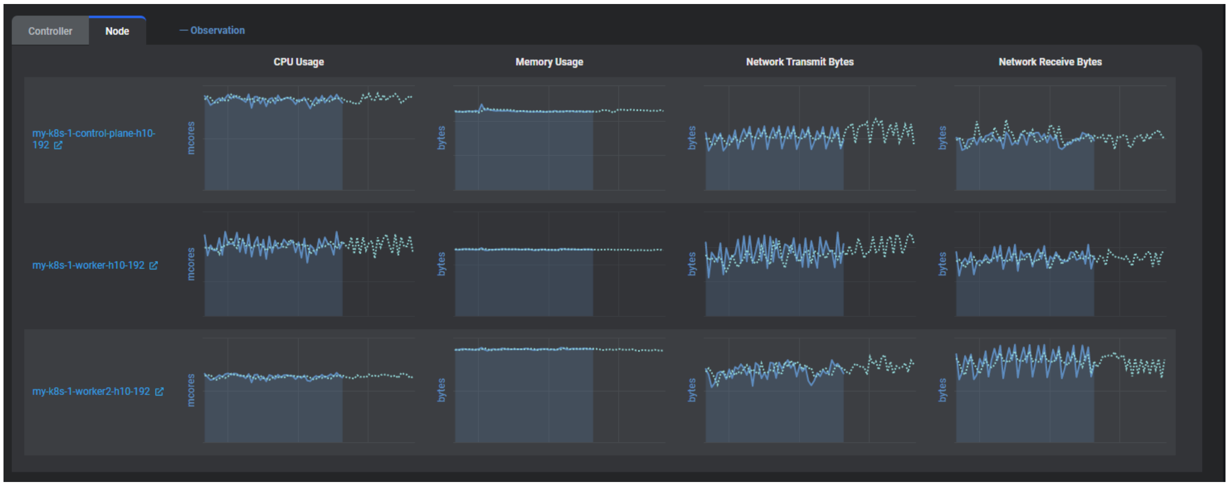
Click anywhere on the chart to see values for a specific point in time.



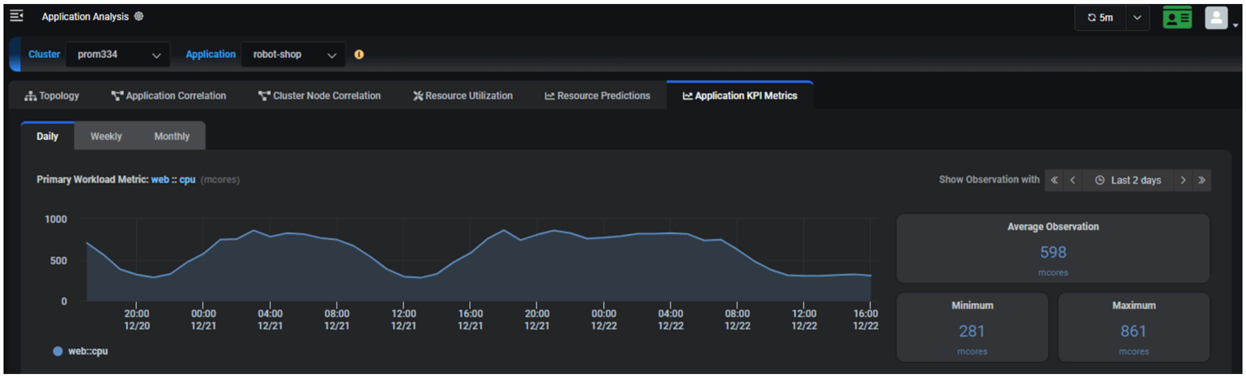
Resource Predictions - Nodes

You can display a birds-eye view of usage, predictions, and recommendations for each metric (CPU, memory, number of network bytes received, number of network bytes transmitted) per node for the specified time frame.

Click anywhere on the chart to see values for a specific point in time.



Application KPI Metrics



Cluster prom334

Application robot-shop

Primary Workload Metric: web :: cpu (mcores)

Average Observation 598 mcores

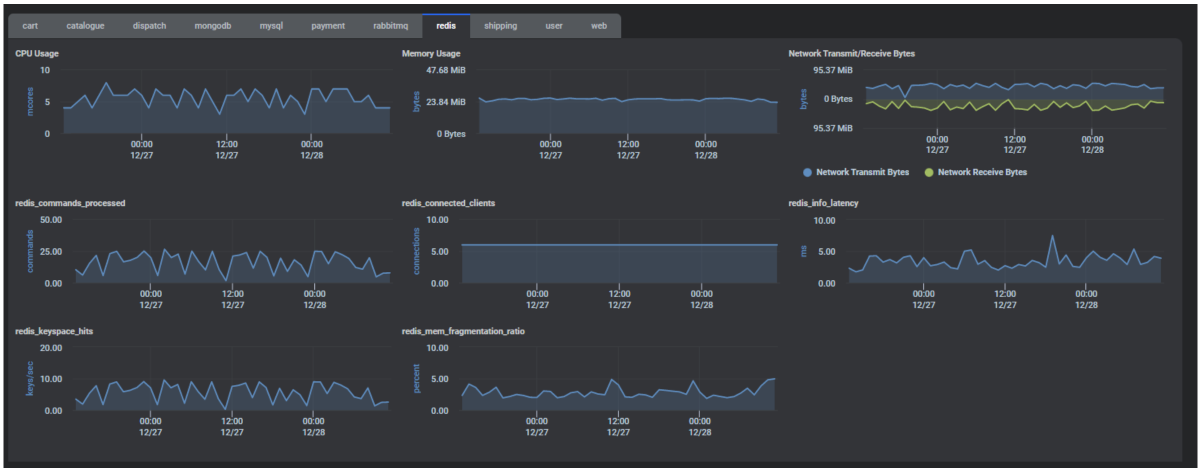
Minimum 281 mcores

Maximum 861 mcores

The *Application Key Performance Indicator (KPI) Metrics* page displays common resource metrics from each controller of an application, as well as application-specific metrics, if configured. It also displays the average, minimum, and maximum for the primary workload metric for the specified time frame:

- Daily – Displays data and predictions for seven days but can be changed to display one day, two days, seven days, 14 days, 28 days, or a custom time frame.
- Weekly – Displays data and predictions for four weeks but can be changed to display one week, two weeks, four weeks, eight weeks, 12 weeks, or a custom time frame.
- Monthly – Displays data and predictions for three months but can be changed to display one month, two months, three months, six months, nine months, or a custom time frame.

Click anywhere on the chart to see values for a specific point in time.

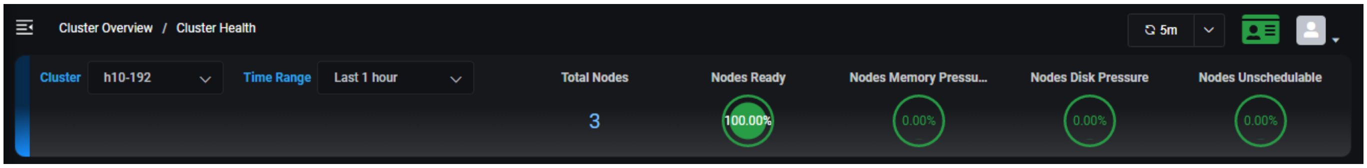


The bottom chart displays metrics for individual controllers. Click the name of the controller to see all KPI metrics for that controller.

Cluster Overview - Cluster Health

The *Cluster Health* page displays actual usage observations about the nodes/VMs in a cluster. Usage for the last 1, 2, 4, 6, or 12 hours can be displayed and can further be filtered by selecting a range of dates.

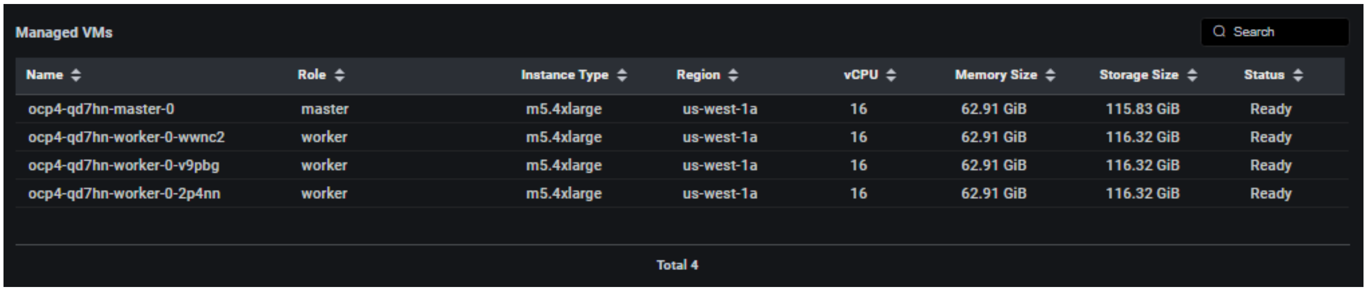
For the selected cluster, you will see the total number of nodes/VMs and the percentage that are ready. For Kubernetes, you will also see the percentage of nodes that are under memory or disk pressure, as well as the percentage of nodes that are not schedulable. Memory pressure and disk pressure are defined by Kubernetes. Refer to https://kubernetes.io/docs/tasks/administer-cluster/out-of-resource/#node-conditions for more information.



Cluster: h10-192
Time Range: Last 1 hour

Total Nodes	Nodes Ready	Nodes Memory Pressu...	Nodes Disk Pressure	Nodes Unschedulable
3	100.00%	0.00%	0.00%	0.00%

The table below displays the current configuration for each cluster node, including Kubernetes role (master, worker), instance types being used at your cloud provider or operating system (OS) for your local cluster, cloud provider region, number of CPUs, memory size, storage size, and node status.

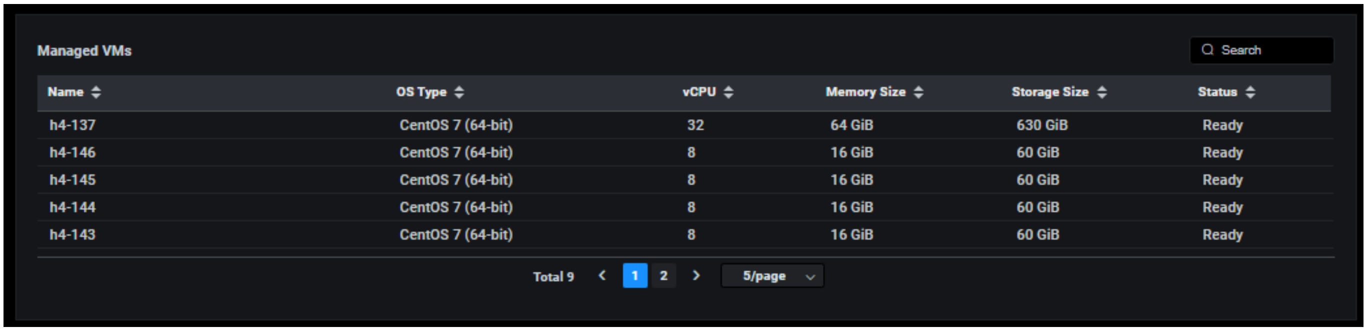


Managed VMs

Name	Role	Instance Type	Region	vCPU	Memory Size	Storage Size	Status
ocp4-qd7hn-master-0	master	m5.4xlarge	us-west-1a	16	62.91 GiB	115.83 GiB	Ready
ocp4-qd7hn-worker-0-wwnc2	worker	m5.4xlarge	us-west-1a	16	62.91 GiB	116.32 GiB	Ready
ocp4-qd7hn-worker-0-v9pbg	worker	m5.4xlarge	us-west-1a	16	62.91 GiB	116.32 GiB	Ready
ocp4-qd7hn-worker-0-2p4nn	worker	m5.4xlarge	us-west-1a	16	62.91 GiB	116.32 GiB	Ready

Total 4

Kubernetes cluster



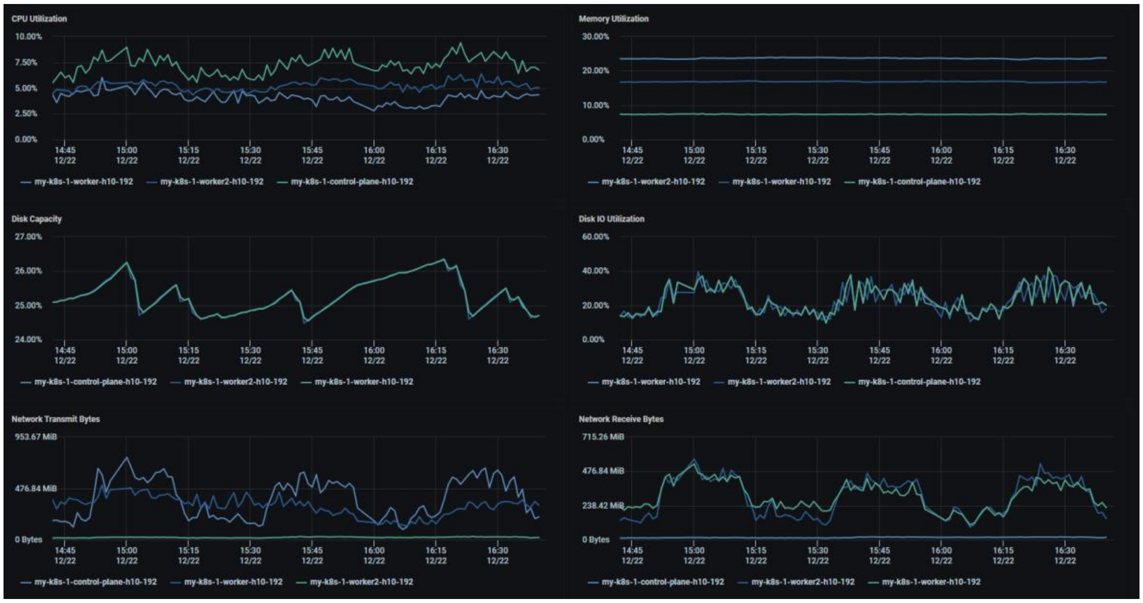
Managed VMs

Name	OS Type	vCPU	Memory Size	Storage Size	Status
h4-137	CentOS 7 (64-bit)	32	64 GiB	630 GiB	Ready
h4-146	CentOS 7 (64-bit)	8	16 GiB	60 GiB	Ready
h4-145	CentOS 7 (64-bit)	8	16 GiB	60 GiB	Ready
h4-144	CentOS 7 (64-bit)	8	16 GiB	60 GiB	Ready
h4-143	CentOS 7 (64-bit)	8	16 GiB	60 GiB	Ready

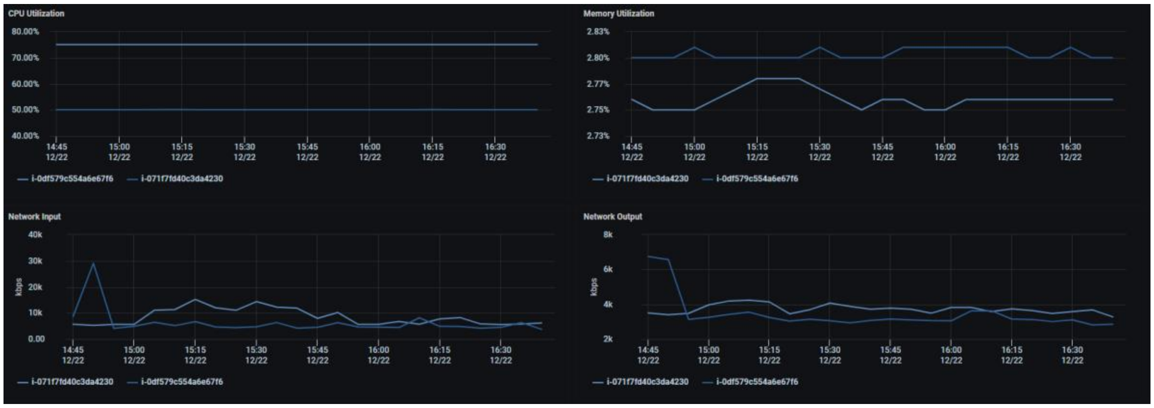
Total 9

VM cluster

CPU utilization, memory utilization, disk capacity, and disk IO utilization, network transmit and receive bytes charts are displayed for each Kubernetes node.



CPU utilization, memory utilization, and network input and output charts are displayed for each VM.



Related topics:

Search/Sort Information in Tables

Show/Hide Information in Charts

Zoom In/Out of Charts

Terminology

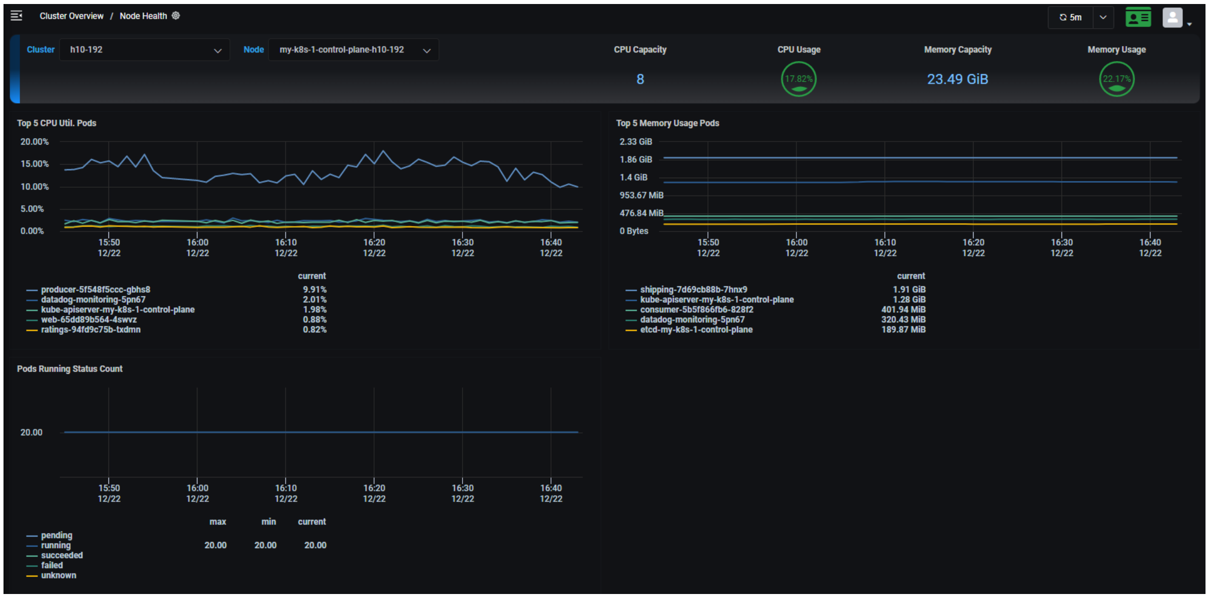
Cluster Overview - Node Health (Kubernetes)

The *Node Health* page displays actual usage observations about each node in a Kubernetes cluster. Select which cluster and node to display.

For the selected node, you will see the total CPU capacity and usage as well as memory capacity and usage.

The *Top 5* charts below display the CPU utilization of the top five pods on each node and the memory usage of the top five pods.

The *Pods Running Status Count* chart displays the number of pods running, the minimum and maximum number of pods that can run, along with the status of each pod.



Cluster: h10-192
Node: my-k8s-1-control-plane-h10-192

CPU Capacity	CPU Usage	Memory Capacity	Memory Usage
8	17.82%	23.49 GiB	22.17%

Top 5 CPU Util. Pods

	current
producer-5f548f5ccc-gbhs8	9.91%
datadog-monitoring-5pn67	2.01%
kube-apiserver-my-k8s-1-control-plane	1.98%
web-65dd89b564-4swvz	0.88%
ratings-94fd9c75b-txdmn	0.82%

Top 5 Memory Usage Pods

	current
shipping-7d69cb88b-7hnx9	1.91 GiB
kube-apiserver-my-k8s-1-control-plane	1.28 GiB
consumer-5b5f866fb6-828f2	401.94 MiB
datadog-monitoring-5pn67	320.43 MiB
etcd-my-k8s-1-control-plane	189.87 MiB

Pods Running Status Count

	max	min	current
pending			
running	20.00	20.00	20.00
succeeded			
failed			
unknown			

Related topics:

Search/Sort Information in Tables

Show/ Hide Information in Charts

Zoom In/Out of Charts

Terminology

Predictions and Planning – Kubernetes or VM Resources

The *Kubernetes Resources* page displays actual CPU and memory usage observations, predicted usage, and recommendations for resources in a Kubernetes cluster.

For Kubernetes, Federator.ai monitors resource usage for monitored clusters, nodes, namespaces, user-defined applications, and controllers and provides workload prediction, recommendations, and utilization analysis at each level.

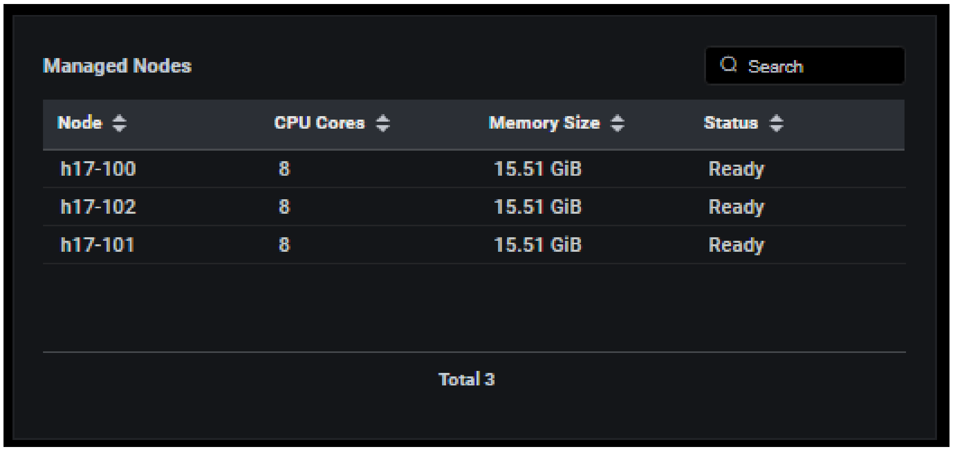
For VM, Federator.ai monitors resource usage for monitored clusters and VMs and provides workload prediction, recommendations, and utilization analysis at each level.

With the analysis and recommendations, you can decide if a resource is over-provisioned (wasting resources), or if it is under-provisioned and will not sustain an increased workload.

Use filters at the top of the page to select the level of information you want to display. When you select a Kubernetes namespace, the namespace status will be displayed.

Managed Nodes Table (Kubernetes)

This table shows the list of nodes in the selected cluster with the number of CPU cores, the size of memory, and the status for each node.



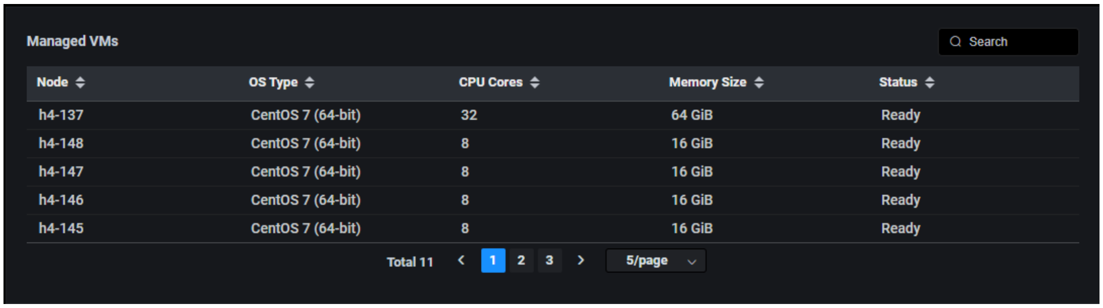
Managed Nodes

Node	CPU Cores	Memory Size	Status
h17-100	8	15.51 GiB	Ready
h17-102	8	15.51 GiB	Ready
h17-101	8	15.51 GiB	Ready

Total 3

Managed VMs Table (VM)

This table shows the list of VMs in the selected cluster with OS type, the number of CPU cores, the size of memory, and the status for each VM.



Managed VMs

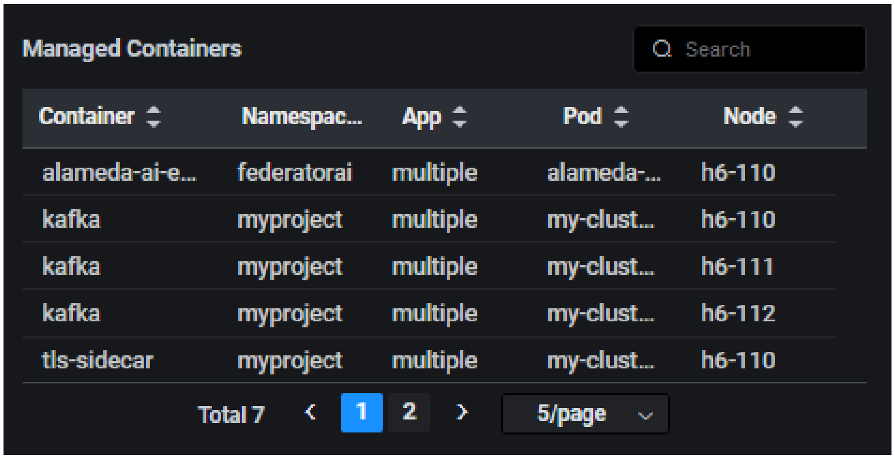
Search

Node	OS Type	CPU Cores	Memory Size	Status
h4-137	CentOS 7 (64-bit)	32	64 GiB	Ready
h4-148	CentOS 7 (64-bit)	8	16 GiB	Ready
h4-147	CentOS 7 (64-bit)	8	16 GiB	Ready
h4-146	CentOS 7 (64-bit)	8	16 GiB	Ready
h4-145	CentOS 7 (64-bit)	8	16 GiB	Ready

Total 11 ‹ 1 2 3 › 5/page

Managed Containers Table (Kubernetes)

Based on the selected scope (cluster, node, namespace, application, or controller), this table lists the containers for the scope. Each container is listed along with its namespace, application, Kubernetes pod name, and the node where this container runs.



Managed Containers

Search

Container	Namespac...	App	Pod	Node
alameda-ai-e...	federatorai	multiple	alameda-...	h6-110
kafka	myproject	multiple	my-clust...	h6-110
kafka	myproject	multiple	my-clust...	h6-111
kafka	myproject	multiple	my-clust...	h6-112
tls-sidecar	myproject	multiple	my-clust...	h6-110

Total 7 ‹ 1 2 › 5/page

Workload Prediction Table and Workload Observation and Prediction Charts

The *Workload Prediction* table displays daily, weekly, and monthly predictive CPU and memory data.

The *Workload Observation and Prediction* charts display observed actual usage for the selected time as well as predictive CPU and memory data:

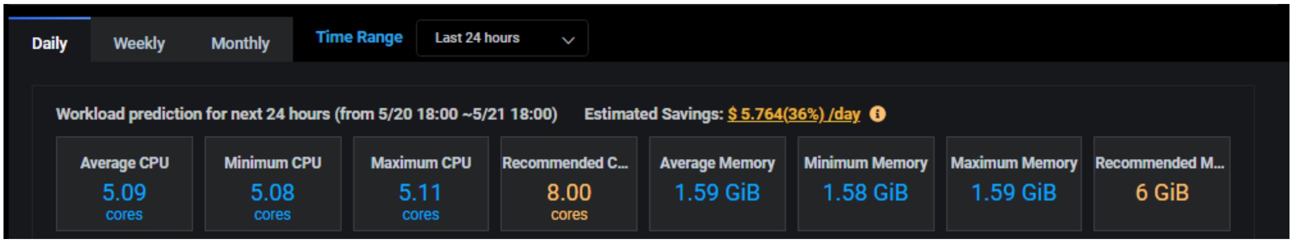
- Daily – Predicts CPU and memory usage every hour for the next 24 hours.
- Weekly – Predicts CPU and memory usage every 6 hours for the next 7 days.
- Monthly – Predicts CPU and memory usage every day for the next 30 days.

Use the *Time Range* field to set a custom time period for observed CPU and memory usage.

Workload Prediction Table

This table displays average/minimum/maximum CPU and memory usage and recommendations for the upcoming time selected - 24 hours (daily), 7 days (weekly), 30 days (monthly).

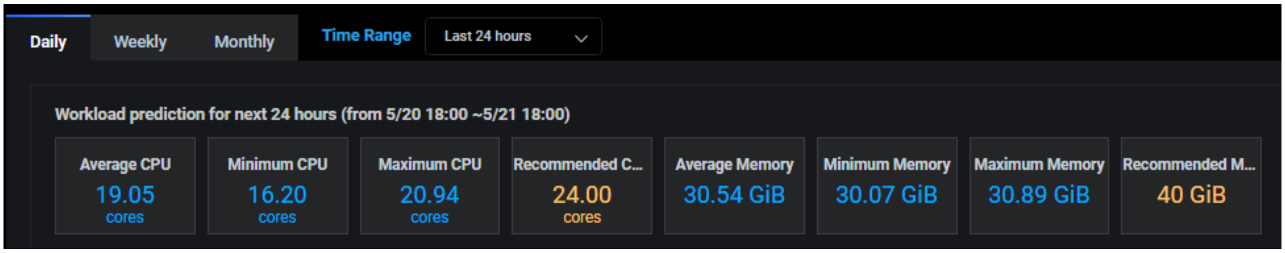
VM:



Workload prediction for next 24 hours (from 5/20 18:00 ~5/21 18:00) Estimated Savings: $ 5.764(36%) /day

Average CPU	Minimum CPU	Maximum CPU	Recommended C...	Average Memory	Minimum Memory	Maximum Memory	Recommended M...
5.09 cores	5.08 cores	5.11 cores	8.00 cores	1.59 GiB	1.58 GiB	1.59 GiB	6 GiB

Kubernetes:

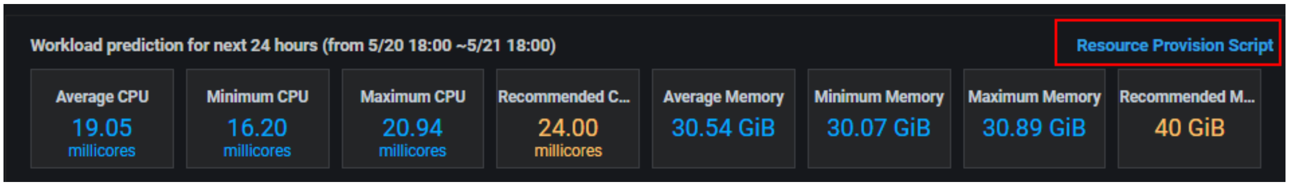


Workload prediction for next 24 hours (from 5/20 18:00 ~5/21 18:00)

Average CPU	Minimum CPU	Maximum CPU	Recommended C...	Average Memory	Minimum Memory	Maximum Memory	Recommended M...
19.05 cores	16.20 cores	20.94 cores	24.00 cores	30.54 GiB	30.07 GiB	30.89 GiB	40 GiB

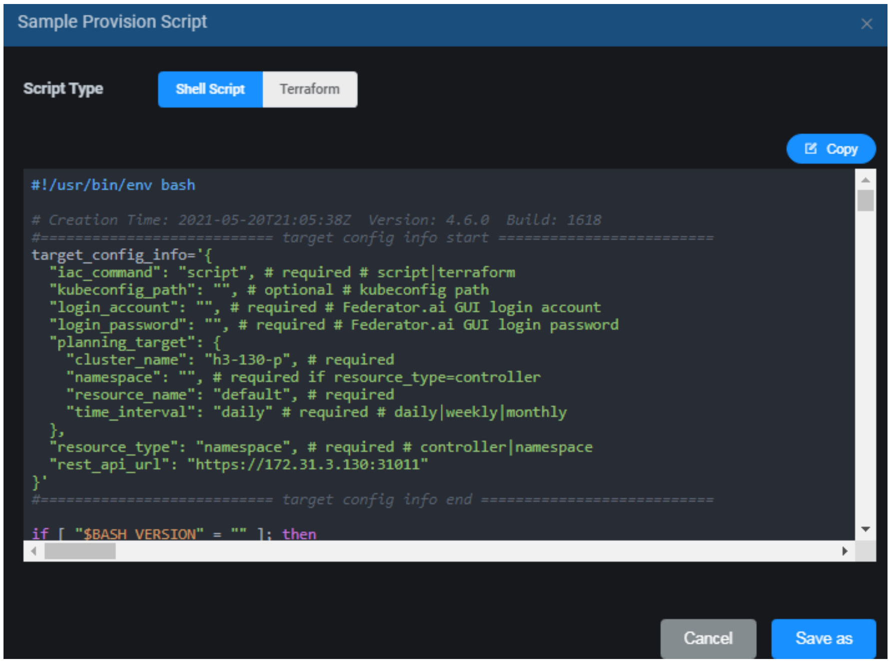
If you are displaying information for a Kubernetes namespace and the status is anything but *Monitoring*, this section will provide more information. For example, you will see the message, "Workload prediction is not configured for this namespace” or "Not enough information for predictions" for newly added, monitored namespaces.

For Kubernetes namespaces and controllers, Federator.ai provides a resource provisioning script that can be used to automatically apply the recommended CPU/memory for the namespace or controller. If an auto provisioning profile is assigned to a namespace or a controller, the resource provisioning script uses recommendations set by the auto provisioning profile. Otherwise, the resource provisioning script uses system recommendations based on the time frame you are viewing (daily/weekly/monthly). For remote Kubernetes clusters, you can copy a resource provisioning script to the remote cluster in order to run auto provisioning.



Workload prediction for next 24 hours (from 5/20 18:00 ~5/21 18:00)

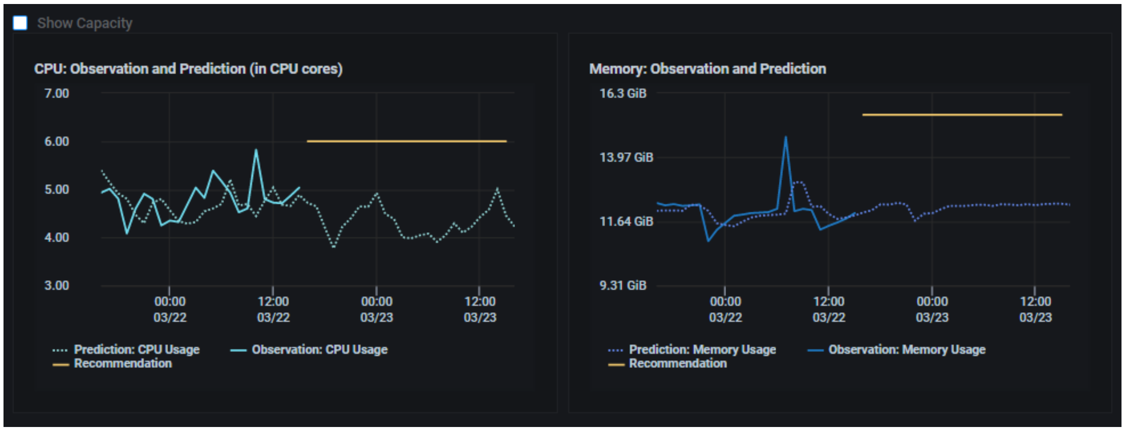
Average CPU	Minimum CPU	Maximum CPU	Recommended C...	Average Memory	Minimum Memory	Maximum Memory	Recommended M...
19.05 millicores	16.20 millicores	20.94 millicores	24.00 millicores	30.54 GiB	30.07 GiB	30.89 GiB	40 GiB



```
#!/usr/bin/env bash

# Creation Time: 2021-05-20T21:05:38Z  Version: 4.6.0  Build: 1618
#========================== target config info start ==========================
target_config_info='{
  "iac_command": "script", # required # script|terraform
  "kubeconfig_path": "", # optional # kubeconfig path
  "login_account": "", # required # Federator.ai GUI login account
  "login_password": "", # required # Federator.ai GUI login password
  "planning_target": {
    "cluster_name": "h3-130-p", # required
    "namespace": "", # required if resource_type=controller
    "resource_name": "default", # required
    "time_interval": "daily" # required # daily|weekly|monthly
  },
  "resource_type": "namespace", # required # controller|namespace
  "rest_api_url": "https://172.31.3.130:31011"
}'
#========================== target config info end ============================

if [ "$BASH VERSION" = "" ]; then
```

You can copy the script and run it in the Kubernetes cluster where the controller or the namespace is located. The script queries Federator.ai for the most recent recommendations and applies them to the controller or the namespace. Refer to Auto Provisioning Scripts for more information.

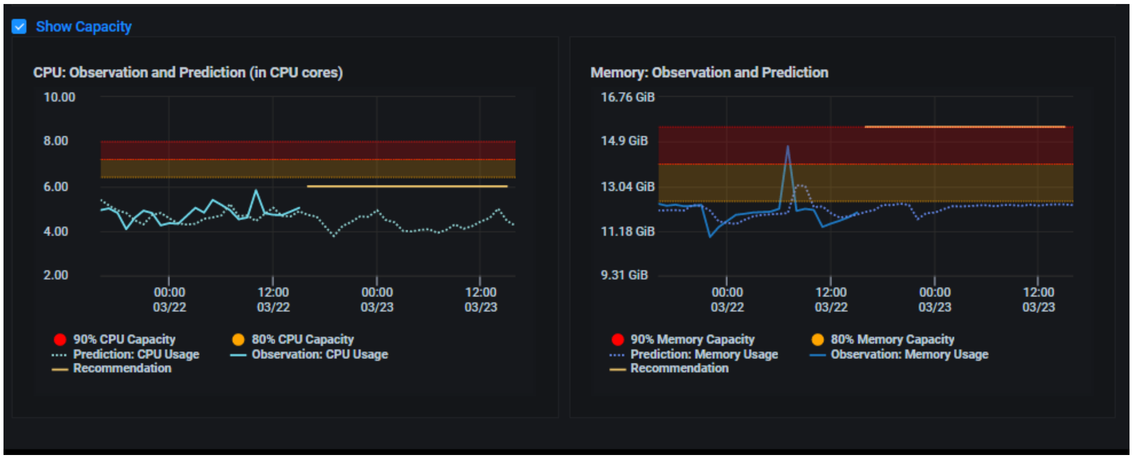
Workload Observation and Prediction Charts



These charts display CPU and memory observations and predictions for all resources specified in the *Filter* panel.

- The solid line represents the observed actual usage.
- The dotted green line represents the past and future predicted usage.
- The solid yellow line represents the recommended usage, which can help you from over-provisioning resources.
- If the selected scope is a Kubernetes node, the solid line represents the node's total CPU and memory. A big difference between total resources and actual and predicted usage can indicate that you are over-provisioned. A small difference between total resources and actual and predicted usage can indicate that you might be under-provisioned.

Check *Show Capacity* to see the maximum CPU and memory usage limits for the cluster, node, or VM. Orange represents 80-90% and red represents 90-100%. This is a useful way to see if the utilization of resources is approaching the overall capacity.

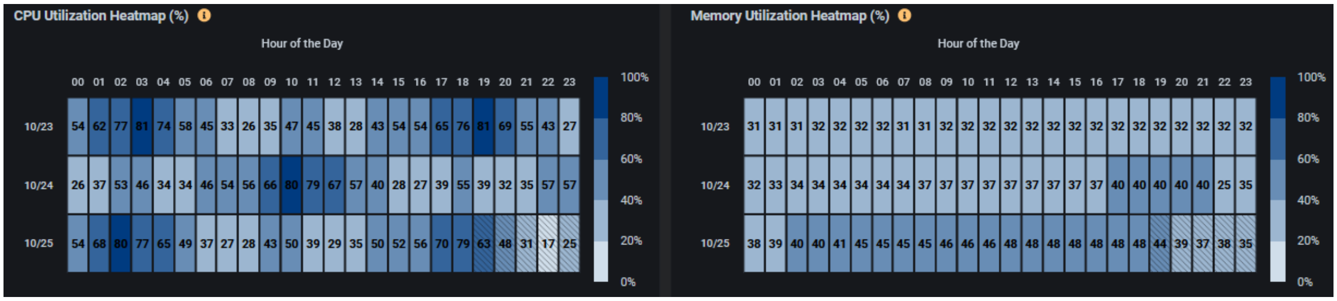


☑ Show Capacity

Utilization Analysis Charts

The *Utilization Analysis* charts display daily, weekly, and monthly CPU and memory utilization data for Kubernetes clusters, nodes, applications, and controllers and for VM clusters and VMs. Use filters to select the resources and the *Day/Week/Month* field to select a time period. For example, if *Weekly* is selected, use the *Week* field to select a different calendar week.

CPU and Memory Utilization Heatmap Charts



CPU Utilization Heatmap (%)

Hour of the Day

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
10/23	54	62	77	81	74	58	45	33	26	35	47	45	38	28	43	54	54	65	76	81	69	55	43	27
10/24	26	37	53	46	34	34	46	54	56	66	80	79	67	57	40	28	27	39	55	39	32	35	57	57
10/25	54	68	80	77	65	49	37	27	28	43	50	39	29	35	50	52	56	70	79	63	48	31	17	25

Memory Utilization Heatmap (%)

Hour of the Day

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
10/23	31	31	31	32	32	32	32	31	31	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
10/24	32	33	34	34	34	34	34	34	37	37	37	37	37	37	37	37	37	40	40	40	40	40	25	35
10/25	38	39	40	40	41	45	45	45	45	46	46	46	48	48	48	48	48	48	48	44	39	37	38	35

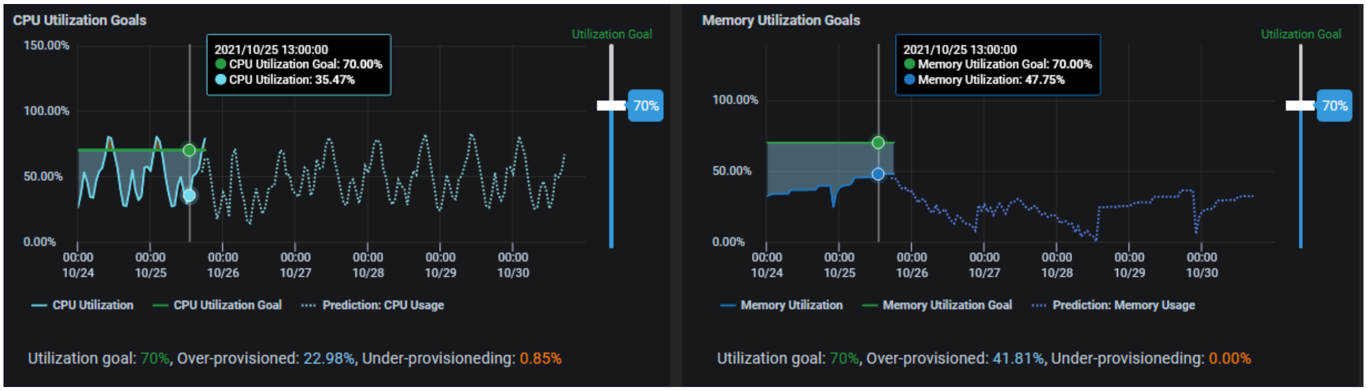
These charts display the actual and predicted CPU and memory usage percentage for all resources specified using filters for the selected time frame:

- Daily – Displays usage every hour for the last three days.
- Weekly – Displays usage each hour for a calendar week.
- Monthly – Displays usage every day for a calendar month.

For clusters and nodes, the percentage is calculated by actual usage divided by capacity. For applications and controllers, the percentage is calculated by actual usage divided by the requested (minimum) CPU/memory or the limit (maximum) CPU/memory.

The color gradient illustrates the percentage range, making it easy to see periods of high and low usage. Boxes with diagonal lines represent future predicted utilization.

CPU and Memory Utilization Goals Charts



Utilization goal: 70%, Over-provisioned: 22.98%, Under-provisioneding: 0.85%

Utilization goal: 70%, Over-provisioned: 41.81%, Under-provisioneding: 0.00%

These interactive charts display target goals along with actual and predicted CPU and memory usage for all resources specified by filters for the selected time frame.

- The blue line represents the actual CPU or memory utilization.
- The green line represents your utilization goal.
- The dotted blue line represents future predicted utilization.
- Gray areas represent periods of time when your resources were over-provisioned (wasted utilization).
- Orange areas represent periods of time when your resources were under-provisioned.

By comparing actual usage to your utilization goals, you can easily see where you are over- or under-provisioned, enabling you to adjust your cloud resources for more efficient usage. For example, if you see times when actual usage is consistently much lower than your utilization goals for a cluster, you may want to deploy additional applications in that cluster.

You can adjust your target utilization goals by using the slider on the right side of the chart.

Related topics:

Terminology
Search/Sort Information in Tables
Show/Hide Information in Charts
Zoom In/Out of Charts
Cluster Configuration
Auto Provisioning

Autoscaling – Generic Application (Kubernetes)

The *Generic Application* page displays usage and recommendation information about replicas for selected controllers. These controllers must be enabled with autoscaling during configuration. Refer to the *Configuration - Applications* section for information about how to enable autoscaling for a controller. When autoscaling is enabled, CPU and memory usage is monitored, and the number of pods is increased/decreased based on the workload. An autoscaled pod is called a *replica*.

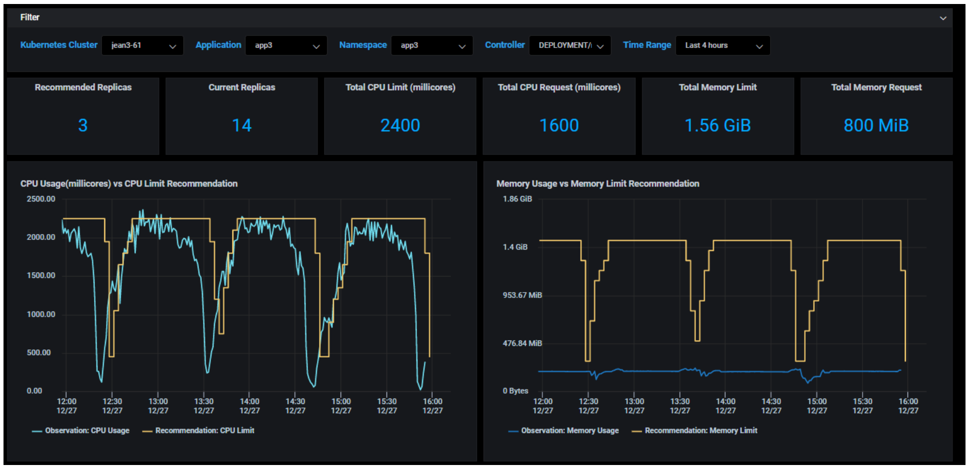
Use the correct namespace, application, and controller name to see the history of autoscaling for a specific controller. The historical number of replicas and CPU/memory usage for the last 1, 2, 4, 6, or 12 hours can be displayed and can further be filtered by selecting a range of dates.

When the system was configured, the requested (minimum) CPU/memory and the limit (maximum) CPU/memory were set. This page shows the number of recommended and current replicas along with the total CPU/memory limit/request being used by all current replicas.

CPU and Memory Charts

These charts display CPU and memory usage and recommendations.

- The blue line represents the observed actual usage.
- The yellow line represents the limits after autoscaling.



Filter

Kubernetes Cluster: jean3-61
Application: app3
Namespace: app3
Controller: DEPLOYMENT/
Time Range: Last 4 hours

Recommended Replicas	Current Replicas	Total CPU Limit (millicores)	Total CPU Request (millicores)	Total Memory Limit	Total Memory Request
3	14	2400	1600	1.56 GiB	800 MiB

Related topics:

Common Administration Portal Functions

Terminology

Configuration

Search/Sort Information in Tables

Show/Hide Information in Charts

Zoom In/Out of Charts

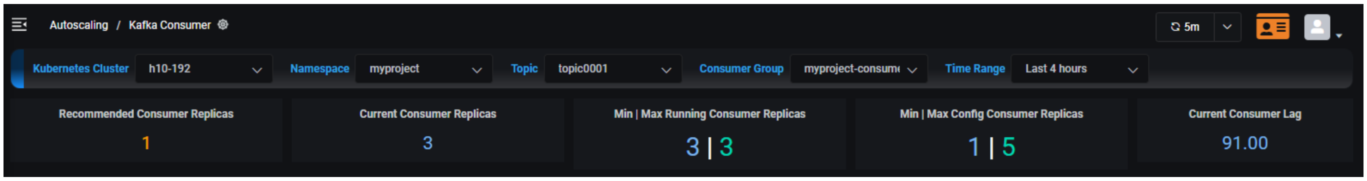
Configure Applications

Autoscaling – Kafka Consumer (Kubernetes)

The traditional method of autoscaling, based on consumer CPU/memory usage, is not enough to achieve desired performance goals for Kafka. Kafka production and consumption of messages (including message arrival rate and consumer lag) offer a better indicator of workload and performance, and is monitored by Federator.ai. Using message production rate predictions, Federator.ai autoscales the number of Kafka consumer pods to fit the workload and optimize performance.

If you have configured Federator.ai to monitor and autoscale Kafka consumers, the *Kafka Consumer* page displays predictions for the message production rate and Federator.ai scales Kafka consumer replicas to satisfy the workload. You can configure multiple Kafka topics and consumer groups. Federator.ai will predict and autoscale consumers for each individual topic/consumer group. Refer to the Add an Application section for information about how to configure Kafka consumer monitoring and autoscaling.

Use filters at the top of the page to select the correct cluster, namespace, topic, and consumer group for the Kafka consumer being monitored. The number of replicas and Kafka message production/consumption rate and the prediction of message production rate for the last 1, 2, 4, 6, or 12 hours can be displayed and can further be filtered by selecting a range of dates.



Kubernetes Cluster: h10-192 | Namespace: myproject | Topic: topic0001 | Consumer Group: myproject-consum | Time Range: Last 4 hours

Recommended Consumer Replicas	Current Consumer Replicas	Min | Max Running Consumer Replicas	Min | Max Config Consumer Replicas	Current Consumer Lag
1	3	3 | 3	1 | 5	91.00

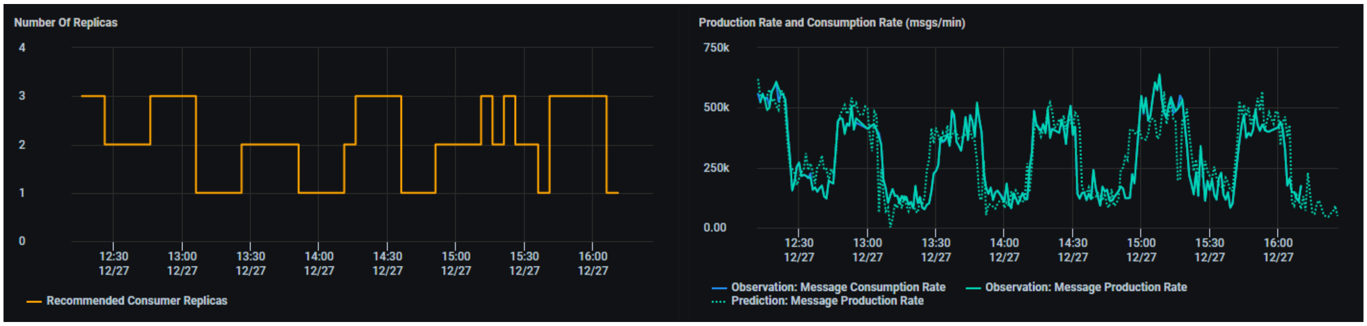
The number of recommended and current consumer replicas is displayed along with the minimum and maximum number of running and configured replicas and the current consumer lag.

Number of Replicas Chart

The *Number of Replicas* chart displays the Kafka consumer replicas for the specified time range as a result of recommendations from Federator.ai.

Production Rate and Consumption Rate Chart

The *Production Rate and Consumption Rate* chart displays the actual observed message production rate (solid green line) and consumption/processed rate (solid blue line) for the specified time range and the historical and future predicted rate (dotted green line).

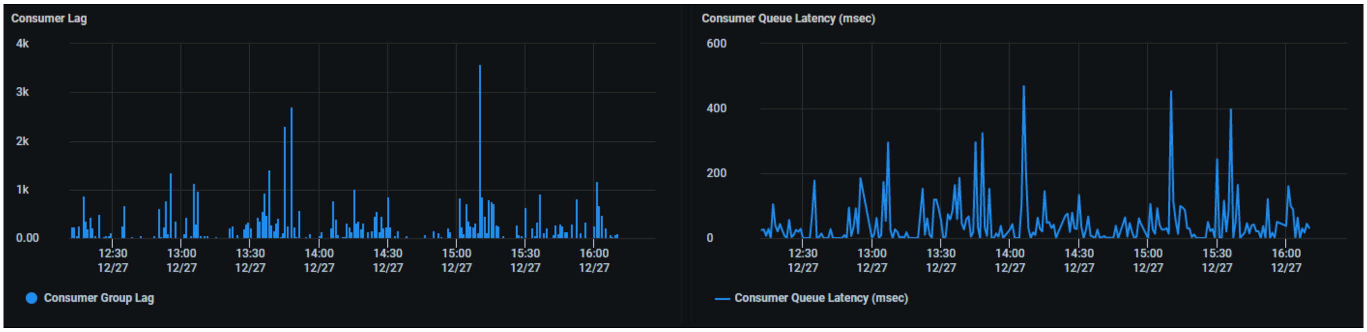


Consumer Lag Chart

The *Consumer Lag* chart displays the number of messages in the Kafka brokers that are yet to be processed by the Kafka consumers for the selected topic and consumer group and time range. This represents messages in the queue waiting to be processed.

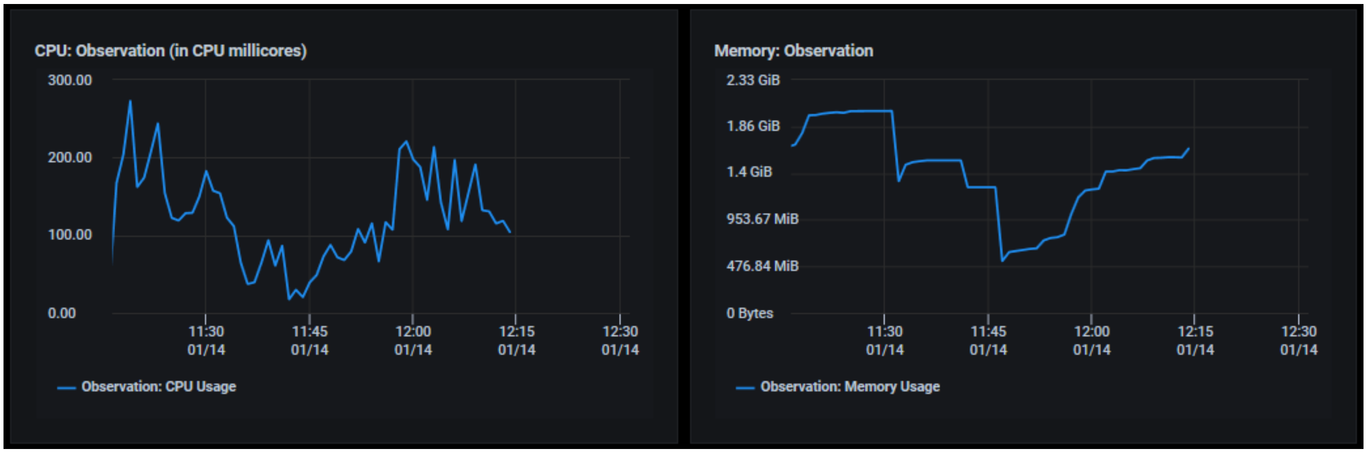
Consumer Queue Latency Chart

The *Consumer Queue Latency* chart displays how long it takes for each message to be processed for the selected time range.



CPU and Memory Observation Charts

The *CPU Observation* and *Memory Observation* charts display observed actual usage for the selected time range. This reflects the resources used as a result of autoscaling consumer pods.



Related topics:

Common Administration Portal Functions

Terminology

Show/Hide Information in Charts

Zoom In/Out of Charts

Setup Wizard

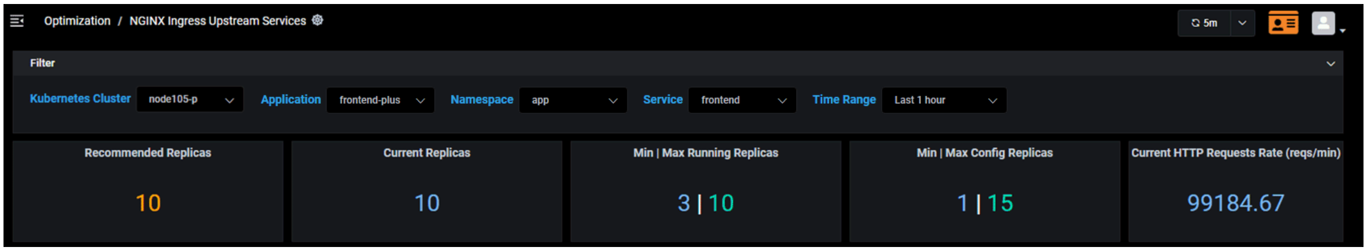
Configure Applications

Autoscaling – Ingress Upstream Services (Kubernetes)

The traditional method of autoscaling, based on CPU/memory usage, is not enough to achieve desired performance goals for Ingress services. Key performance goals include the ability of Ingress to forward requests to upstream services with minimal response time and errors; this provides a better indicator of workload and performance, and is monitored by Federator.ai. Using HTTP request rate predictions, Federator.ai autoscales the number of services to fit the workload and optimize performance.

If you have configured Federator.ai to monitor and autoscale Ingress upstream services, the *Ingress Upstream Services* page displays predictions for the HTTP request rate and Federator.ai scales replicas to satisfy the workload. You can configure multiple upstream services. Federator.ai will predict and autoscale for each individual service. Refer to the Add an Application section for information about how to configure Ingress upstream services for autoscaling.

Use filters at the top of the page to select the correct cluster, namespace, application, and service. The information and predictions for the last 1, 2, 4, 6, or 12 hours can be displayed and can further be filtered by selecting a range of dates.



Kubernetes Cluster: node105-p
Application: frontend-plus
Namespace: app
Service: frontend
Time Range: Last 1 hour

Recommended Replicas	Current Replicas	Min | Max Running Replicas	Min | Max Config Replicas	Current HTTP Requests Rate (reqs/min)
10	10	3 | 10	1 | 15	99184.67

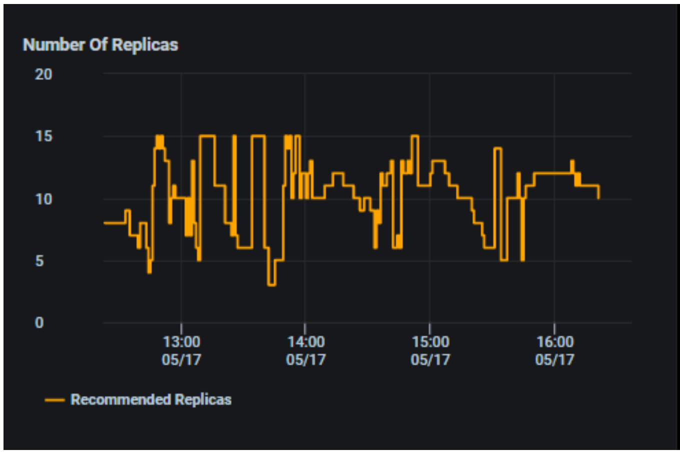
The number of recommended and current replicas is displayed along with the minimum and maximum number of running and configured replicas and the current HTTP request rate.

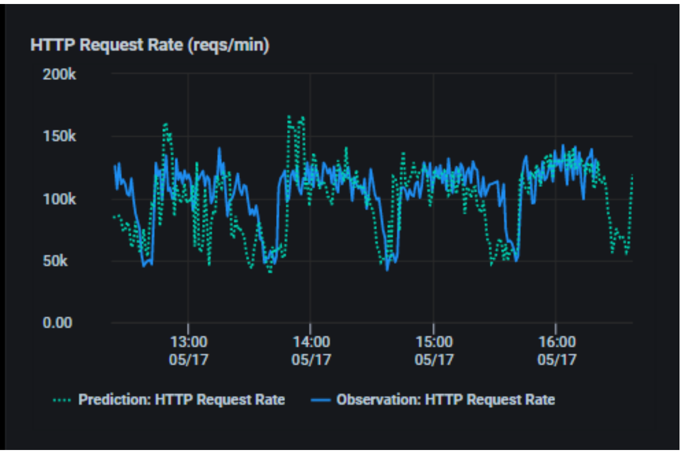
Number of Replicas Chart

The *Number of Replicas* chart displays the number of replicas of an upstream service for the specified time range as a result of recommendations from Federator.ai.

HTTP Request Rate Chart

The *HTTP Request Rate* chart displays the actual observed HTTP request rate (solid blue line) for the specified time range and the historical and future predicted rate (dotted green line).



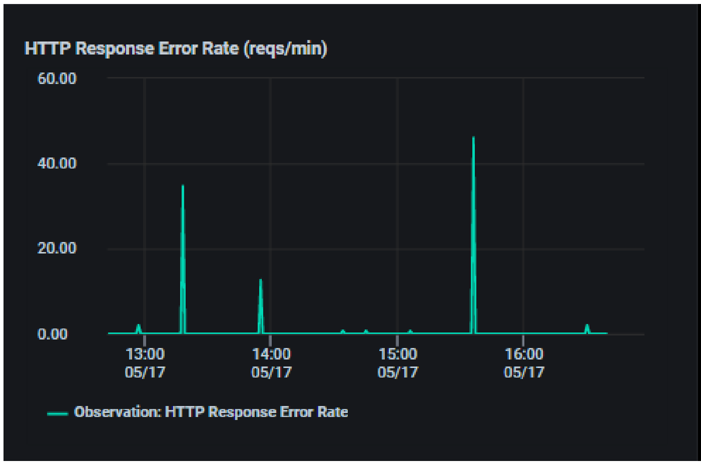


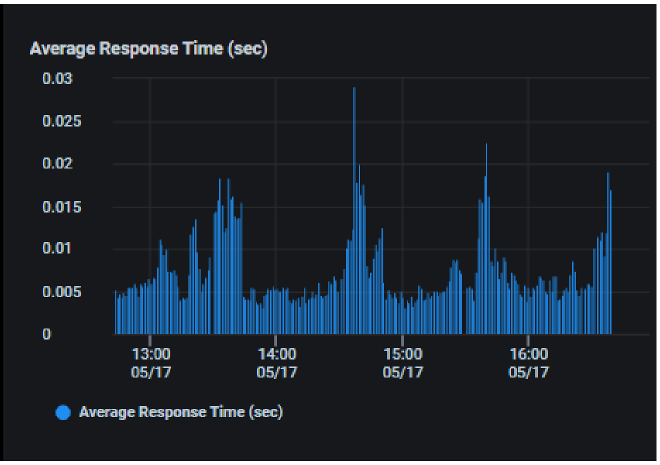
HTTP Response Error Rate Chart

The *HTTP Response Error Rate* chart displays the actual observed number of 5xx server errors for the specified time range. The goal is a zero-error rate.

Average Response Time Chart

The *Average Response Time* chart displays the average amount of time, in seconds, it takes for HTTP requests to be processed for the specified time range.



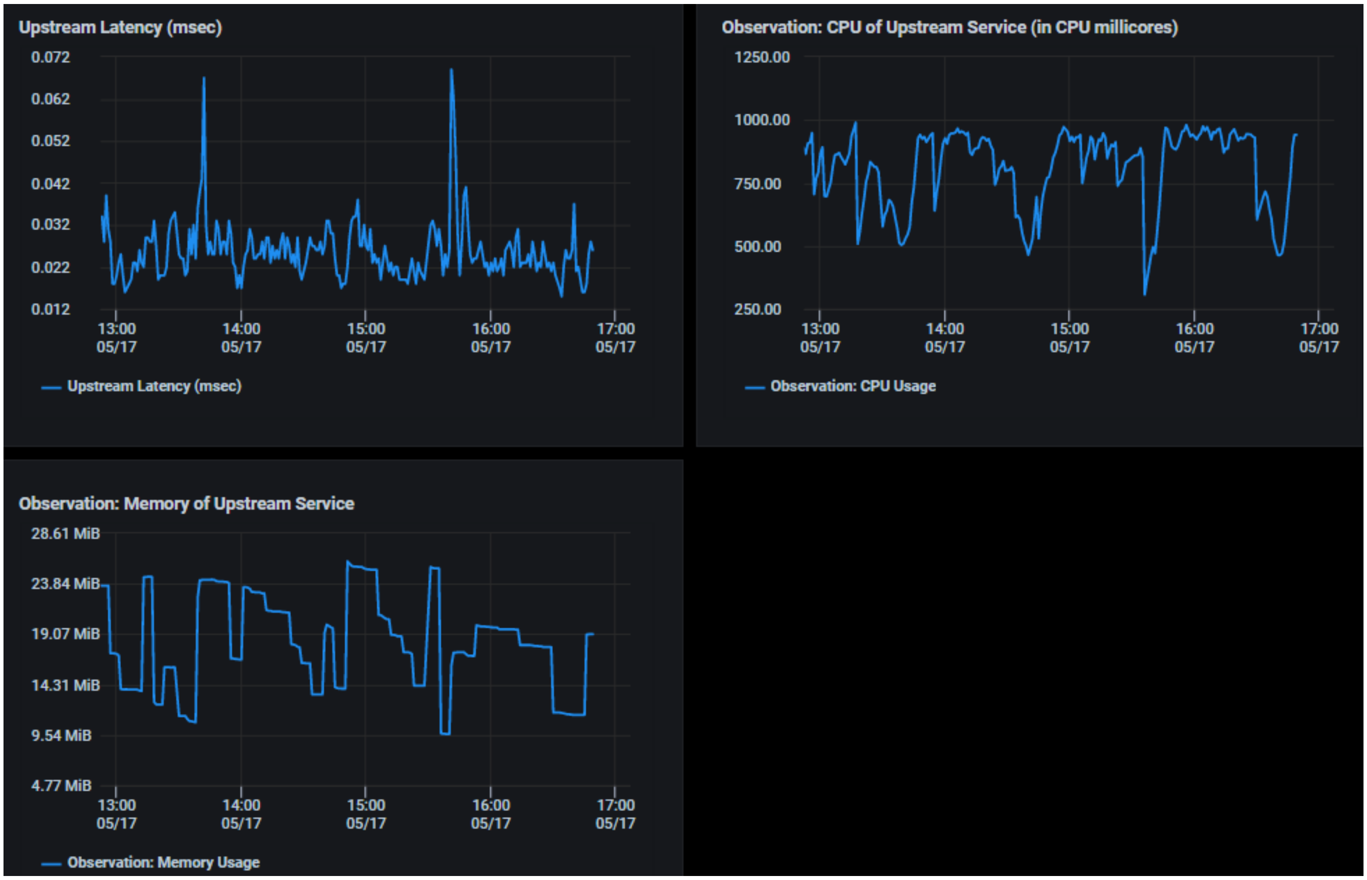


Upstream Latency Chart

The *Upstream Latency* chart displays the actual average delay for a request to upstream, in milliseconds, for the specified time range.

CPU and Memory Observation Charts

The *CPU Observation* and *Memory Observation* charts display observed actual usage of the upstream service for the selected time range. This reflects the services used as a result of autoscaling.



Related topics:

Common Administration Portal Functions

Terminology

Show/Hide Information in Charts

Zoom In/Out of Charts

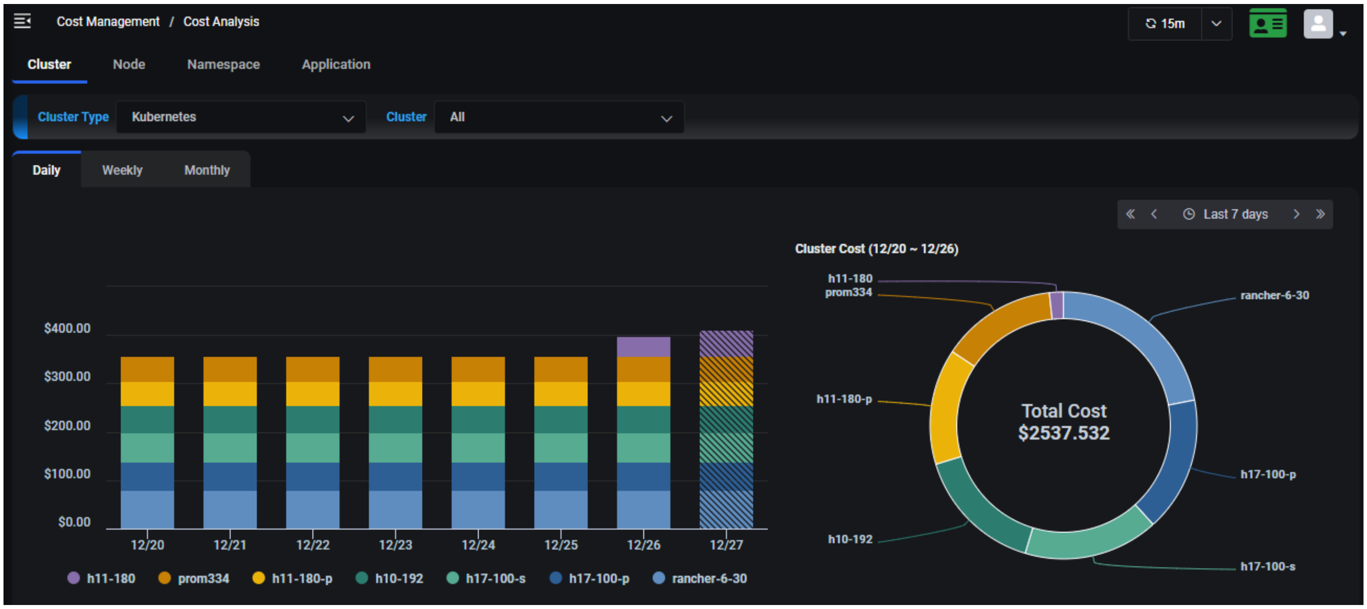
Setup Wizard

Configure Applications

Cost Management – Cost Analysis

The *Cost Analysis* page displays daily, weekly, and monthly costs and predictions. By default, information is displayed for all clusters but can be changed to display one or more Kubernetes or VM clusters or nodes, or one or more Kubernetes namespaces or applications.

Cost Analysis Charts



Cost Management / Cost Analysis

Cluster Type: Kubernetes

Cluster: All

Last 7 days

Cluster Cost (12/20 ~ 12/26)

Total Cost $2537.532

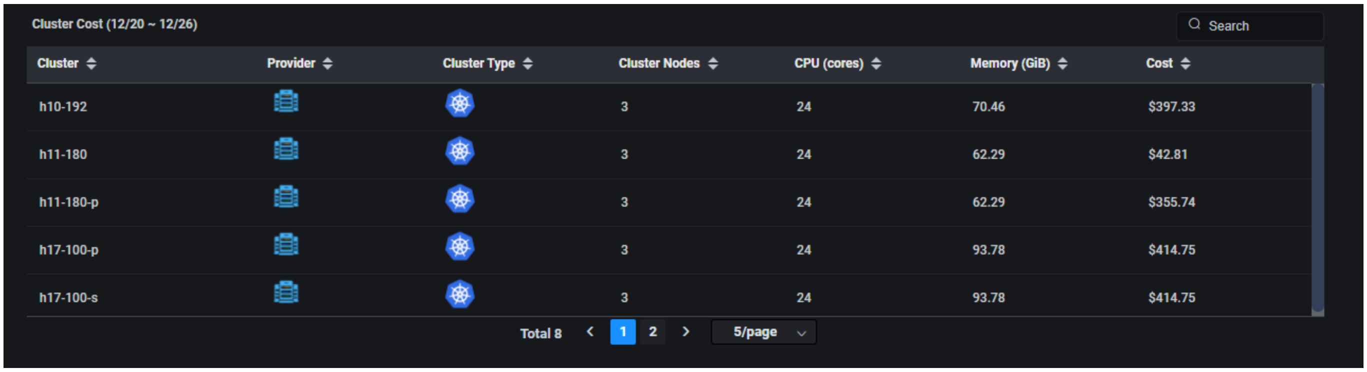
The left chart displays actual costs and predictions for the specified time frame:

- Daily – Displays costs for each day in the last seven days but can be changed to the last 14 or 28 days, or to a custom time range.
- Weekly – Displays costs for each week in the last four weeks but can be changed to the last eight or 12 weeks, or to a custom time range
- Monthly – Displays costs for the last three months but can be changed to the last six or nine months, or to a custom time range.

Click anywhere on the chart to see values for a specific point in time. Highlight or click on the key at the bottom of the chart to show/hide individual metrics (i.e., clusters).

The right chart displays the total cost for all clusters. Highlight a section of the chart to see values for a specific cluster.

Cost Analysis Summary Table



Cluster Cost (12/20 ~ 12/26)

Cluster	Provider	Cluster Type	Cluster Nodes	CPU (cores)	Memory (GiB)	Cost
h10-192	▤	☸	3	24	70.46	$397.33
h11-180	▤	☸	3	24	62.29	$42.81
h11-180-p	▤	☸	3	24	62.29	$355.74
h17-100-p	▤	☸	3	24	93.78	$414.75
h17-100-s	▤	☸	3	24	93.78	$414.75

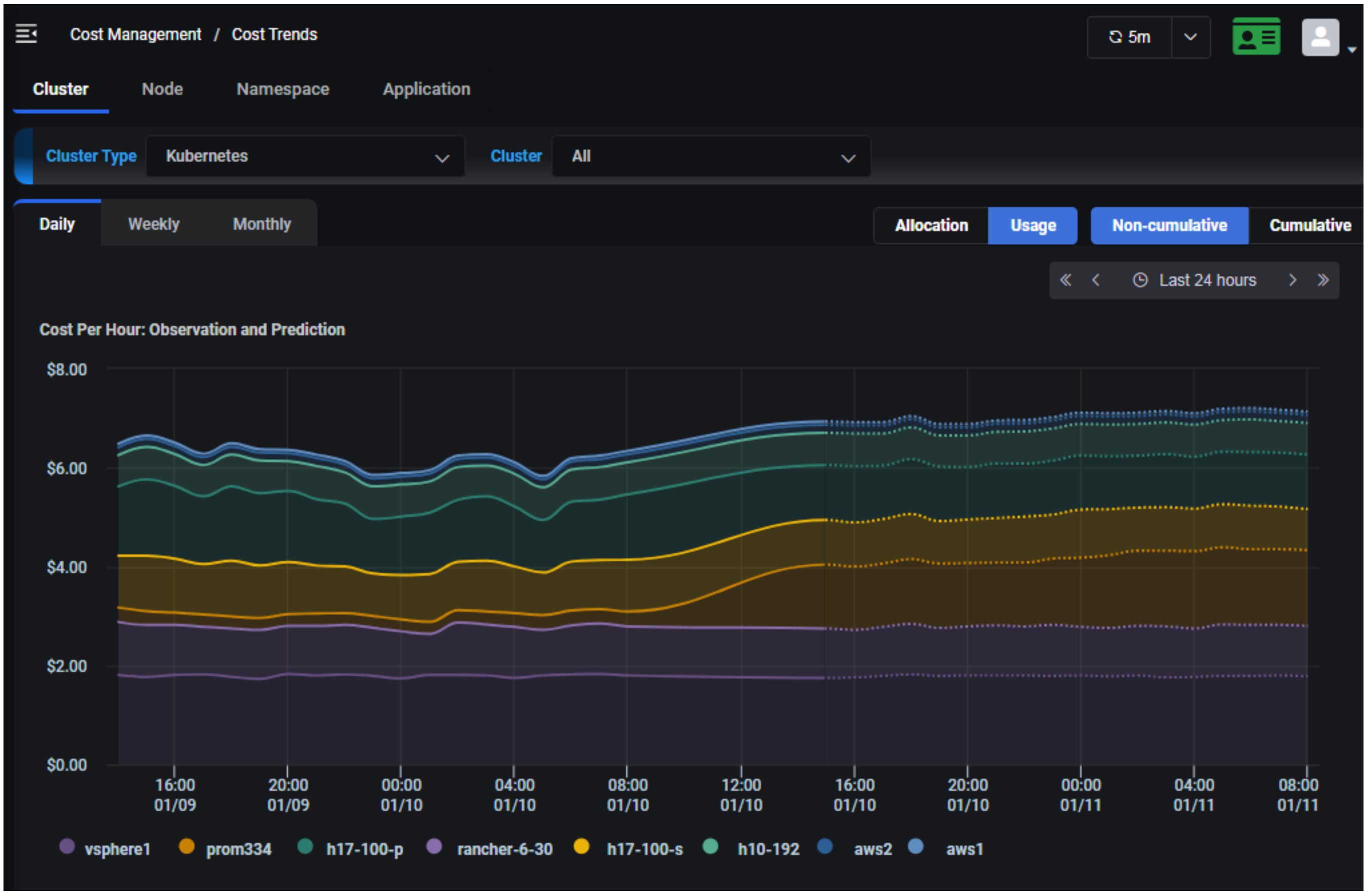
Summary information includes:

- Cluster – Cluster name, provider (AWS, Azure, Google, or on-premises), cluster type (Kubernetes, VM), number of cluster nodes, CPU cores and memory capacity allocated for the cluster during the specified time frame, cost for each cluster for the specified time frame.
- Node – Node name, cluster name, provider, cluster type, instance type used at your cloud provider, CPU cores and memory capacity allocated for the node during the specified time frame, cost for each node for the specified time frame.
- Namespace – Kubernetes namespace name, cluster name, provider, cluster type, number of CPU millicores and amount of memory used during the specified time frame, cost for each namespace for the specified time frame.
- Application – Kubernetes application name, cluster name, provider, cluster type, namespace, CPU millicore and memory requests/limits, cost for each application for the specified time frame.

Cost Management – Cost Trends

The *Cost Trends* page displays daily, weekly, and monthly costs and predictions based on your expected workload. By default, information is displayed for all clusters but can be changed to display one or more Kubernetes or VM clusters or nodes, or one or more Kubernetes namespaces or applications.

Cost Trends Chart



The chart displays actual costs and predictions for the specified time frame:

- Daily – Displays hourly costs for the last 24 hours and predictions for the next 24 hours but can be changed to display costs for the last two or four days, or to a custom time range.
- Weekly – Displays hourly costs for the last seven days and predictions for the next seven days but can be changed to display costs for the last two or four weeks, or to a custom time range.
- Monthly – Displays daily costs for the last 30 days and predictions for the next month but can be changed to display costs for the last two, three, or four months, or to a custom time range.

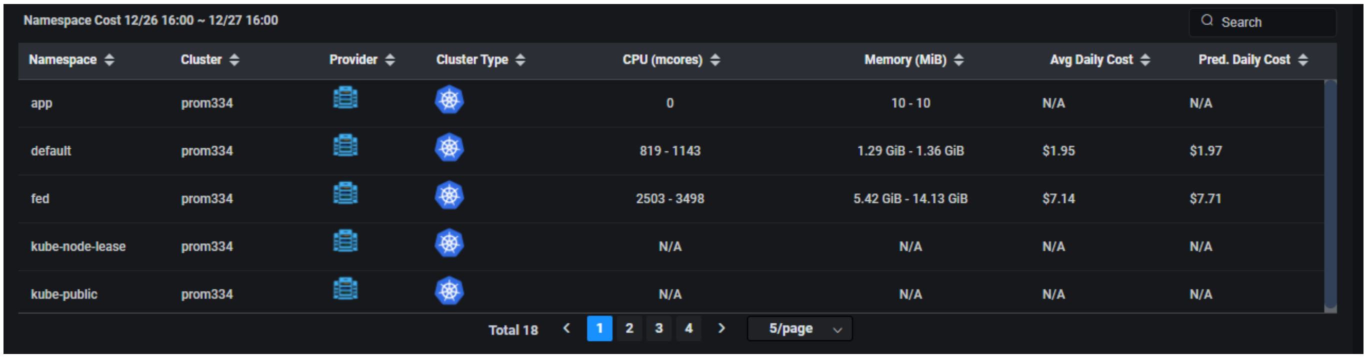
For clusters and nodes, when all clusters/nodes are displayed, you can select to view cost based on allocation or usage. Click *Allocation* to see fixed costs for cluster/node capacity, which will be consistent, as they do not fluctuate much unless a cluster node is added or removed. Click *Usage* to see the cost based on actual usage. When an individual cluster is selected, allocation and usage will both be displayed.

For namespaces and applications, the cost trends are based on the actual usage of resources.

Click the *Cumulative* button to view trends cumulatively through the end of the selected time period.

Click anywhere on the chart to see values for a specific point in time. Click on the key at the bottom of the chart to show/hide individual metrics (i.e., namespaces).

Cost Trends Summary Table



Namespace Cost 12/26 16:00 ~ 12/27 16:00

Search

Namespace	Cluster	Provider	Cluster Type	CPU (mcores)	Memory (MiB)	Avg Daily Cost	Pred. Daily Cost
app	prom334	▤	☸	0	10 - 10	N/A	N/A
default	prom334	▤	☸	819 - 1143	1.29 GiB - 1.36 GiB	$1.95	$1.97
fed	prom334	▤	☸	2503 - 3498	5.42 GiB - 14.13 GiB	$7.14	$7.71
kube-node-lease	prom334	▤	☸	N/A	N/A	N/A	N/A
kube-public	prom334	▤	☸	N/A	N/A	N/A	N/A

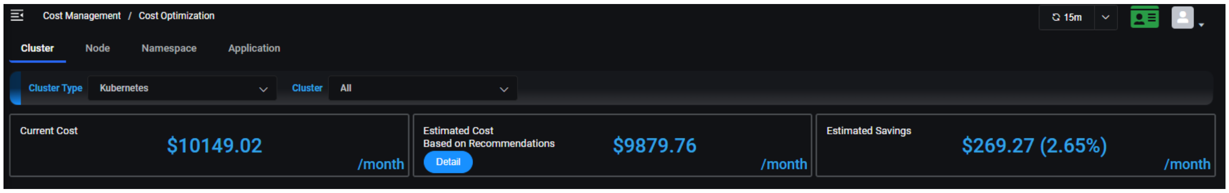
Summary information includes:

- Cluster – Cluster name, provider (AWS, Azure, Google, or on-premises), cluster type (Kubernetes, VM), number of cluster nodes, CPU cores and memory capacity allocated for the cluster during the specified time frame, average and predicted cost for each cluster for the specified time frame (costs are typically the same, unless a cluster node is added or removed).
- Node – Node name, cluster name, provider, cluster type, instance type used at your cloud provider, CPU cores and memory capacity allocated for the node during the specified time frame, average and predicted cost for each node for the specified time frame (costs are typically the same, unless a node is added or removed).
- Namespace – Kubernetes namespace name, cluster name, provider, cluster type, amount CPU/memory used during the specified timeframe, average and predicted cost for each namespace for the specified time frame.
- Application – Kubernetes application name, cluster name, provider, cluster type, CPU millicore and memory requests/limits, average and predicted cost for each application for the specified time frame.

Cost Management – Cost Optimization

The *Cost Optimization* page displays daily, weekly, and monthly recommendations and potential savings based on your expected workload. By default, information is displayed for all clusters but can be changed to display one or more Kubernetes or VM clusters or nodes, or one or more Kubernetes namespaces or applications.

Cost Optimization Chart



Cost Management / Cost Optimization

15m

Cluster | Node | Namespace | Application

Cluster Type: Kubernetes

Cluster: All

Current Cost $10149.02 /month

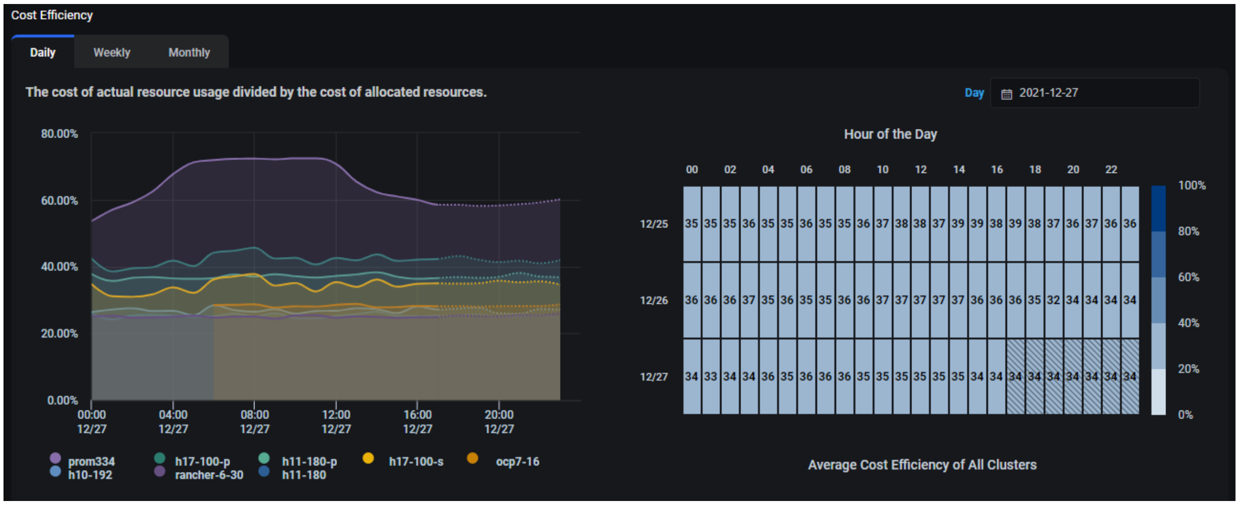
Estimated Cost Based on Recommendations $9879.76 /month

Detail

Estimated Savings $269.27 (2.65%) /month

This chart displays your current monthly cost with your existing resource configuration, the estimated monthly cost based on recommendations, and the potential monthly savings. Click the *Detail* link to view the recommendations (further down the page) to see how these savings are calculated. The savings typically come from reducing idle resources. The information continually refreshes itself as new data becomes available.

Cost Efficiency Charts



Cost Efficiency

Daily | Weekly | Monthly

The cost of actual resource usage divided by the cost of allocated resources.

Day: 2021-12-27

Hour of the Day

	00		02		04		06		08		10		12		14		16		18		20		22	
12/25	35	35	35	36	35	35	36	35	35	36	37	38	38	37	39	39	38	39	38	37	36	37	36	36
12/26	36	36	36	37	35	36	35	35	36	36	37	37	37	37	37	36	36	36	35	32	34	34	34	34
12/27	34	33	34	34	36	35	36	36	36	35	35	35	35	35	35	34	34	34	34	34	34	34	34	34

Average Cost Efficiency of All Clusters

The *Cost Efficiency* charts display the actual cost of resource usage divided by the cost of allocated resources for the specified time frame as well as cost efficiency predictions going forward. Information is displayed in a time-series graph and heatmap for each cluster/node/namespace/application or an average for all clusters/nodes/namespaces/applications, depending upon what was selected. The lower the percentage, the more wasteful your configuration. The goal is to get to as close to 100% cost efficiency without performance risk as possible.

- Daily – Displays hourly cost efficiency from the start of today and predictions going forward for the day but can be changed to display cost efficiency for a different day.
- Weekly – Displays hourly cost efficiency for the current week and predictions going forward for the week but can be changed to display cost efficiency for a different week.

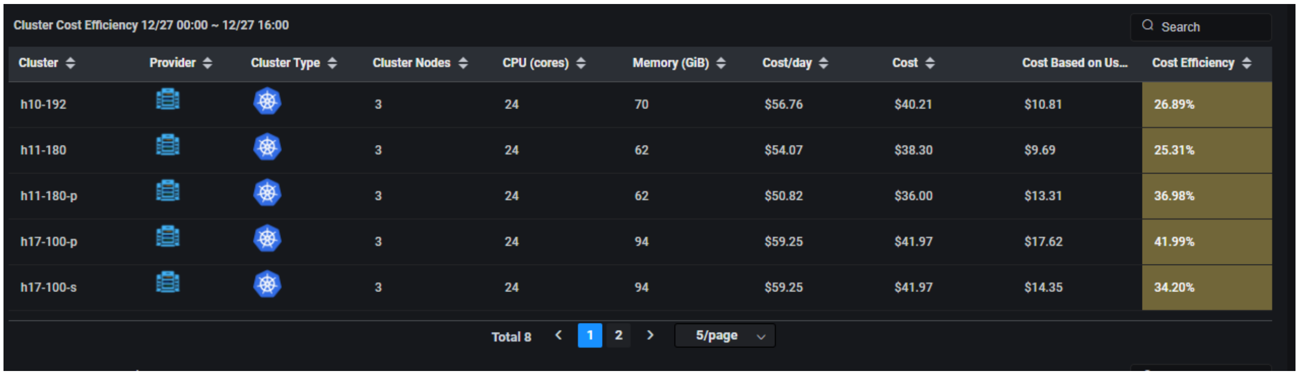
- Monthly – Displays daily cost efficiency for the current month and predictions going forward for the month but can be changed to display cost efficiency for a different month.

Click anywhere on either chart to see values for a specific point in time. Click on the key at the bottom of the graph to show/hide individual metrics (i.e., clusters).

The color gradient in the heatmap illustrates the percentage range, making it easy to see periods of high and low usage. Boxes with diagonal lines represent future predicted cost efficiency values.

Cluster Cost Optimization

Cluster Cost Efficiency Table

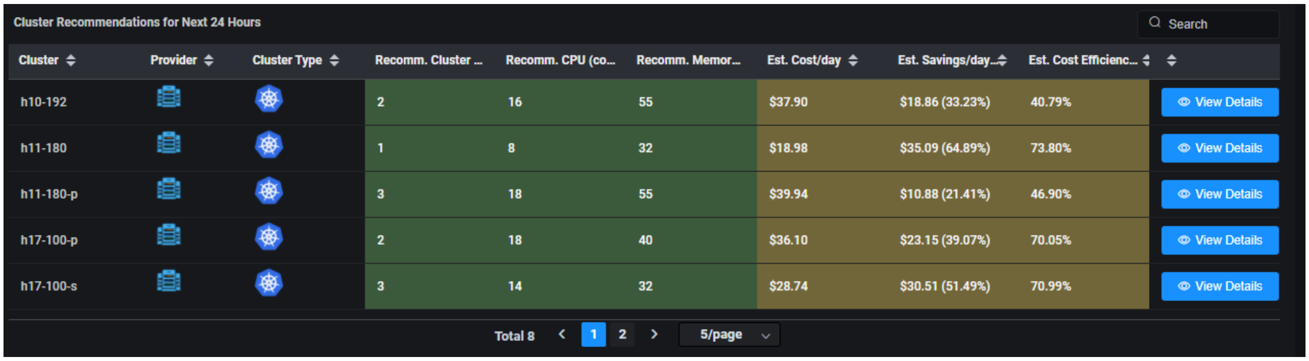


Cluster Cost Efficiency 12/27 00:00 ~ 12/27 16:00

Cluster	Provider	Cluster Type	Cluster Nodes	CPU (cores)	Memory (GiB)	Cost/day	Cost	Cost Based on Us...	Cost Efficiency
h10-192	▤	☸	3	24	70	$56.76	$40.21	$10.81	26.89%
h11-180	▤	☸	3	24	62	$54.07	$38.30	$9.69	25.31%
h11-180-p	▤	☸	3	24	62	$50.82	$36.00	$13.31	36.98%
h17-100-p	▤	☸	3	24	94	$59.25	$41.97	$17.62	41.99%
h17-100-s	▤	☸	3	24	94	$59.25	$41.97	$14.35	34.20%

Cluster Cost Efficiency information includes cluster name, provider (AWS, Azure, Google, or on-premises), cluster type (Kubernetes, VM), number of cluster nodes, CPU cores and memory capacity allocated for the cluster during the specified time frame, cost for each cluster for the specified time frame, cost from the actual cluster usage, and the cost efficiency percentage (the cost from actual cluster usage divided by the allocated cost) for the specified time frame.

Cluster Recommendations Table



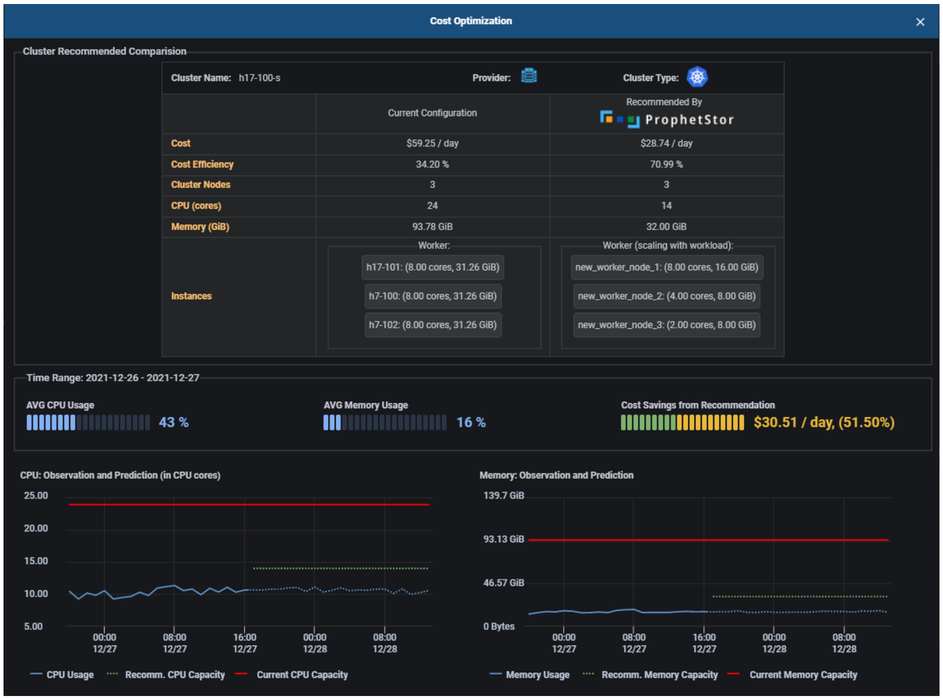
Cluster Recommendations for Next 24 Hours

Cluster	Provider	Cluster Type	Recomm. Cluster ...	Recomm. CPU (co...	Recomm. Memor...	Est. Cost/day	Est. Savings/day...	Est. Cost Efficienc...	
h10-192	▤	☸	2	16	55	$37.90	$18.86 (33.23%)	40.79%	View Details
h11-180	▤	☸	1	8	32	$18.98	$35.09 (64.89%)	73.80%	View Details
h11-180-p	▤	☸	3	18	55	$39.94	$10.88 (21.41%)	46.90%	View Details
h17-100-p	▤	☸	2	18	40	$36.10	$23.15 (39.07%)	70.05%	View Details
h17-100-s	▤	☸	3	14	32	$28.74	$30.51 (51.49%)	70.99%	View Details

The *Cluster Recommendations* table shows the changes you can make to save money. For example, it may show that your cost efficiency is 60%, meaning you are only using 60% of your currently allocated resources, and based on your predicted workload, four nodes are sufficient instead of the five current ones.

The table displays cluster name, provider (AWS, Azure, Google, or on-premises), cluster type (Kubernetes, VM), recommended number of cluster nodes, CPU cores, and memory, as well as the estimated cost, savings, and cost efficiency percentage (the cost from predicted cluster usage divided by the allocated cost) per day/week/month for the cluster if the recommendations are followed. Click the *View Details* link to view the recommendations to see how these savings are calculated.

Cluster Cost Optimization Details



Cost Optimization ×

Cluster Recommended Comparision

Cluster Name: h17-100-s   Provider: Cluster Type:

	Current Configuration	Recommended By ProphetStor
Cost	$59.25 / day	$28.74 / day
Cost Efficiency	34.20 %	70.99 %
Cluster Nodes	3	3
CPU (cores)	24	14
Memory (GiB)	93.78 GiB	32.00 GiB
Instances	Worker: h17-101: (8.00 cores, 31.26 GiB) h7-100: (8.00 cores, 31.26 GiB) h7-102: (8.00 cores, 31.26 GiB)	Worker (scaling with workload): new_worker_node_1: (8.00 cores, 16.00 GiB) new_worker_node_2: (4.00 cores, 8.00 GiB) new_worker_node_3: (2.00 cores, 8.00 GiB)

Time Range: 2021-12-26 - 2021-12-27

AVG CPU Usage 43 %

AVG Memory Usage 16 %

Cost Savings from Recommendation $30.51 / day, (51.50%)

CPU: Observation and Prediction (in CPU cores)

Memory: Observation and Prediction

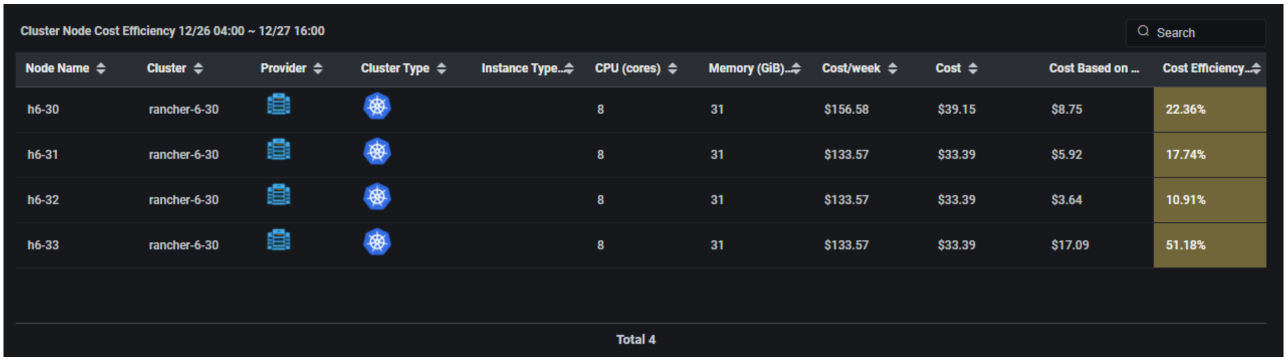
The *Cost Optimization* details page for a cluster shows the current configuration vs. the recommended number of cluster nodes, CPU cores, memory, and specific CPU/memory of each instance for the specified time frame (day/week/month).

Average actual CPU and memory usage is displayed along with the cost savings if the recommendations are followed.

Current CPU and memory capacity, recommended CPU and memory capacity, along with observed and predicted CPU and memory usage are also shown for Kubernetes clusters and AWS CloudWatch VM clusters configured with AWS ASG.

Cluster Node Cost Optimization

Cluster Node Cost Efficiency Table



Cluster Node Cost Efficiency 12/26 04:00 ~ 12/27 16:00

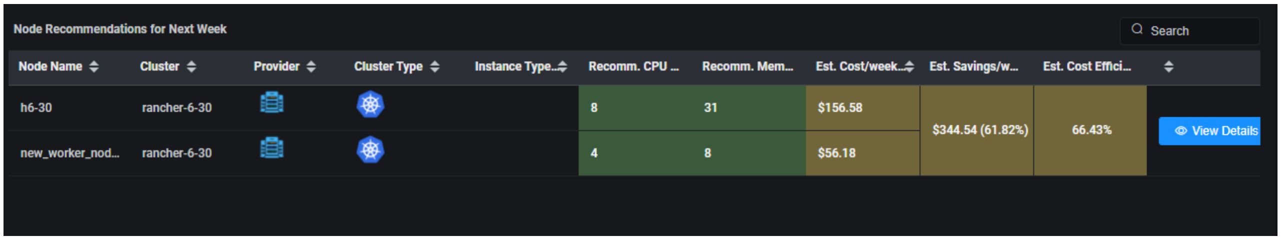
Search

Node Name	Cluster	Provider	Cluster Type	Instance Type...	CPU (cores)	Memory (GiB)...	Cost/week	Cost	Cost Based on ...	Cost Efficiency...
h6-30	rancher-6-30				8	31	$156.58	$39.15	$8.75	22.36%
h6-31	rancher-6-30				8	31	$133.57	$33.39	$5.92	17.74%
h6-32	rancher-6-30				8	31	$133.57	$33.39	$3.64	10.91%
h6-33	rancher-6-30				8	31	$133.57	$33.39	$17.09	51.18%

Total 4

Cluster Node Cost Efficiency information includes node name, cluster name, provider (AWS, Azure, Google, or on-premises), cluster type (Kubernetes, VM), instance type used at your cloud provider, CPU cores and memory capacity allocated for the node during the specified time frame, cost for each node and cost based on actual usage for the specified time frame, and the cost efficiency percentage (the cost from actual node usage divided by the allocated cost) for the specified time frame.

Cluster Node Recommendations Table



Node Recommendations for Next Week

Search

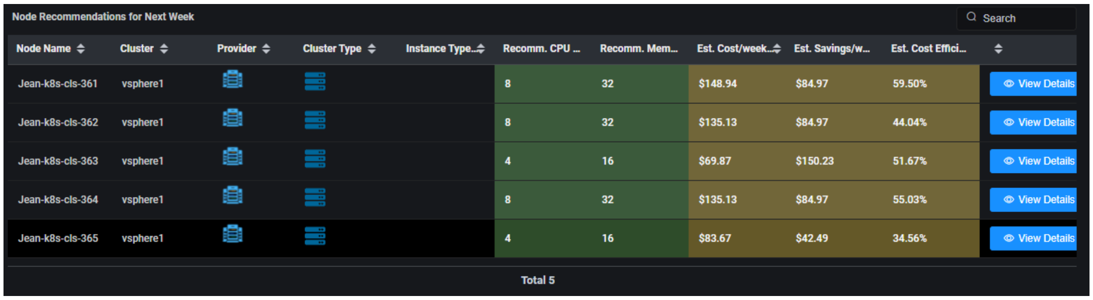
Node Name	Cluster	Provider	Cluster Type	Instance Type...	Recomm. CPU ...	Recomm. Mem...	Est. Cost/week...	Est. Savings/w...	Est. Cost Effici...	
h6-30	rancher-6-30				8	31	$156.58	$344.54 (61.82%)	66.43%	View Details
new_worker_nod...	rancher-6-30				4	8	$56.18			

The *Cluster Node Recommendations* table displays cluster name, provider (AWS, Azure, Google, or on-premises), cluster type (Kubernetes, VM), instance type, recommended CPU cores and memory, as well as the estimated cost, savings, and cost efficiency percentage (the cost from predicted usage divided by the allocated cost) per day/week/month if the recommendations are followed. Note that since containers are deployed to different cluster nodes by Kubernetes, there is no individual recommendation for a cluster node. Instead, the cost optimization and recommendations are applied to the entire cluster. Click the *View Details* link to view the recommendations to see how these savings are calculated.

This table shows the changes you can make to save money. However, the information presented is based on the cluster type:

- Kubernetes – Since containers are deployed to different cluster nodes by Kubernetes, there is no individual recommendation of individual cluster node. Recommendations for the optimal number of nodes in Kubernetes clusters are based on the cluster, even if an individual node is selected for display. Recommendations for CPU and memory are per node. For example, the system may recommend fewer nodes for the cluster but more CPU and memory for each node.

- AWS CloudWatch VM clusters configured with AWS ASG – Similar to a Kubernetes cluster, recommendations for the optimal number of nodes are based on the cluster, even if an individual node is selected for display. Recommendations for CPU and memory are per node.
- AWS CloudWatch VM clusters with individual VMs or VMware VM cluster – Each VM node will have its own recommendations for optimal number of CPU cores and memory.



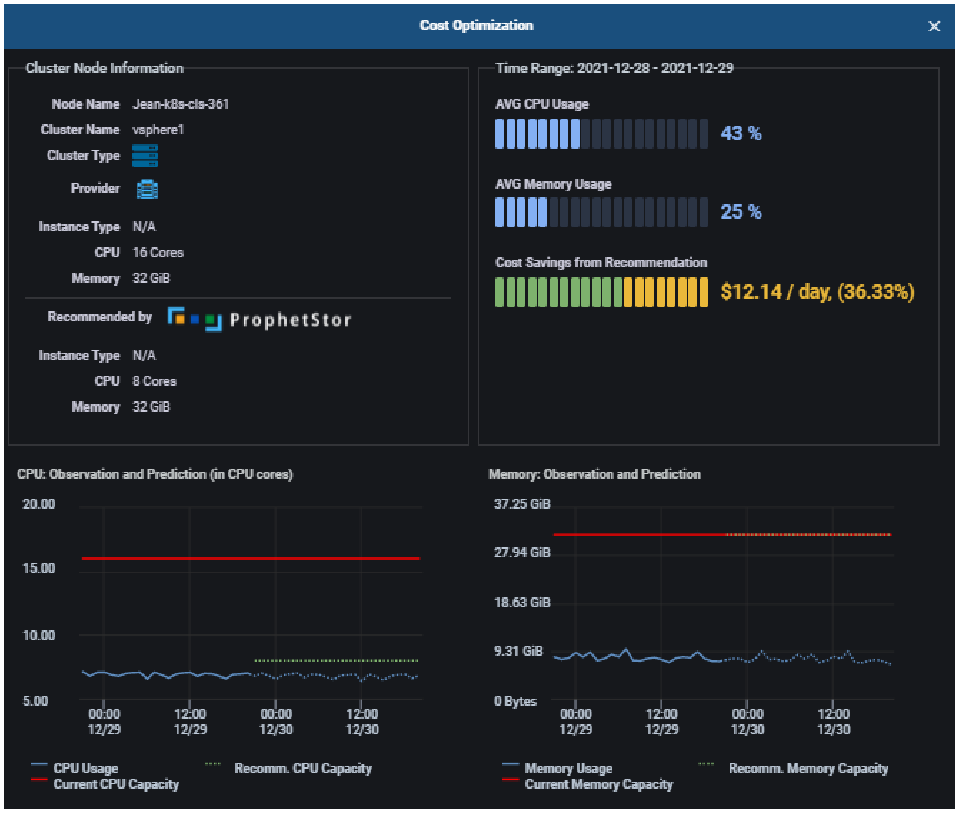
Node Recommendations for Next Week

Search

Node Name	Cluster	Provider	Cluster Type	Instance Type...	Recomm. CPU ...	Recomm. Mem...	Est. Cost/week...	Est. Savings/w...	Est. Cost Effici...	
Jean-k8s-cls-361	vsphere1				8	32	$148.94	$84.97	59.50%	View Details
Jean-k8s-cls-362	vsphere1				8	32	$135.13	$84.97	44.04%	View Details
Jean-k8s-cls-363	vsphere1				4	16	$69.87	$150.23	51.67%	View Details
Jean-k8s-cls-364	vsphere1				8	32	$135.13	$84.97	55.03%	View Details
Jean-k8s-cls-365	vsphere1				4	16	$83.67	$42.49	34.56%	View Details

Total 5

Cluster Node Cost Optimization Details



Cost Optimization

Cluster Node Information

Node Name Jean-k8s-cls-361
Cluster Name vsphere1
Cluster Type
Provider
Instance Type N/A
CPU 16 Cores
Memory 32 GiB

Recommended by ProphetStor

Instance Type N/A
CPU 8 Cores
Memory 32 GiB

Time Range: 2021-12-28 - 2021-12-29

AVG CPU Usage 43 %

AVG Memory Usage 25 %

Cost Savings from Recommendation $12.14 / day, (36.33%)

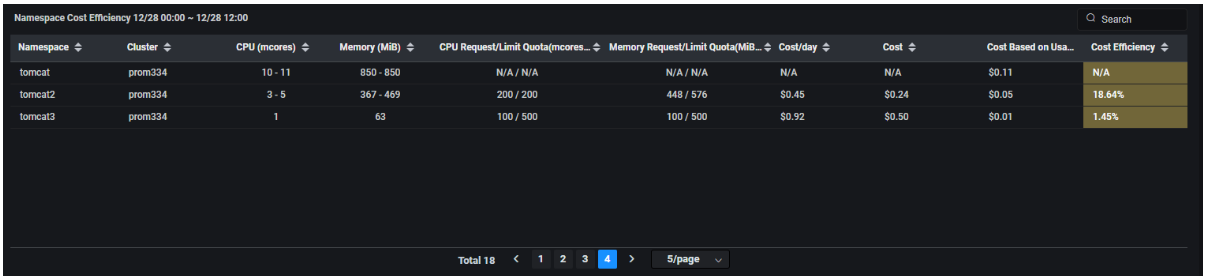
The *Cost Optimization* details page shows the current configuration vs. the recommended instance type, number of CPU cores and memory for the specified time frame (day/week/month) of a specific node.

For Kubernetes and AWS CloudWatch VM clusters configured with AWS Auto Scaling groups, recommendations are displayed for the entire cluster. For AWS CloudWatch VM clusters with individual VMs or VMware VM clusters, recommendations are for each VM node.

Cost from usage, cost from recommendations, and potential savings are also displayed. Click *Cumulative* to view information cumulatively through the end of the selected time period.

Namespace Cost Optimization

Namespace Cost Efficiency Table



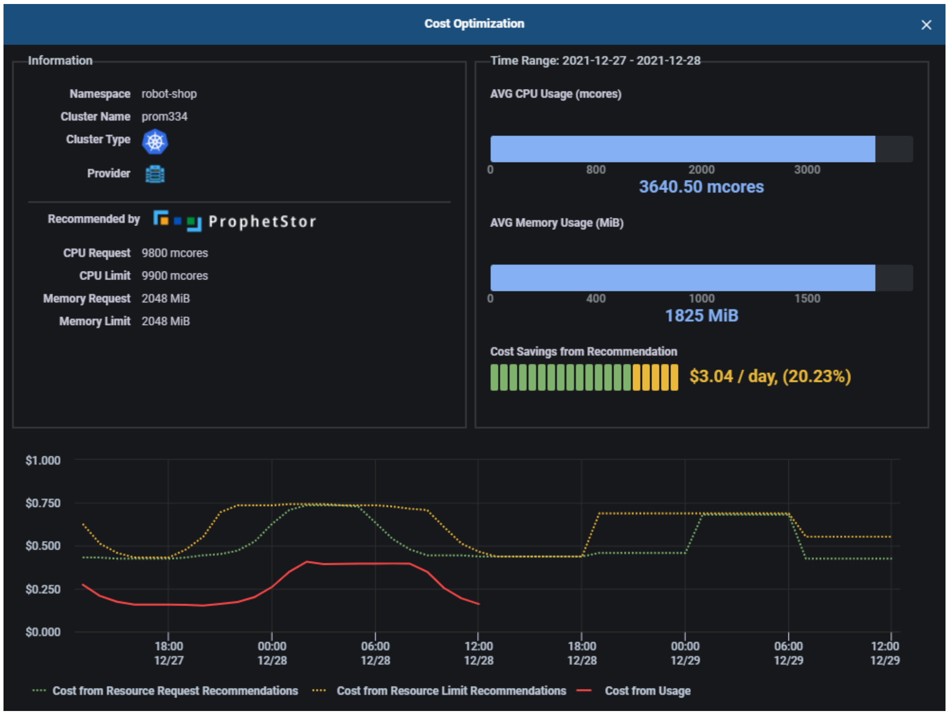
Namespace Cost Efficiency 12/28 00:00 ~ 12/28 12:00

Namespace	Cluster	CPU (mcores)	Memory (MiB)	CPU Request/Limit Quota(mcores...	Memory Request/Limit Quota(MiB...	Cost/day	Cost	Cost Based on Usa...	Cost Efficiency
tomcat	prom334	10 - 11	850 - 850	N/A / N/A	N/A / N/A	N/A	N/A	$0.11	N/A
tomcat2	prom334	3 - 5	367 - 469	200 / 200	448 / 576	$0.45	$0.24	$0.05	18.64%
tomcat3	prom334	1	63	100 / 500	100 / 500	$0.92	$0.50	$0.01	1.45%

Namespace Cost Efficiency information includes Kubernetes namespace, cluster, number of CPU millicores and amount of memory used during the specified time frame, CPU and memory request/limit quota, as well as the daily/weekly/monthly cost based on resource quota, current up-to-date cost based on quota, cost based on actual usage, and cost efficiency percentage (the cost from actual resource usage by this namespace divided by the namespace resource quota) per day/week/month for the namespace.

Namespace Recommendation information includes Kubernetes namespace, cluster, recommended CPU and memory request/limit quota, as well as the estimated cost, estimated savings, and estimated cost efficiency percentage based on recommended resource quota per day/week/month for the namespace. Click the *View Details* link to view the recommendations to see how these savings are calculated.

Namespace Optimization Details



Cost Optimization

Information

Namespace robot-shop
Cluster Name prom334
Cluster Type
Provider

Recommended by ProphetStor

CPU Request 9800 mcores
CPU Limit 9900 mcores
Memory Request 2048 MiB
Memory Limit 2048 MiB

Time Range: 2021-12-27 - 2021-12-28

AVG CPU Usage (mcores) 3640.50 mcores

AVG Memory Usage (MiB) 1825 MiB

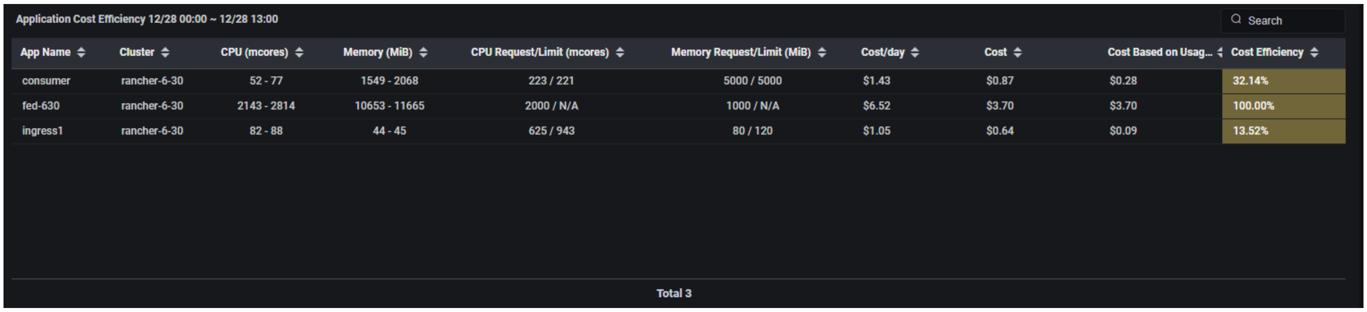
Cost Savings from Recommendation $3.04 / day, (20.23%)

The *Namespace Optimization* details page shows the current configuration vs. the recommended amount of CPU and memory requests/limits quota for the specified time frame (day/week/month).

Average actual CPU and memory usage is displayed along with the cost savings if the recommendations are followed. Cost from usage and cost from request/limit recommendations are also displayed. Note that the calculation of cost efficiency and estimated savings requires namespace resource quota information. If the information is not available, the current cost efficiency and estimated cost savings will not be available.

Application Cost Optimization

Application Cost Efficiency Table



Application Cost Efficiency 12/28 00:00 ~ 12/28 13:00

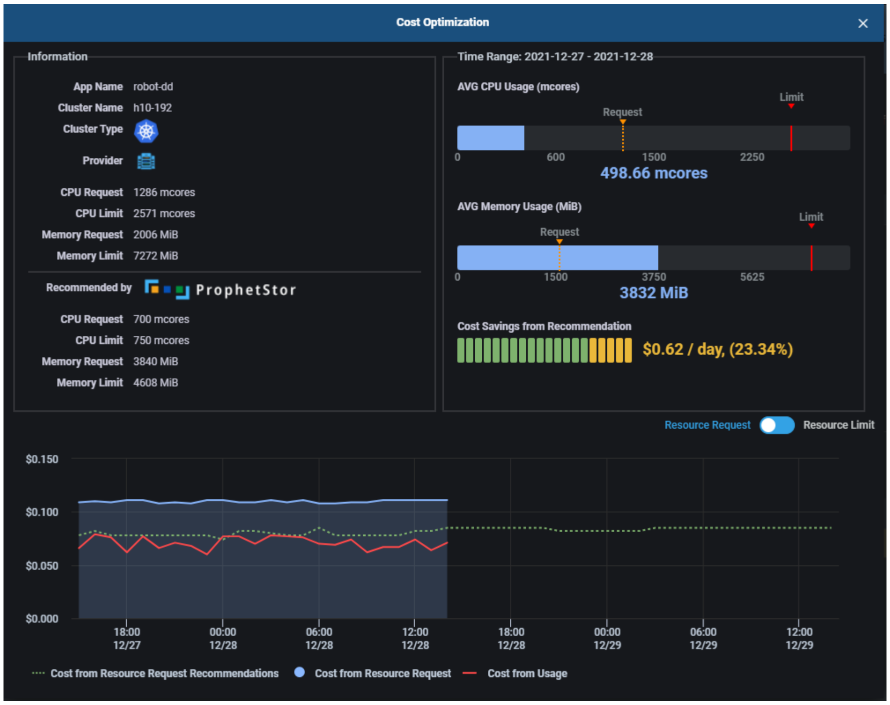
App Name	Cluster	CPU (mcores)	Memory (MiB)	CPU Request/Limit (mcores)	Memory Request/Limit (MiB)	Cost/day	Cost	Cost Based on Usag...	Cost Efficiency
consumer	rancher-6-30	52 - 77	1549 - 2068	223 / 221	5000 / 5000	$1.43	$0.87	$0.28	32.14%
fed-630	rancher-6-30	2143 - 2814	10653 - 11665	2000 / N/A	1000 / N/A	$6.52	$3.70	$3.70	100.00%
ingress1	rancher-6-30	82 - 88	44 - 45	625 / 943	80 / 120	$1.05	$0.64	$0.09	13.52%

Total 3

Cost Efficiency information includes Kubernetes application name, cluster, CPU and memory usage during the time period, CPU and memory requests/limits, as well as the daily/weekly/monthly cost, up-to-date cost, actual cost based on usage, and cost efficiency percentage (the cost from actual application usage divided by the allocated cost) per day/week/month.

Application Recommendations information includes Kubernetes application name, cluster, recommended CPU and memory requests/limits, as well as the estimated cost, savings, and cost efficiency percentage (the cost from actual application usage divided by the allocated cost) per day/week/month for the application if the recommendations are followed. Each application has its own recommendations. Click the *View Details* link to view the recommendations to see how these savings are calculated.

Application Optimization Details



Cost Optimization ×

Information

App Name robot-dd
Cluster Name h10-192
Cluster Type
Provider

CPU Request 1286 mcores
CPU Limit 2571 mcores
Memory Request 2006 MiB
Memory Limit 7272 MiB

Recommended by ProphetStor

CPU Request 700 mcores
CPU Limit 750 mcores
Memory Request 3840 MiB
Memory Limit 4608 MiB

Time Range: 2021-12-27 - 2021-12-28

AVG CPU Usage (mcores)
498.66 mcores

AVG Memory Usage (MiB)
3832 MiB

Cost Savings from Recommendation
$0.62 / day, (23.34%)

Resource Request / Resource Limit

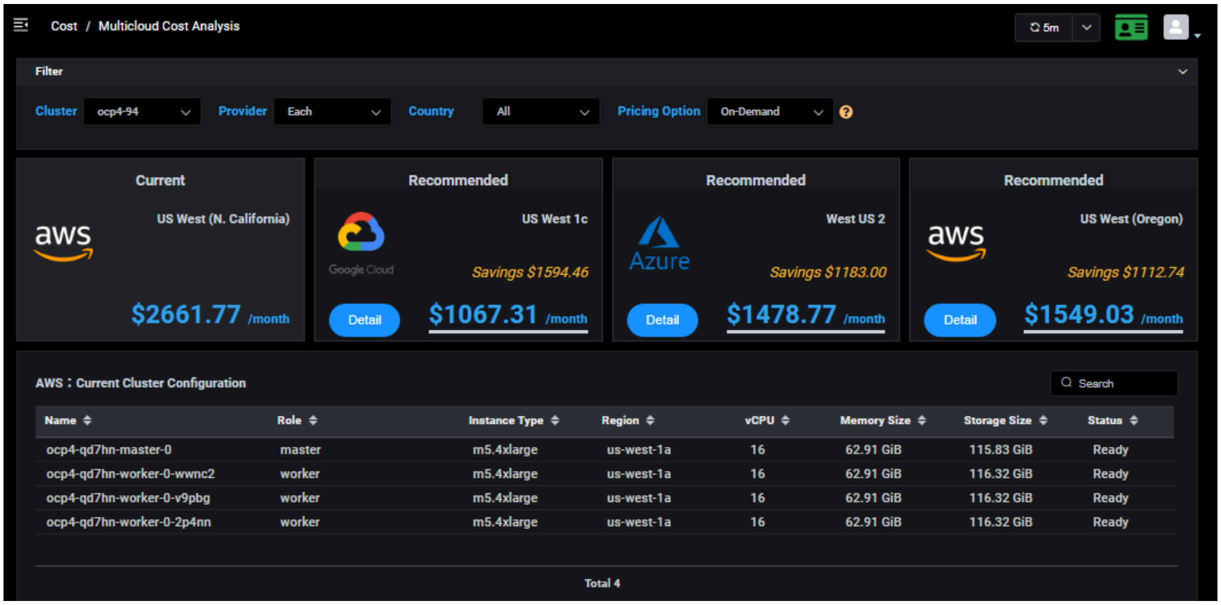
The *Application Optimization* details page shows the current configuration vs. the recommended amount of CPU and memory requests/limits for the specified time frame (day/week/month).

Average actual CPU and memory usage is displayed along with the cost savings if the recommendations are followed.

Cost from usage, cost from current resource requests/limits, and cost from recommendations are also displayed. You can switch between *Resource Request and Resource Limit* to see the savings comparison.

Cost – Multi-cloud Cost Analysis

The *Multi-cloud Cost Analysis* page calculates the amount you are spending with your current cloud provider (if applicable) or calculates what you are spending for your local cluster (based on your custom price book) and outlines your estimated cost for doing business with a variety of cloud providers based on fees charged by each provider and recommended changes to your current resource configuration. Federator.ai monitors and analyzes the overall CPU/memory usage of a cluster and predicts the usage for the next 30 days. Based on the past usage and the forecast, Federator.ai recommends the right instance types of cluster nodes for the workload with the lowest cost from each cloud provider.



Cluster: ocp4-94 | Provider: Each | Country: All | Pricing Option: On-Demand

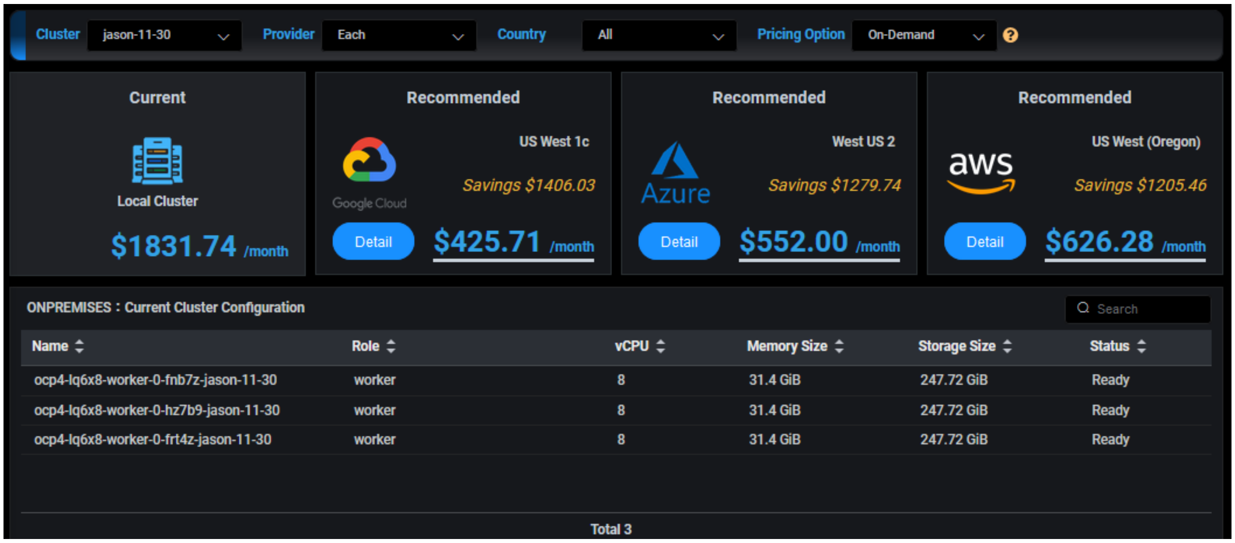
Current	Recommended	Recommended	Recommended
aws US West (N. California) $2661.77 /month	Google Cloud US West 1c Savings $1594.46 $1067.31 /month	Azure West US 2 Savings $1183.00 $1478.77 /month	aws US West (Oregon) Savings $1112.74 $1549.03 /month

AWS : Current Cluster Configuration

Name	Role	Instance Type	Region	vCPU	Memory Size	Storage Size	Status
ocp4-qd7hn-master-0	master	m5.4xlarge	us-west-1a	16	62.91 GiB	115.83 GiB	Ready
ocp4-qd7hn-worker-0-wwnc2	worker	m5.4xlarge	us-west-1a	16	62.91 GiB	116.32 GiB	Ready
ocp4-qd7hn-worker-0-v9pbg	worker	m5.4xlarge	us-west-1a	16	62.91 GiB	116.32 GiB	Ready
ocp4-qd7hn-worker-0-2p4nn	worker	m5.4xlarge	us-west-1a	16	62.91 GiB	116.32 GiB	Ready

Total 4

Spending with current cloud provider



Cluster: jason-11-30 | Provider: Each | Country: All | Pricing Option: On-Demand

Current	Recommended	Recommended	Recommended
Local Cluster $1831.74 /month	Google Cloud US West 1c Savings $1406.03 $425.71 /month	Azure West US 2 Savings $1279.74 $552.00 /month	aws US West (Oregon) Savings $1205.46 $626.28 /month

ONPREMISES : Current Cluster Configuration

Name	Role	vCPU	Memory Size	Storage Size	Status
ocp4-lq6x8-worker-0-fnb7z-jason-11-30	worker	8	31.4 GiB	247.72 GiB	Ready
ocp4-lq6x8-worker-0-hz7b9-jason-11-30	worker	8	31.4 GiB	247.72 GiB	Ready
ocp4-lq6x8-worker-0-frt4z-jason-11-30	worker	8	31.4 GiB	247.72 GiB	Ready

Total 3

Spending for local cluster

You can filter the data that appears on the whole page by selecting:

- Cluster
- Provider – Display data for the:
    - Three lowest costs among all providers.
    - Lowest cost from each provider.
    - Three lowest costs from a selected provider.
- Country – All or a specific country.
- Pricing option - The following pricing options are available:
    - Reserved - Stationary workloads will be served with Reserved instances and temporary increased workloads will be served with On-Demand instances.
    - On-Demand - All workloads will be served with On-Demand instances.
    - Spot – (Kubernetes only) Evictable workloads will be served with Spot instances while the other workloads will be served with On-Demand instances.
    - Spot + Reserved - (Kubernetes only) Evictable workloads will be served with Spot instances while the other workloads will be served with Reserved and On-Demand instances.

The *Current* box displays what you are currently paying (locally or to your cloud provider) with your existing resource configuration. The current configuration for each cluster node, including Kubernetes role (master, worker), instance types being used at your cloud provider, cloud provider region, number of CPUs, memory size, and storage size, is displayed in the table below.

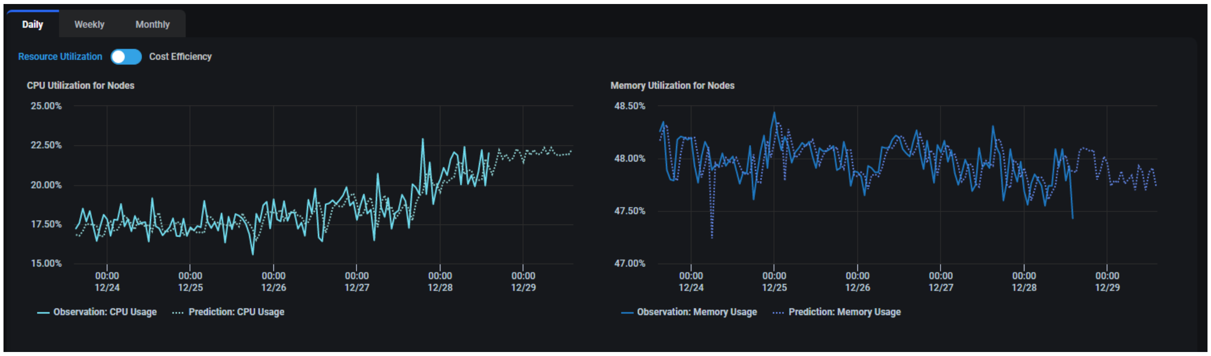
The *Recommended* boxes display potential savings with different cloud providers or different regions for your current provider. Click the *Detail* button to see how these savings are calculated. The savings typically come from lower provider fees, as well as from reducing idle resources and using more cost-effective instance types with the provider. The information continually refreshes itself as new data becomes available.

While it is difficult to move between cloud providers, this information can help you reduce costs by changing instance types, reducing idle resources, and selecting a different region from your current provider.

Even if you are not currently using a cloud provider, this information can show you what your options are if you move to the cloud.

Resource Utilization and Cost Efficiency Charts

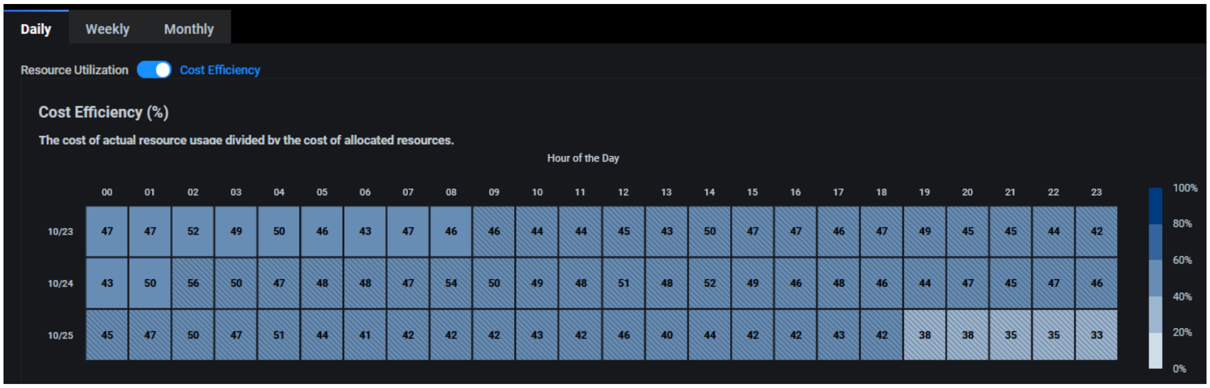
The *Resource Utilization* charts display CPU and memory utilization information for all nodes/VMs in the selected cluster based on your daily, weekly, or monthly workload.



The solid lines represent the observed actual usage while the dotted lines show the historical and future predicted usage. The predicted usage is used in the analysis for cluster configuration recommendations.

If there are evictable Kubernetes applications in this cluster, you will also see their actual and predicted usage.

The *Cost Efficiency* chart displays the actual cost of resource usage compared to the cost of allocated resources for the selected cluster based on your daily, weekly, or monthly workload.



Cost Efficiency (%)

The cost of actual resource usaae divided bv the cost of allocated resources.

Hour of the Day

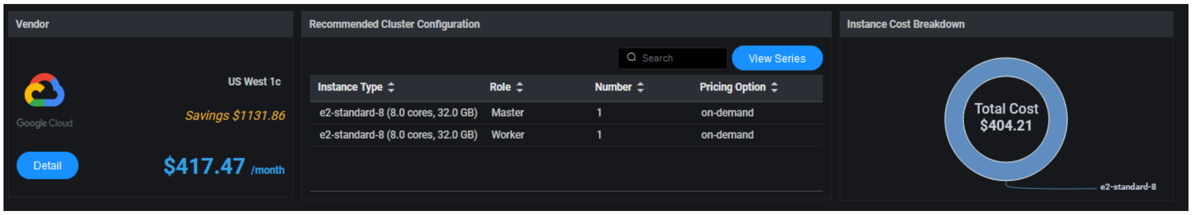
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
10/23	47	47	52	49	50	46	43	47	46	46	44	44	45	43	50	47	47	46	47	49	45	45	44	42
10/24	43	50	56	50	47	48	48	47	54	50	49	48	51	48	52	49	46	48	46	44	47	45	47	46
10/25	45	47	50	47	51	44	41	42	42	42	43	42	46	40	44	42	42	43	42	38	38	35	35	33

The color gradient illustrates the percentage range. Boxes with diagonal lines represent future predicted cost efficiency.

By comparing actual usage costs to the cost of allocated resources, you can easily see where you are over-provisioned, enabling you to adjust your resources for better cost efficiency. For example, if you see that the percentage is consistently lower than expected (indicating low cost efficiency), you may want to reduce size of the cluster or allocate less CPU/memory.

Recommended Cluster Configuration

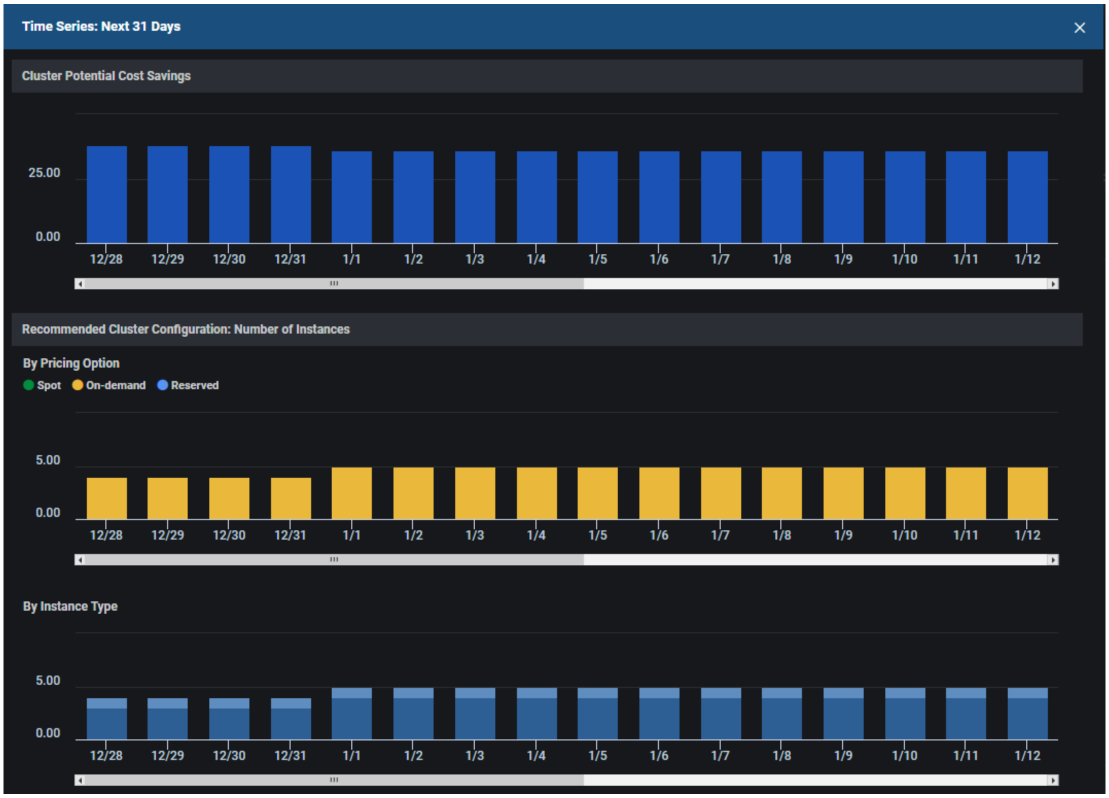
The recommended configuration with each provider is displayed based on your daily, weekly, or monthly workload prediction (e.g., if *Daily* is selected for the *Resource Utilization* charts, the cost and saving are calculated based on the workload prediction for the next 24 hours). This includes the recommended instance types (scroll down in the table to see all instance types), as well as the number of Kubernetes master and worker servers for the selected pricing option.



Instance Type	Role	Number	Pricing Option
e2-standard-8 (8.0 cores, 32.0 GB)	Master	1	on-demand
e2-standard-8 (8.0 cores, 32.0 GB)	Worker	1	on-demand

The *Instance Cost Breakdown* chart shows the cost per instance type for the specified time frame.

Click the *View Series* button to view additional, more fine-grained, savings information. For daily predictions, data is displayed every hour for the next 24 hours. For weekly predictions, data is displayed every six hours for the next seven days. For monthly predictions, data is displayed every day for the next 31 days.


Time Series: Next 31 Days
×
Cluster Potential Cost Savings
25.00
0.00
12/28
12/29
12/30
12/31
1/1
1/2
1/3
1/4
1/5
1/6
1/7
1/8
1/9
1/10
1/11
1/12
Recommended Cluster Configuration: Number of Instances
By Pricing Option
● Spot ● On-demand ● Reserved
5.00
0.00
By Instance Type
5.00
0.00

The *Cluster Potential Cost Savings* chart displays how much you are predicted to save by following the recommended configuration.

The *Recommended Cluster Configuration: Number of Instances* charts display the recommended number of on-demand, reserved, or spot instances (for Kubernetes), depending on the pricing option selected.

The recommendations for instance type include the number and type for future workloads at specific times and can help you determine how many resources to reserve. Place your cursor over one of the bars to see the recommended number of instances.

In the case of a VM cluster, the recommended cluster configuration will display the recommended instance type for each individual VM.

Related topics:

VM Cost Analysis
Application Cost Analysis
Cost Allocation Kubernetes Namespaces
Common Administration Portal Functions
Terminology
Search/Sort Information in Tables
Show/Hide Information in Charts
Zoom In/Out of Charts
Price Books

Configuration - Clusters

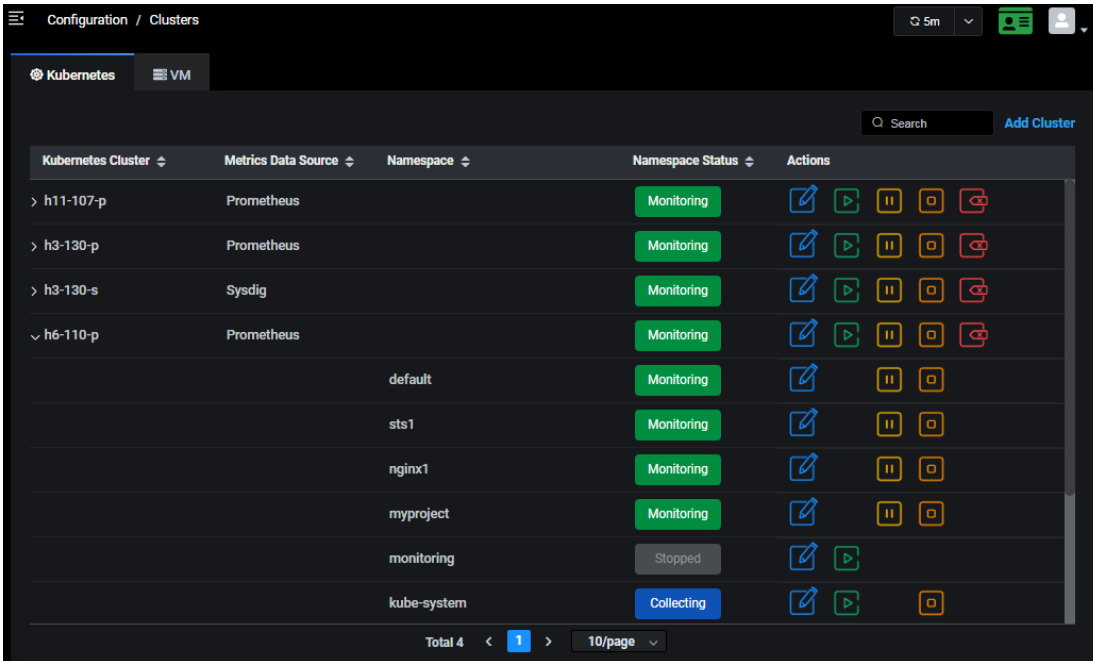
The *Clusters* page displays all Kubernetes or VM clusters being monitored by Federator.ai. You can add, manage, and remove clusters from this page.

Kubernetes Clusters

Select the *Kubernetes* tab to see the list of monitored Kubernetes clusters. Expand a cluster to see the namespaces for a cluster.

You can also see namespace status for the cluster and individual namespaces. The status will be *Monitoring* when the system is collecting metrics and providing workload predictions. The status will be *Collecting* when all the namespaces in the cluster are in collecting state. When a namespace is in collecting state, the system will still collect metrics, but prediction tasks are paused. When a namespace is added to an existing cluster, it will collect data but will not be automatically monitored until it is manually set for monitoring.

A status of *Stopped* means there is no collecting of metrics and no predictions. A cluster is *Stopped* if all of its namespaces are stopped.



Kubernetes Cluster	Metrics Data Source	Namespace	Namespace Status	Actions
› h11-107-p	Prometheus		Monitoring	✎ ▷ ⏸ ◻ ⊗
› h3-130-p	Prometheus		Monitoring	✎ ▷ ⏸ ◻ ⊗
› h3-130-s	Sysdig		Monitoring	✎ ▷ ⏸ ◻ ⊗
⌄ h6-110-p	Prometheus		Monitoring	✎ ▷ ⏸ ◻ ⊗
		default	Monitoring	✎ ⏸ ◻
		sts1	Monitoring	✎ ⏸ ◻
		nginx1	Monitoring	✎ ⏸ ◻
		myproject	Monitoring	✎ ⏸ ◻
		monitoring	Stopped	✎ ▷
		kube-system	Collecting	✎ ▷ ◻

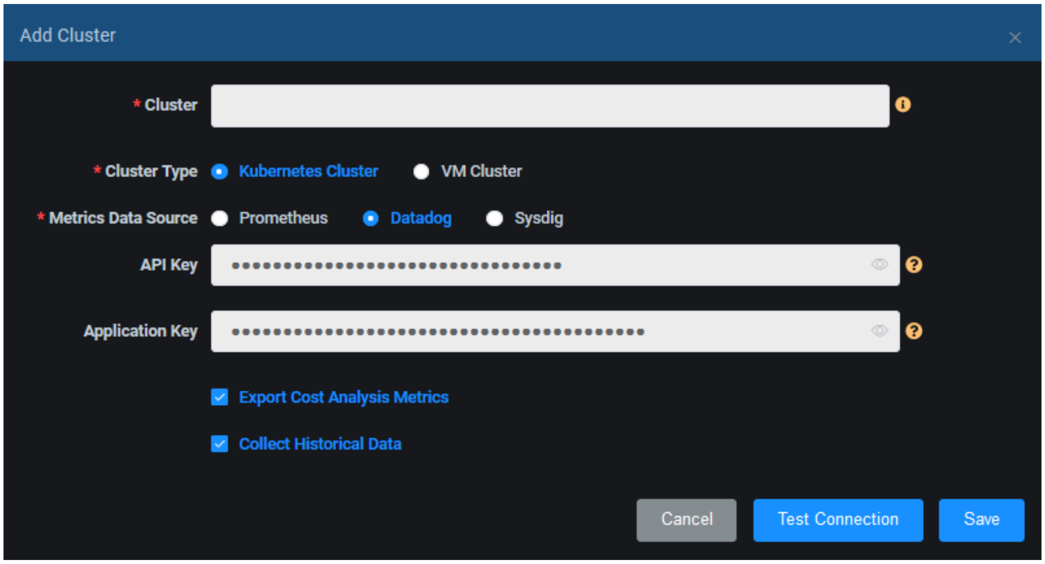
In addition to adding a cluster, you can perform the following functions for existing clusters and namespaces:

Icon	Function
✎	Edit settings for clusters and namespaces.
▷	Start monitoring and predictions for all namespaces or a specific namespace.
⏸	Pause monitoring and predictions for all namespaces or a specific namespace.

⏹	Stop collecting metrics and making predictions for all namespaces or a specific namespace.
⌫	Remove a cluster.

Add a Kubernetes Cluster

1. On the *Configuration / Clusters* page, click *Add Cluster*.



Add Cluster

* Cluster

* Cluster Type ● Kubernetes Cluster ○ VM Cluster

* Metrics Data Source ○ Prometheus ● Datadog ○ Sysdig

API Key ••••••••••••••••••••••••••••••••

Application Key ••

☑ Export Cost Analysis Metrics

☑ Collect Historical Data

Cluster – Specify the name of a cluster to be managed. There is a maximum of 253 lowercase characters,"-", or "." allowed. The name must start and end with an alphanumeric character.

Cluster Type – By default, *Kubernetes Cluster* is selected for you.

Metrics Data Source – Select the source of metrics for this cluster.

Prometheus Federation – If *Prometheus* is your data source, specify if you are using Federation, which is a group of Prometheus servers that send metrics to a centralized Prometheus server. You will need to specify the target label of the centralized Prometheus server. The format is: `<label-name>:<label-value>` (e.g., clusterID:host-1).

API Key/Application Key – For Datadog, the API key and application key are required for authentication. By default, they are set as Datadog API key and application key from the *Metrics Data Source* tab under *Configuration / System Settings*. Each cluster can use a different API key and application key.

URL/Token – For Sysdig, a URL and token are required for authentication. Click the information icon to see supported URLs. For the Prometheus open-source monitoring system, the URL is required but the token is optional. By default, they are set as Sysdig URL/token or Prometheus URL from the *Metrics Data Source* tab under *Configuration / System Settings*. Each cluster can use different values.

- Authenticate Prometheus by using basic authentication with a username and password.

  Use the following command to generate the token:

  ```
  # echo -n “<username>:<password>” | base64
  ```

  Refer to the following for information about securing the Prometheus API using basic authentication (Basic Auth): https://prometheus.io/docs/guides/basic-auth/

- Authenticate Prometheus by using a service account token in OpenShift:

  Use the following commands to get the service account name for Prometheus:

  ```
  # oc get prometheus -n openshift-monitoring
  NAME AGE
  k8s 169d
  # oc get prometheus -n openshift-monitoring k8s -oyaml | grep serviceAccount
  serviceAccountName: prometheus-k8s
  ```

  Use the following command to get the token for the Prometheus service account:

  ```
   # oc serviceaccounts get-token prometheus-k8s -n openshift-monitoring
  ```

Export Cost Analysis Metrics – Specify if you want cost analysis metrics to be automatically exported to Datadog, so that the information can appear in Datadog’s user interface.

Collect Historical Data – Specify if you want the system to collect up to three months' worth of historical data for existing nodes and namespaces in this cluster. This will enable weekly and monthly predictions, recommendations, and cost analysis for newly added clusters without waiting to collect weeks’ or months’ worth of data. If less than three months’ worth of data exists, the system will collect the maximum data that is available. Typically, collection of historical data will complete in about 2-3 hours. If you need to pause collection, you can edit cluster settings.

Note that Federator.ai uses the APIs provided by your metrics data source (Datadog, Sysdig) to access historical data. The data source imposes limits on how many calls can be made to their service per hour. If the rate limit is too low, the queries for historical data may exceed the limit and the API will return an error. You will need to contact your metrics data service provider to raise your API rate limit.

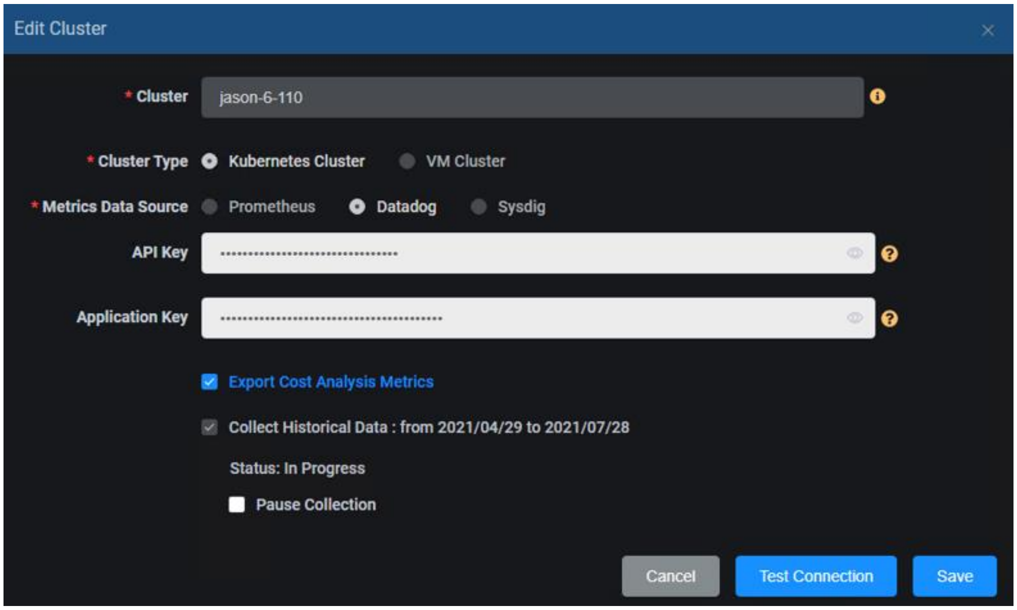
2. Click *Test Connection* to confirm that all information is correct.

3. Click *Save* when the system can connect to the cluster.

Manage Kubernetes Clusters

You can do the following from the *Configuration / Clusters* page:

- Edit cluster settings – Click the *Edit Cluster* icon to manage export of cost analysis metrics (Datadog), configure Prometheus Federation (Prometheus), or test the connection to the cluster. For historical data collection, you can:
  - Start the collection of up to three months' worth of historical data (from the current time) for existing nodes and namespaces in the cluster, if it was not enabled when the cluster was added.
  - See the status of data collection, including the time period collected.
  - Pause/resume historical data collection that is in progress.



Edit Cluster

* Cluster: jason-6-110

* Cluster Type: ● Kubernetes Cluster ○ VM Cluster

* Metrics Data Source: ○ Prometheus ● Datadog ○ Sysdig

API Key: ••••••••••••••••••••••••••••••••

Application Key: ••

☑ Export Cost Analysis Metrics

☑ Collect Historical Data : from 2021/04/29 to 2021/07/28

Status: In Progress

☐ Pause Collection

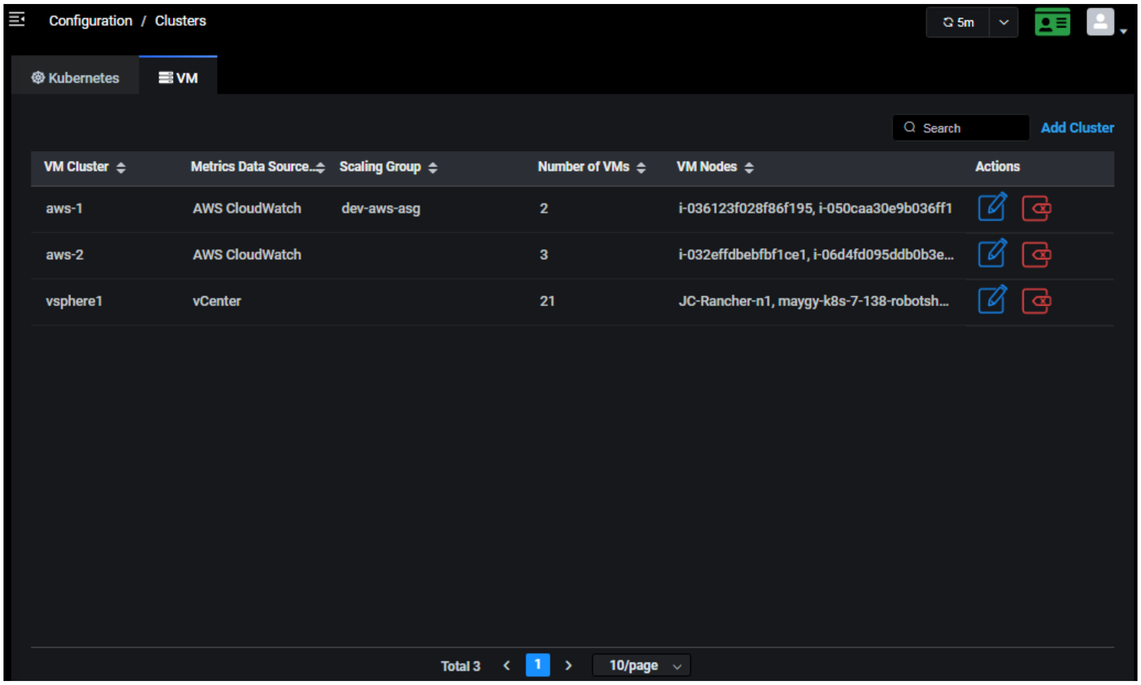
- Edit namespace settings – Change monitoring status and configure auto provisioning for the namespace. To do this, click the *Edit Namespace* icon.
- Start monitoring and prediction for all namespaces in the cluster or a specific namespace. To do this, click the *Start Monitoring and Predictions* icon.
- Pause monitoring and prediction for all namespaces or a specific namespace. To do this, click the *Pause Monitoring and Predictions* icon for a cluster or for a namespace.
- Stop collecting metrics and making predictions for all namespaces or a specific namespace. To do this, click the *Stop Collecting Metrics and Predictions* icon for a cluster or for a namespace.
- Remove a cluster that does not have any applications configured. To do this, click the *Remove Cluster* icon.

Related topic:

Terminology
Search/Sort Information in Tables
Configure Applications

VM Clusters

Select the *VM* tab to see the list of monitored VM clusters and the number and names of the virtual machines (VMs) in each cluster.



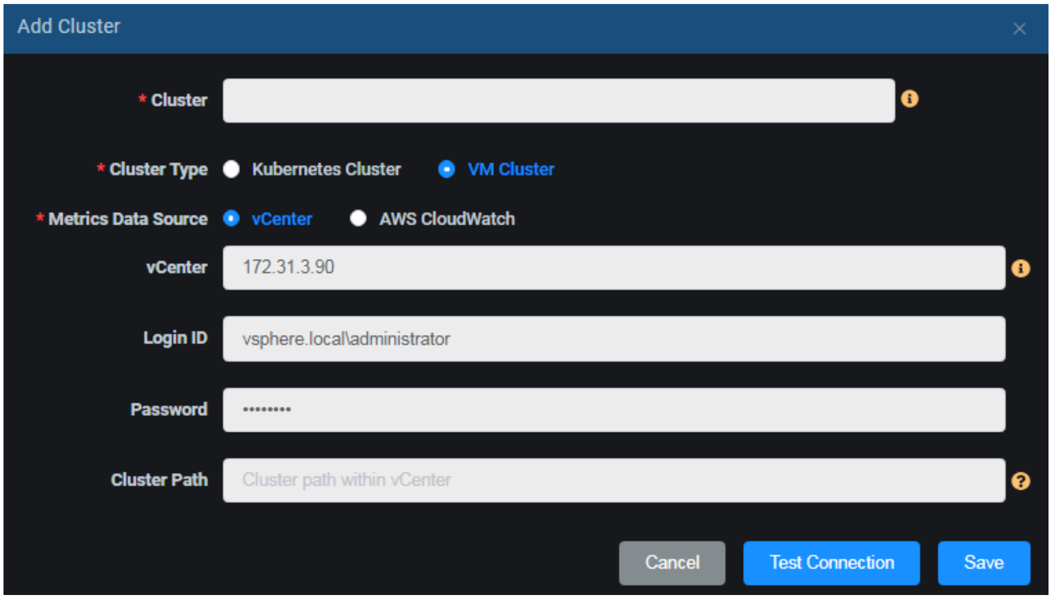
VM Cluster	Metrics Data Source...	Scaling Group	Number of VMs	VM Nodes	Actions
aws-1	AWS CloudWatch	dev-aws-asg	2	i-036123f028f86f195, i-050caa30e9b036ff1	✎ ⌫
aws-2	AWS CloudWatch		3	i-032effdbebfbf1ce1, i-06d4fd095ddb0b3e...	✎ ⌫
vsphere1	vCenter		21	JC-Rancher-n1, maygy-k8s-7-138-robotsh...	✎ ⌫

In addition to adding a cluster, you can perform the following functions for existing clusters:

Icon	Function
✎	Add/remove the cluster nodes to monitor.
⌫	Remove a cluster.

Add a VM Cluster

1. On the *Configuration / Clusters* page, click *Add Cluster*.



Add Cluster

* Cluster

* Cluster Type ○ Kubernetes Cluster ● VM Cluster

* Metrics Data Source ● vCenter ○ AWS CloudWatch

vCenter: 172.31.3.90

Login ID: vsphere.local\administrator

Password: ••••••••

Cluster Path: Cluster path within vCenter

Cluster – Specify the name of the VM cluster to be managed. There is a maximum of 253 lowercase characters,"-", or "." allowed. The name must start and end with an alphanumeric character.

Cluster Type – By default, *VM Cluster* is selected for you.

Metrics Data Source – Select the source of metrics for this cluster:

vCenter

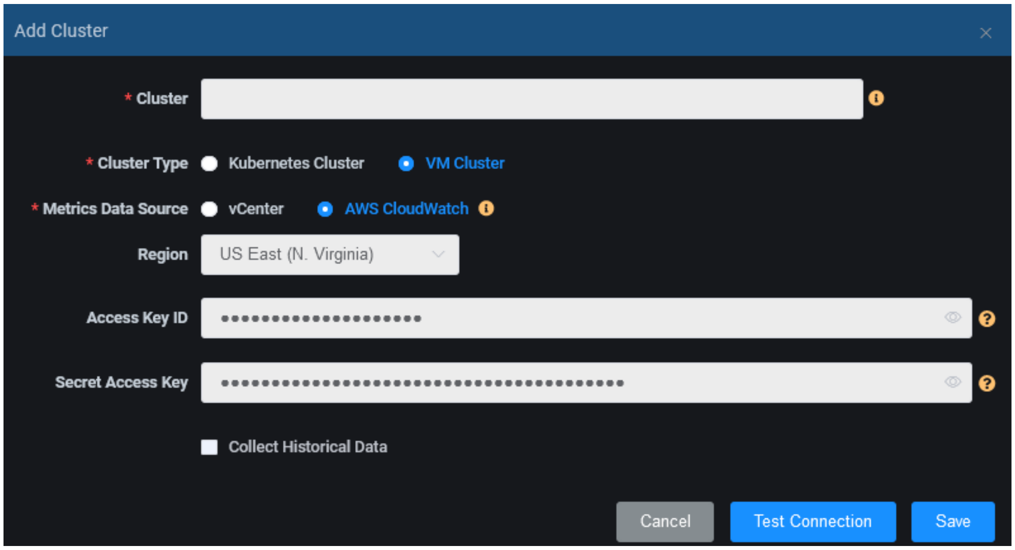
vCenter – Specify the vCenter IP address. You can have multiple vCenters in your system.

Login ID and Password – Specify the login credentials.

Cluster Path – Specify the path to the cluster, within vCenter. If needed, you can click on the link to the vCenter website, which is included in the popup help text.

AWS CloudWatch

Note: The CloudWatch agent must be installed on the EC2 node in order to use this data source.



Add Cluster

* Cluster

* Cluster Type ○ Kubernetes Cluster ● VM Cluster

* Metrics Data Source ○ vCenter ● AWS CloudWatch

Region: US East (N. Virginia)

Access Key ID: ••••••••••••••••••••

Secret Access Key: ••

□ Collect Historical Data

Region - Specify the region of Amazon AWS EC2 service.

Access Key ID - Specify the access key ID of an IAM user (16 to 128 bytes).

Secret Access Key - Specify the secret access key of the key ID that is used for access.

Collect Historical Data – Specify if you want the system to collect up to three months' worth of historical data for VMs in this cluster. This will enable weekly and monthly predictions, recommendations, and cost analysis for newly added clusters without waiting to collect weeks’ or months’ worth of data. If less than three months’ worth of data exists, the system will collect the maximum data that is available. Typically, collection of historical data will complete in about 2-3 hours. If you need to pause collection, you can edit cluster settings.

2. Click *Test Connection* to confirm that all information is correct.

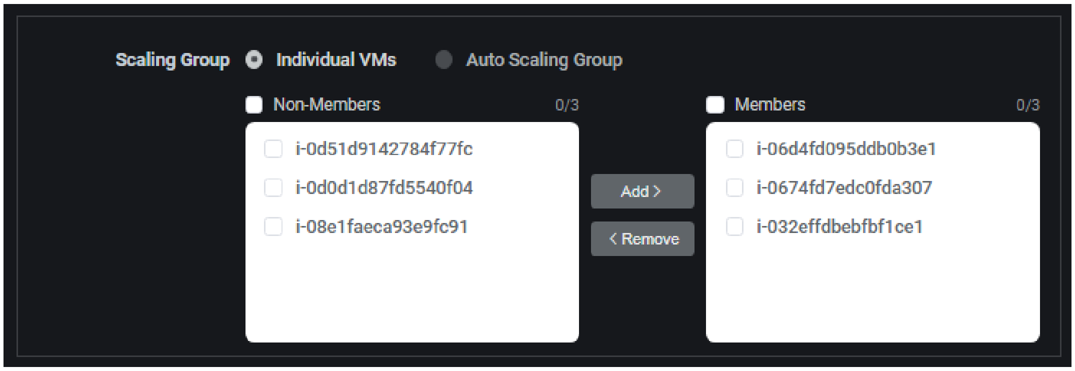
3. Click *Save* when the system can connect to the cluster.

   The system will discover the EC2 VMs and auto scaling groups in this region, which may take a few minutes to complete. Use the *Edit Cluster* function to select which nodes that you want to monitor.

Manage VM Clusters

You can do the following from the *Configuration / Clusters* page:

- Add/remove nodes or auto scaling groups to monitor. To do this, click the *Edit Cluster* icon. Select which members of a VM cluster to monitor.



Scaling Group ● Individual VMs ○ Auto Scaling Group

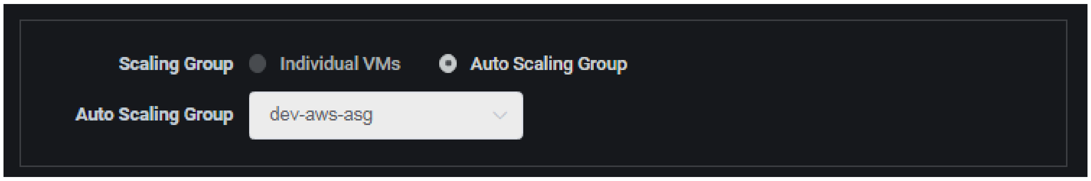
□ Non-Members 0/3

□ i-0d51d9142784f77fc
□ i-0d0d1d87fd5540f04
□ i-08e1faeca93e9fc91

□ Members 0/3

□ i-06d4fd095ddb0b3e1
□ i-0674fd7edc0fda307
□ i-032effdbebfbf1ce1

If you have Auto Scaling groups configured in AWS, select a group to monitor.



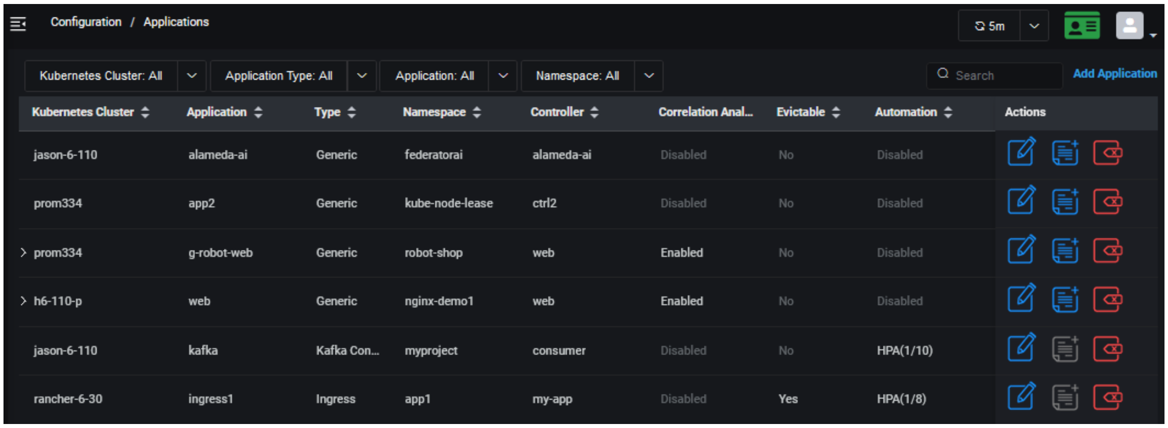
Scaling Group ○ Individual VMs ● Auto Scaling Group

Auto Scaling Group: dev-aws-asg

- Remove a cluster. To do this, click the *Remove Cluster* icon.

Configuration – Applications

The *Applications* page displays all Kubernetes applications being monitored by Federator.ai and allows you to manage them.



Kubernetes Cluster	Application	Type	Namespace	Controller	Correlation Anal...	Evictable	Automation	Actions
jason-6-110	alameda-ai	Generic	federatorai	alameda-ai	Disabled	No	Disabled	✎ 📄 ⌫
prom334	app2	Generic	kube-node-lease	ctrl2	Disabled	No	Disabled	✎ 📄 ⌫
› prom334	g-robot-web	Generic	robot-shop	web	Enabled	No	Disabled	✎ 📄 ⌫
› h6-110-p	web	Generic	nginx-demo1	web	Enabled	No	Disabled	✎ 📄 ⌫
jason-6-110	kafka	Kafka Con...	myproject	consumer	Disabled	No	HPA(1/10)	✎ 📄 ⌫
rancher-6-30	ingress1	Ingress	app1	my-app	Disabled	Yes	HPA(1/8)	✎ 📄 ⌫

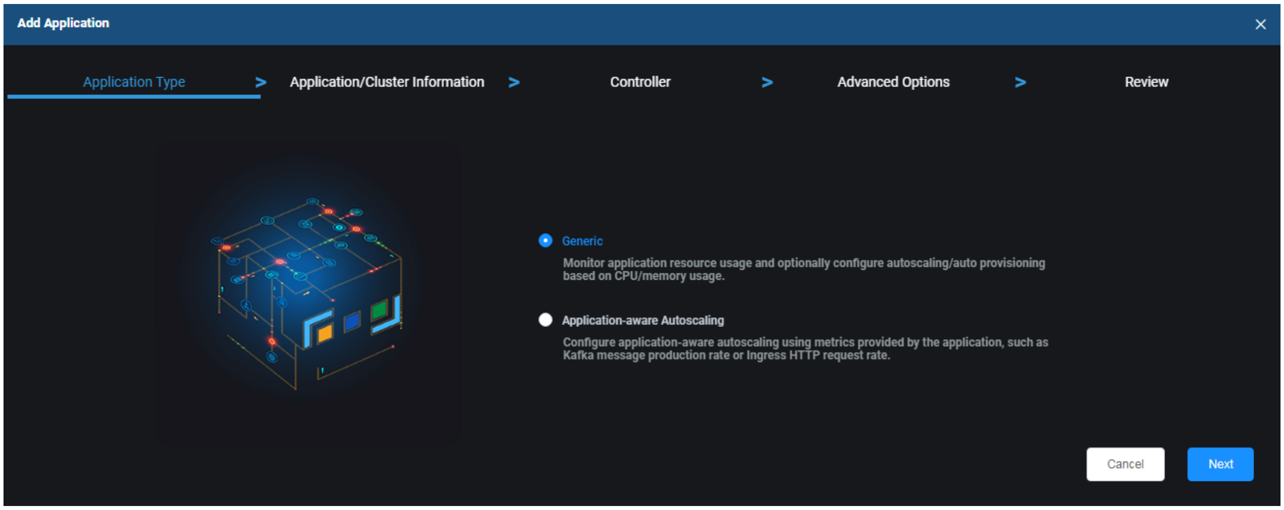
In addition to adding an application, you can perform the following functions for existing applications:

Icon	Function
✎	Edit an application.
📄	View the resource provisioning script.
⌫	Remove an application.

Add an Application

Applications are configured via a wizard.

1. On the *Configuration / Applications* page, click *Add Application* and specify the type of application you are adding.



● Generic

Monitor application resource usage and optionally configure autoscaling/auto provisioning based on CPU/memory usage.

○ Application-aware Autoscaling

Configure application-aware autoscaling using metrics provided by the application, such as Kafka message production rate or Ingress HTTP request rate.

Depending upon the type of application you are adding, the system will be monitored and can optionally provide autoscaling.

The traditional method of autoscaling, based on CPU/memory usage, is not enough to achieve desired performance goals for certain applications. For example, Kafka production and consumption of messages offer a better indicator of workload and performance. Using message production rate, Federator.ai provides application-aware autoscaling of Kafka consumer pods to fit the workload and optimize performance.

As another example, Ingress services use key performance goals, including the ability of Ingress to forward requests to upstream services with minimal response time and errors, as an indicator of workload and performance. Using HTTP request rate, Federator.ai provides application-aware autoscaling of Ingress services to fit the workload and optimize performance.

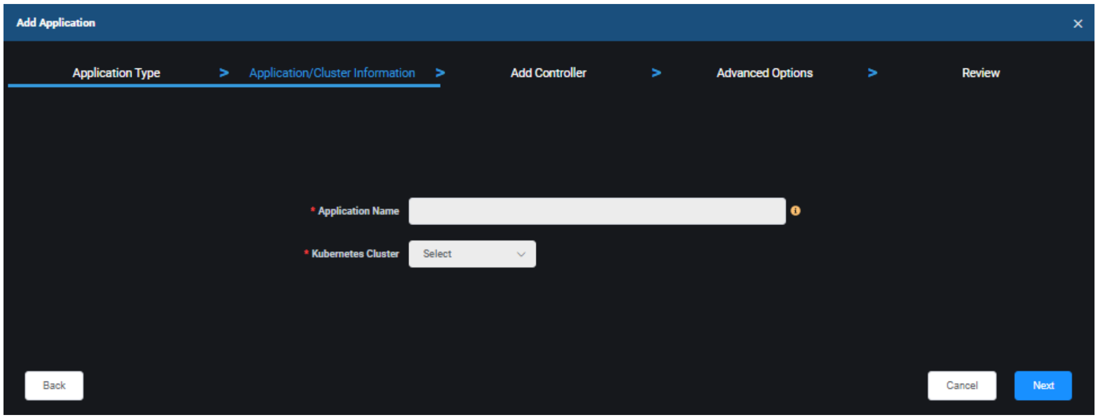
For other Kubernetes applications, CPU and memory usage is monitored and Federator.ai autoscales the number of pods based on workload and proactive usage predictions.

For Kafka and Ingress applications, select *Application-aware Autoscaling*. For all other Kubernetes applications, select *Generic*.

2. Click *Next* to continue.

Generic Kubernetes Application

1. Specify an application name and Kubernetes cluster and then click *Next* to continue.



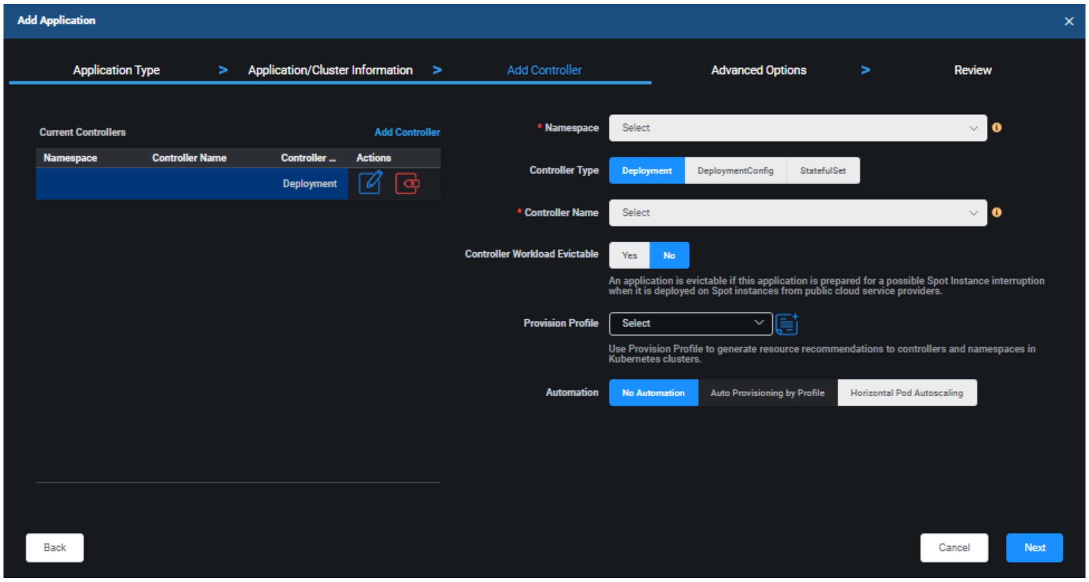
* Application Name

* Kubernetes Cluster: Select

Application Name - Name of the application you want to manage. The name must be a maximum of 253 characters, contain only lowercase alphanumeric characters, "-", or ".", and start and end with an alphanumeric character. Note, an application is not a native Kubernetes object. You will define the controllers that are part of your application later.

Kubernetes Cluster – Select an existing cluster where this application resides.

2. Configure controllers.



Current Controllers

Namespace	Controller Name	Controller ...	Actions
		Deployment	

* Namespace: Select

Controller Type: Deployment | DeploymentConfig | StatefulSet

* Controller Name: Select

Controller Workload Evictable: Yes | No

An application is evictable if this application is prepared for a possible Spot Instance interruption when it is deployed on Spot instances from public cloud service providers.

Provision Profile: Select

Use Provision Profile to generate resource recommendations to controllers and namespaces in Kubernetes clusters.

Automation: No Automation | Auto Provisioning by Profile | Horizontal Pod Autoscaling

Namespace – Select the Kubernetes namespace where the controller is deployed.

Controller Type – Supported controller types are *Deployment, StatefulSet,* and *DeploymentConfig* (OpenShift only).

Controller Name – Specify the name of controller to be monitored.

Controller Workload Evictable – Indicate if the controller can be interrupted if the node is shut down. Evictable controllers are good candidates to be deployed in Spot instances.

Provision Profile – Indicate if you want to select a provision profile for this application. If needed, you can even create a profile. If Federator.ai is installed in the same Kubernetes cluster as the application, this profile will be used to automatically apply resource recommendations. For remote clusters, you can click the script icon and copy the resource provisioning script for a profile to the remote cluster in order to run auto provisioning. If you do not select a profile, you can click the script icon and copy the generic provisioning script provided by the system. When a resource provisioning script is run in a Kubernetes cluster, it queries Federator.ai for the most recent recommendations for this controller and applies the resource recommendations. Refer to Auto Provisioning Scripts for more information.

Automation: *No Automation* – Indicate if Federator.ai should monitor resource usage only.

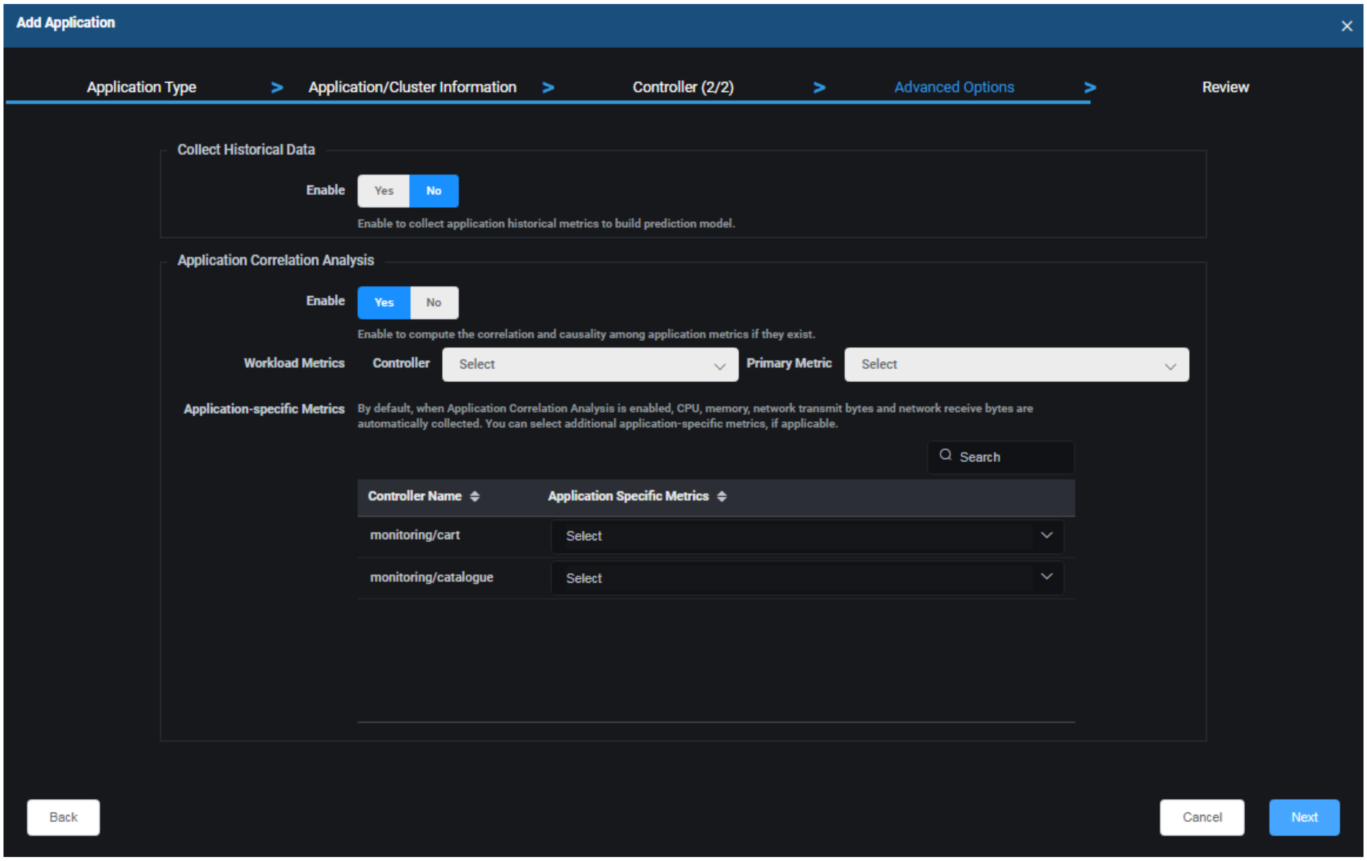
Automation: *Auto Provisioning by Profile* - Indicate if you want to use auto provisioning based on the selected profile. This option can only be selected if Federator.ai is installed in this Kubernetes cluster.

Automation: Horizontal Pod Autoscaling – Indicate if you want to enable Horizontal Pod Autoscaling (HPA). When enabled, the number of pods is automatically increased/decreased based on the CPU/memory usage workload. HPA and Auto Provisioning are mutually exclusive; you can use HPA or auto provisioning, but not both.

Mix/Max Replicas – If *Horizontal Pod Autoscaling* is selected, specify the minimum and maximum number of pods.

3. Click *Add Controller* and repeat the previous step to add more controllers. When done, click *Next* to continue.

4. Configure advanced options. When done, click *Next* to continue.



Add Application
×
Application Type > Application/Cluster Information > Controller (2/2) > Advanced Options > Review

Collect Historical Data

Enable: Yes [✓ No]

Enable to collect application historical metrics to build prediction model.

Application Correlation Analysis

Enable: [✓ Yes] No

Enable to compute the correlation and causality among application metrics if they exist.

Workload Metrics: Controller: Select; Primary Metric: Select

Application-specific Metrics: By default, when Application Correlation Analysis is enabled, CPU, memory, network transmit bytes and network receive bytes are automatically collected. You can select additional application-specific metrics, if applicable.

Controller Name	Application Specific Metrics
monitoring/cart	Select
monitoring/catalogue	Select

Collect Historical Data – Specify if you want the system to collect up to three months' worth of historical data for existing controllers of this application. This will enable weekly and monthly predictions, recommendations, and cost analysis for newly added applications. If less than three months' worth of data exists, the system will collect the maximum data that is available. Typically, collection of historical data will complete in about 2-3 hours. If you need to pause collection, you can edit application settings.

Note that Federator.ai uses the APIs provided by your metrics data source (Datadog, Sysdig) to access historical data. The data source imposes limits on how many calls can be made to their service per hour. If the rate limit is too low, the queries for historical data may exceed the limit and the API will return an error. You will need to contact your metrics data service provider to raise your API rate limit.

Application Correlation Analysis – Specify if you want the system to provide statistical analysis and predictions based on the correlation between resource usage and application workload. Information will appear on the *Application Analysis* pages.

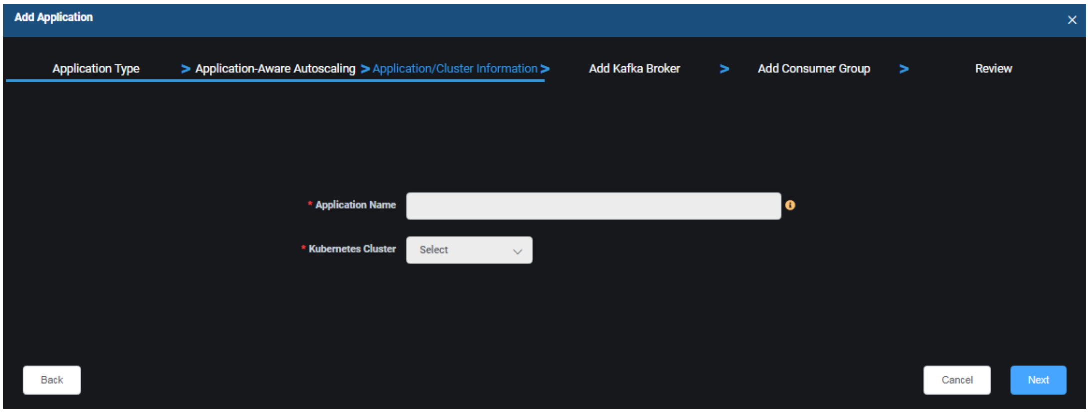
Workload Metrics – If you enabled *Correlation Analysis*, select which controller and which metric from this controller is the primary indicator of application load. The metric can be CPU, memory, number of network bytes received or transmitted.

Application-specific Metrics – Select additional application-specific metrics, if applicable. For example, if you are using mongoDB, mySQL, NGINX, RabbitMQ, or Redis, you can select available metrics for these applications.

5. Review all information and click *Create* to create the application.

Kafka Consumer Application

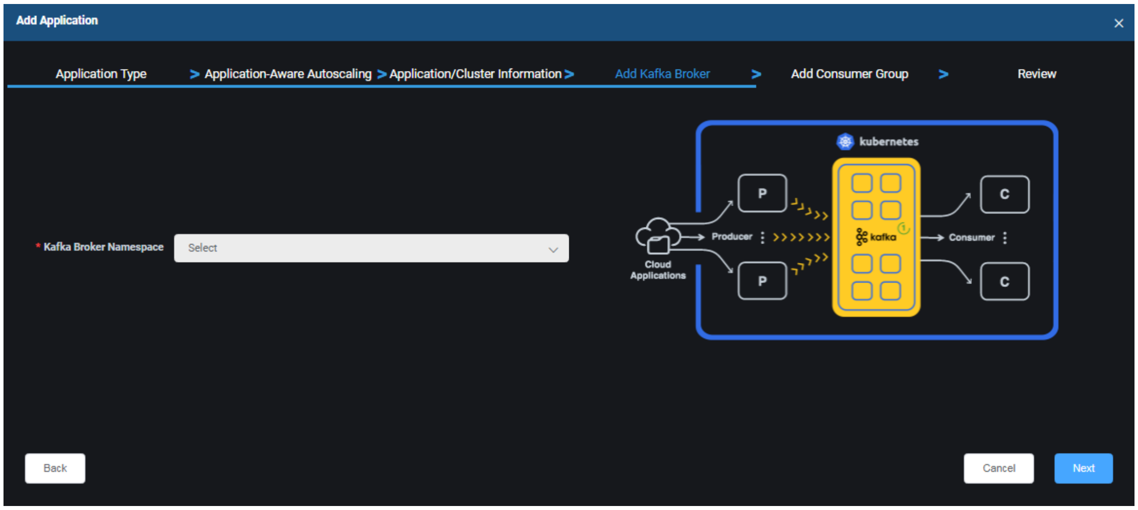
1. Select *Configure Autoscaling for Kafka Consumers* and then click *Next* to continue.
2. Specify an application name and Kubernetes cluster and then click *Next* to continue.



Application Name - Name of the application you want to manage. The name must be a maximum of 253 characters, contain only lowercase alphanumeric characters, "-", or ".", and start and end with an alphanumeric character. Note, an application is not a native Kubernetes object. You will define the controllers that are part of your application later.

Kubernetes Cluster – Select an existing cluster where this application resides.

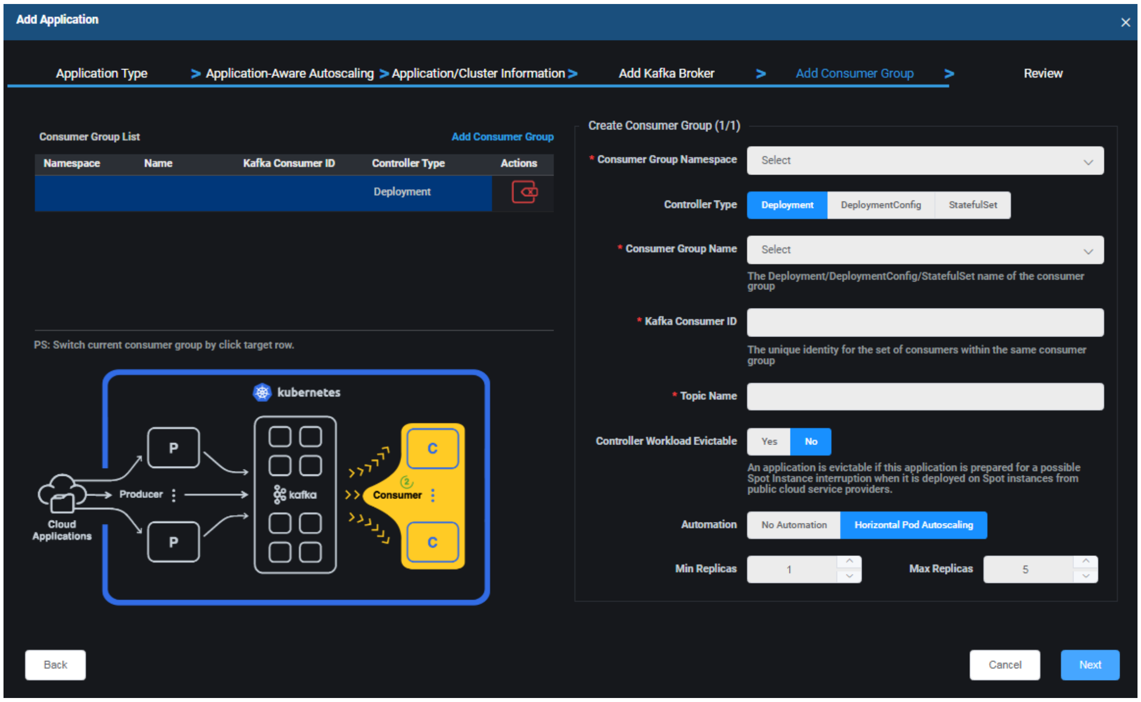
3. Specify the Kafka broker namespace and then click *Next* to continue.



* Kafka Broker Namespace: Select

Kafka Broker Namespace – Select the namespace for the Kafka broker that receives and stores messages from producers and allows consumers to fetch the messages.

4. Configure Kafka consumer groups.



Consumer Group List

Namespace	Name	Kafka Consumer ID	Controller Type	Actions
			Deployment	

PS: Switch current consumer group by click target row.

Create Consumer Group (1/1)

* Consumer Group Namespace: Select

Controller Type: [✓ Deployment] DeploymentConfig StatefulSet

* Consumer Group Name: Select

The Deployment/DeploymentConfig/StatefulSet name of the consumer group

* Kafka Consumer ID:

The unique identity for the set of consumers within the same consumer group

* Topic Name:

Controller Workload Evictable: Yes [✓ No]

An application is evictable if this application is prepared for a possible Spot Instance interruption when it is deployed on Spot instances from public cloud service providers.

Automation: No Automation [✓ Horizontal Pod Autoscaling]

Min Replicas: 1 Max Replicas: 5

Consumer Group Namespace – Select the namespace of the consumer group used for checking the offset on topics and processing messages.

Controller Type – Specify the controller type of the consumer group. Supported controller types are *Deployment, DeploymentConfig* (OpenShift only), and *StatefulSet.*

Consumer Group Name – Specify the name of the Kafka consumer group used for checking the offset on topics and processing messages.

Kafka Consumer ID – Specify the unique ID of the Kafka consumer group for the set of consumers within the same consumer group.

Topic Name – Indicate the topic (where records are stored and published) that will be processed by consumers.

Controller Workload Evictable – Indicate if the controller can be interrupted if the node is shut down. Evictable controllers are good candidates to be deployed in Spot instances.

Automation: *No Automation* – Indicate if Federator.ai should monitor the message production rate only and not autoscale Kafka consumer pods.

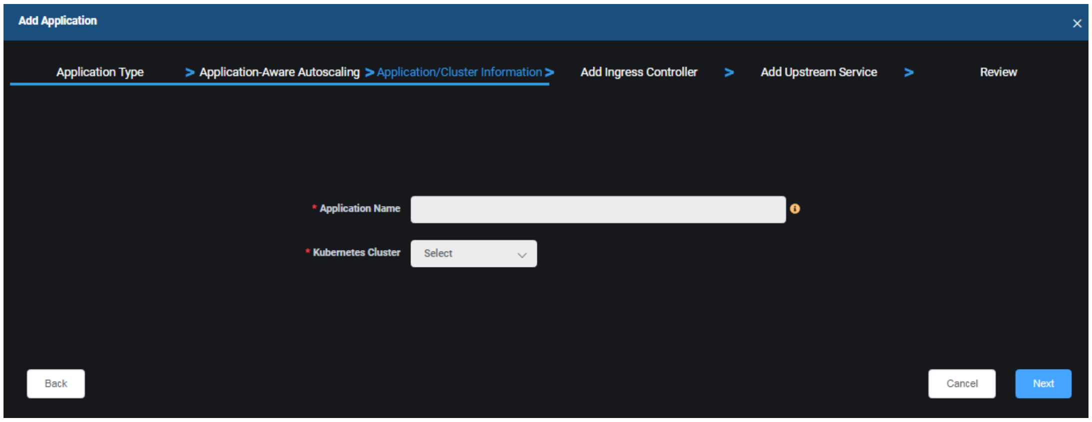
Automation: Horizontal Pod Autoscaling – Indicate if you want to enable Horizontal Pod Autoscaling (HPA). When enabled, the Kafka message production rate is monitored, and the number of Kafka consumer pods is automatically increased/decreased based on the Kafka message production rate.

Mix/Max Replicas – Specify the minimum and maximum number of consumers to be autoscaled.

5. Click *Add Consumer Group* and repeat the previous step to add more consumer groups. When done, click *Next* to continue.
6. Review all information and click *Create* to create the application.

Ingress Application

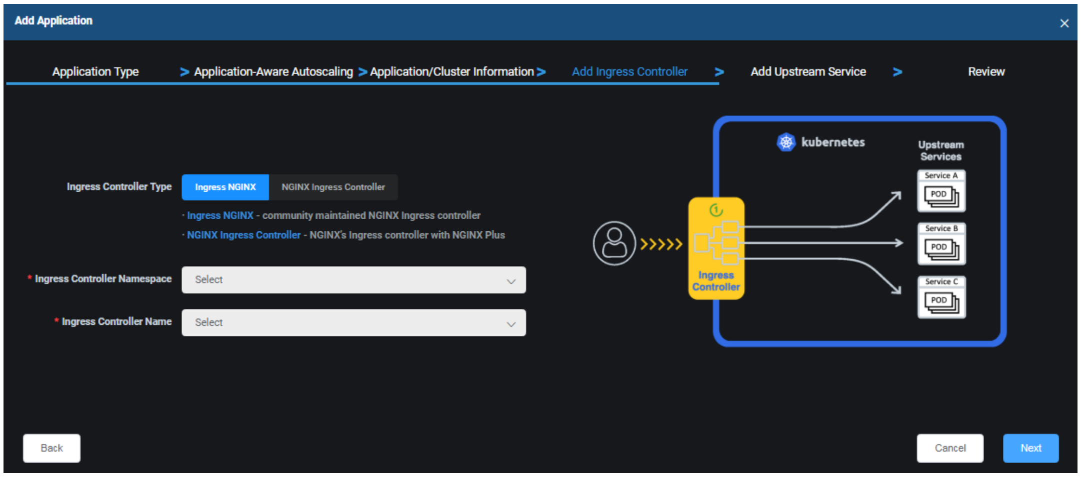
1. Select *Configure Autoscaling for NGINX Ingress Controller Upstream* and then click *Next* to continue.
2. Specify an application name and Kubernetes cluster and then click *Next* to continue.



Application Name - Name of the application you want to manage. The name must be a maximum of 253 characters, contain only lowercase alphanumeric characters, “-“, or “.”, and start and end with an alphanumeric character. Note, an application is not a native Kubernetes object. You will define the controllers that are part of your application later.

Kubernetes Cluster – Select an existing cluster where this application resides.

3. Configure the Ingress controller and then click *Next* to continue.



Add Application

Application Type > Application-Aware Autoscaling > Application/Cluster Information > Add Ingress Controller > Add Upstream Service > Review

Ingress Controller Type: [✓ Ingress NGINX] NGINX Ingress Controller

- Ingress NGINX - community maintained NGINX Ingress controller
- NGINX Ingress Controller - NGINX's Ingress controller with NGINX Plus

* Ingress Controller Namespace: Select

* Ingress Controller Name: Select

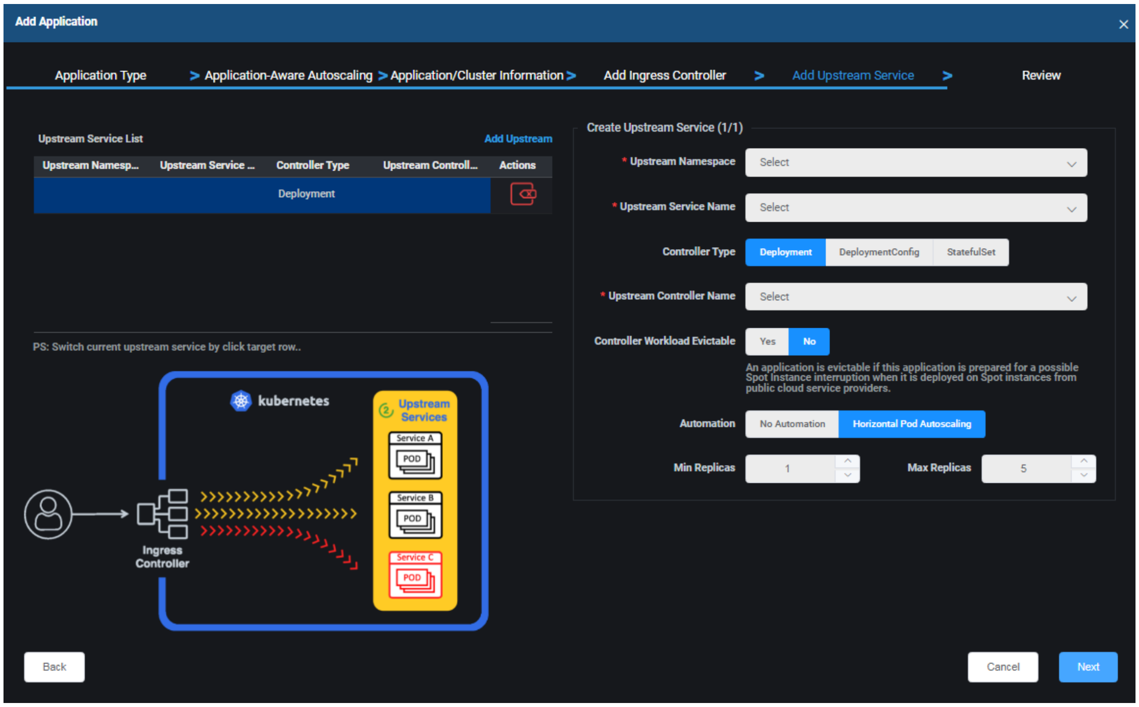
Ingress Controller Type – Select the controller type. Ingress NGINX is the community-supported version and NGINX Ingress Controller is a commercial version by F5. The NGINX Ingress Controller option is only available when you select a Kubernetes cluster that uses Prometheus as the metrics data source.

Ingress Controller Namespace – Select the namespace of the Ingress controller.

Ingress Controller Name – Specify the deployment name of the Ingress controller.

4. Configure upstream services.

   An *upstream* service is provided by a group of upstream servers that receive HTTP requests from an Ingress controller. You must identify the Ingress controller and upstream HTTP service that you want to scale.



Add Application

Application Type > Application-Aware Autoscaling > Application/Cluster Information > Add Ingress Controller > Add Upstream Service > Review

Upstream Service List — Add Upstream

Upstream Namesp...	Upstream Service ...	Controller Type	Upstream Controll...	Actions
		Deployment		

PS: Switch current upstream service by click target row..

Create Upstream Service (1/1)

* Upstream Namespace: Select

* Upstream Service Name: Select

Controller Type: [✓ Deployment] DeploymentConfig StatefulSet

* Upstream Controller Name: Select

Controller Workload Evictable: Yes [✓ No]

An application is evictable if this application is prepared for a possible Spot Instance interruption when it is deployed on Spot instances from public cloud service providers.

Automation: No Automation [✓ Horizontal Pod Autoscaling]

Min Replicas: 1 Max Replicas: 5

Upstream Namespace – Specify the namespace of the upstream service.

Upstream Service Name – Specify the name of the upstream service to be scaled.

Controller Type – Specify the Ingress controller type. Supported controller types are *Deployment, DeploymentConfig* (OpenShift only), and *StatefulSet.*

Upstream Controller Name – Specify the deployment name of the controller that is used to scale the upstream service.

Controller Workload Evictable – Indicate if the controller can be interrupted if the node is shut down. Evictable controllers are good candidates to be deployed in Spot instances.

Automation: *No Automation* – Indicate if Federator.ai should monitor the HTTP services only and not autoscale them.

Automation: Horizontal Pod Autoscaling – Indicate if you want to enable Horizontal Pod Autoscaling (HPA). When enabled, HTTP services are monitored, and the number of services is automatically increased/decreased based on the workload.

Mix/Max Replicas – Specify the minimum and maximum number of services to be autoscaled.

5. Click *Add Upstream* and repeat the previous step to add more upstream services. When done, click *Next* to continue.
6. Review all information and click *Create* to create the application.

Manage Applications

You can do the following from the *Applications* page:

- Edit application settings – Using the application wizard, you can add, edit, or remove a controller (generic application), consumer group (Kafka application), or upstream HTTP service (Ingress application) as well as enable/disable correlation analysis for a generic application. For historical data collection of generic applications, you can:
    - Start the collection of up to three months' worth of historical data (from the current time) for existing controllers of the application, if it was not enabled when the application was added.
    - See the status of data collection, including the time period collected.
    - Pause/resume historical data collection that is in progress.
- View the resource provisioning script associated with a generic controller. If no script is associated, you will see the generic system script. A script can be copied to a remote Kubernetes cluster in order to run auto provisioning for that controller. To do this, click the *Resource Provisioning Script* icon.
- Remove an application. To do this, click the *Remove Application* icon and confirm the removal.

Related topics:

Auto Provisioning
Terminology
Search/Sort Information in Tables
Application Analysis

Configuration – Auto Provisioning

Federator.ai predicts CPU and memory usage for each application controller and application namespace in Kubernetes clusters and makes recommendations for the optimal amount of resources. Auto provisioning can automatically deploy resource recommendations to controllers and namespaces for generic applications based on a pre-defined profile.

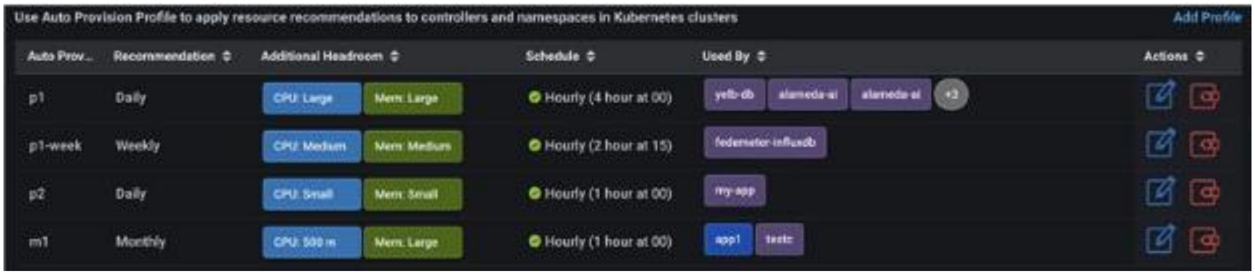
An auto provisioning profile defines the conditions under which the resource recommendations will be automatically applied. It defines which recommendations to use (daily, weekly, or monthly), any adjustments to make on top of the system recommendations, and the schedule for when the resource recommendations should be applied.

If Federator.ai is installed in the same Kubernetes cluster as the application, you can assign auto provisioning profiles to controllers via the *Configuration / Applications* page or assign profiles to namespaces via the *Configuration / Clusters* page.

For remote clusters, you can copy a resource provisioning script to the remote cluster in order to run auto provisioning. Refer to Auto Provisioning Scripts below.

Note that auto provisioning and Horizontal Pod Autoscaling (HPA) are mutually exclusive; you can use HPA or auto provisioning, but not both.

The *Auto Provisioning* page displays all the existing profiles and allows you to add, edit, and remove profiles.



Use Auto Provision Profile to apply resource recommendations to controllers and namespaces in Kubernetes clusters

Auto Prov...	Recommendation ⬍	Additional Headroom ⬍		Schedule ⬍	Used By ⬍	Actions ⬍
p1	Daily	CPU: Large	Mem: Large	Hourly (4 hour at 00)	yelb-db alameda-ai alameda-ai +3	
p1-week	Weekly	CPU: Medium	Mem: Medium	Hourly (2 hour at 15)	federator-influxdb	
p2	Daily	CPU: Small	Mem: Small	Hourly (1 hour at 00)	my-app	
m1	Monthly	CPU: 500 m	Mem: Large	Hourly (1 hour at 00)	app1 testc	

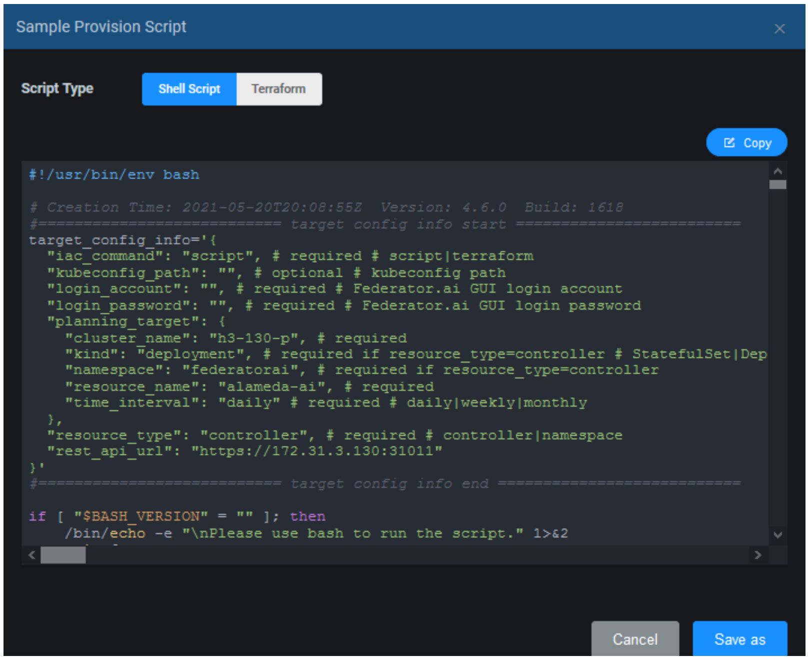
For each profile, you will see the frequency of the recommendations, CPU and memory adjustments, auto provisioning schedule, and which controllers and namespaces are using the profile. Purple represents a controller and blue represents a namespace.

Auto Provisioning Scripts

When you create an auto provisioning profile, the system generates a *resource provisioning script* that contains all the conditions in the profile.

For remote clusters, you can copy a resource provisioning script to the remote cluster in order to run auto provisioning. You can use a script associated with an auto provisioning profile or you can use the generic provisioning script provided by the system. This generic script uses system recommendations and does not have any adjustments or boundary (min/max) settings. When a resource provisioning script is run in a Kubernetes cluster, it queries Federator.ai for the most recent recommendations and applies them to a controller or a namespace.

You can find the scripts, save the scripts locally, and copy these scripts via the following pages: *Configuration / Applications, Configuration / Clusters* (when you edit a namespace), or *Planning / Kubernetes Workload Prediction* (when you are viewing information for controllers or namespaces).

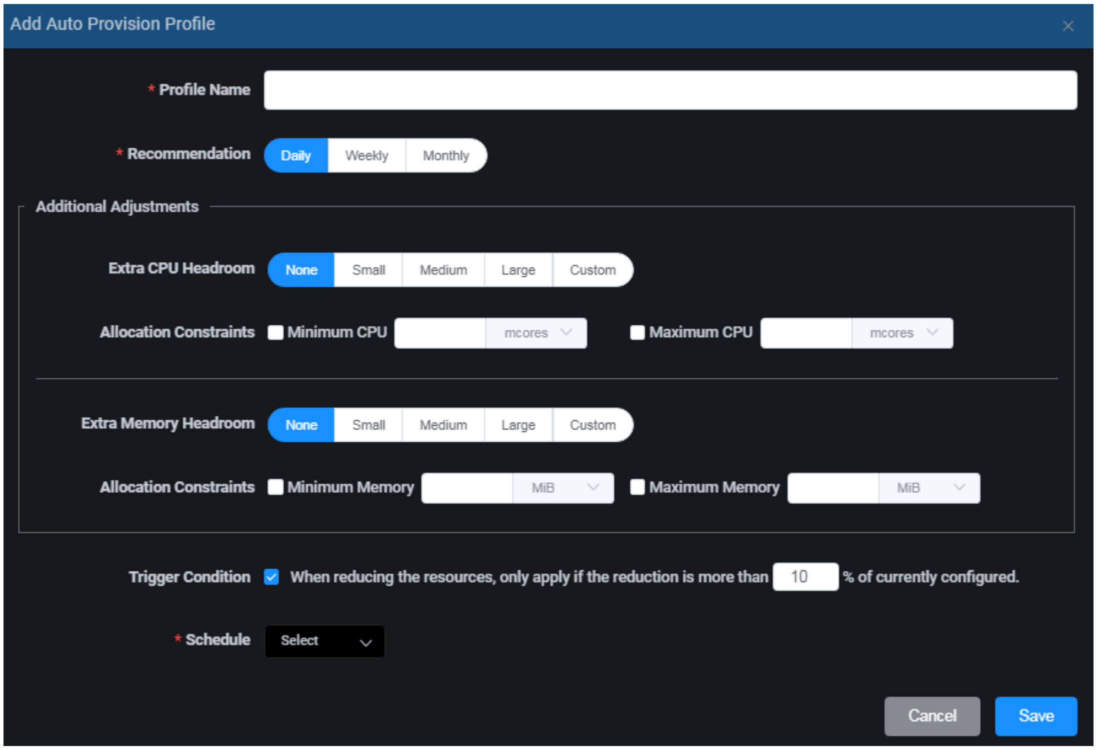


```
#!/usr/bin/env bash

# Creation Time: 2021-05-20T20:08:55Z  Version: 4.6.0  Build: 1618
#=========================== target config info start ==========================
target_config_info='{
  "iac_command": "script", # required # script|terraform
  "kubeconfig_path": "", # optional # kubeconfig path
  "login_account": "", # required # Federator.ai GUI login account
  "login_password": "", # required # Federator.ai GUI login password
  "planning_target": {
    "cluster_name": "h3-130-p", # required
    "kind": "deployment", # required if resource_type=controller # StatefulSet|Dep
    "namespace": "federatorai", # required if resource_type=controller
    "resource_name": "alameda-ai", # required
    "time_interval": "daily" # required # daily|weekly|monthly
  },
  "resource_type": "controller", # required # controller|namespace
  "rest_api_url": "https://172.31.3.130:31011"
}'
#=========================== target config info end ============================

if [ "$BASH_VERSION" = "" ]; then
    /bin/echo -e "\nPlease use bash to run the script." 1>&2
```

You can select a shell script to be run in a Kubernetes cluster where the application is run. For Terraform integration, you can select the Terraform script.

Add a Profile

1. On the *Configuration / Auto Provisioning* page, click *Add Profile*.



Add Auto Provision Profile

* Profile Name

* Recommendation: [✓ Daily] Weekly Monthly

Additional Adjustments

Extra CPU Headroom: [✓ None] Small Medium Large Custom

Allocation Constraints: □ Minimum CPU mcores □ Maximum CPU mcores

Extra Memory Headroom: [✓ None] Small Medium Large Custom

Allocation Constraints: □ Minimum Memory MiB □ Maximum Memory MiB

Trigger Condition: ☑ When reducing the resources, only apply if the reduction is more than 10 % of currently configured.

* Schedule: Select

Profile Name – Specify a name for the profile.

Recommendation – Specify which system recommendations to use (daily, weekly, or monthly).

Adjustments: Extra Headroom – If desired, specify any adjustments to make on top of the system recommendations for CPU and memory. *Small* means that the adjustment is 10% more than the recommendation, *Medium* means 20%, and *Large* means 30%. You can also specify a custom adjustment (millicores or percent for CPU; MB, GB, or percent for memory).

Adjustments: *Allocation Constraints* – If desired, specify minimum and maximum limits for CPU and memory. Resources will not be deployed if above or below these boundaries.

Trigger Condition – Recommendations may trigger a reduction of resources. If desired, specify a percentage to limit reduction of resources that are currently configured. The application will be restarted when resources are reduced. Therefore, you should be careful not to make the difference too small, causing frequent application restarts.

Schedule – Specify when the resource recommendations should be applied. If you are using *Daily* recommendations, you may want to apply changes hourly, daily, or automatically at midnight. For *Weekly* recommendations, you may want to apply changes hourly, daily, weekly, or automatically (12:00 a.m. Sunday). For *Monthly* recommendations, you may want to apply changes daily, weekly, monthly, or automatically (12:00 a.m. on the first day of the month). Note that all times are local.

2. Click *Save* when you are done.

Manage Profiles

You can do the following from the *Configuration / Auto Provisioning* page:

- Edit a profile. To do this, click the *Edit Profile* icon.
- Remove a profile. You can only remove a profile if it is not being used by a namespace or controller. To do this, click the *Remove Profile* icon.

Related topic:

Applications
Terminology
Search/Sort Information in Tables
Configure Applications

Configuration – System Settings

The *System Settings* page has tabs that allow you to:

- Change the admin password.
- Update metrics data source information.
- Set system notification.
- Manage the system license.
- Set the policy/update price books.

Admin Password

To access this page, select *Configuration, System Settings, Admin Password* tab. The *Admin Password* page allows you to change the admin password.

You must know the current password and *New Password* must match *Confirm Password*.

Metrics Data Source

The *Metrics Data Source* page allows you to set the authentication values that are needed by clusters to access metrics from different data sources. To access this page, select *Configuration, System Settings, Metrics Data Source* tab. You will need to select a cluster type, data source, and cluster.

For Kubernetes clusters, the available data sources are Prometheus, Datadog, and Sysdig.

For VM clusters, the available data sources are vCenter and AWS CloudWatch.

- For Datadog, the *API Key* and *Application Key* are required for authentication.
- For Sysdig, a *URL* and *Token* are required for authentication. Click the information icon to see supported URLs.
- For the Prometheus open-source monitoring system, the *URL* is required but the *Token* is optional.
- For vCenter, a *Login ID* and *Password* are required for authentication to the specified vCenter. A *Cluster Path* is needed for identifying VMs to be managed.
- For AWS CloudWatch, the *Region, Access Key ID*, and *Secret Access Key* are required for authentication.

When you are done, click *Test Connection* to confirm that all information is correct.

Note that for VMware, you can only modify cluster information if there are no VMs being monitored for the cluster.

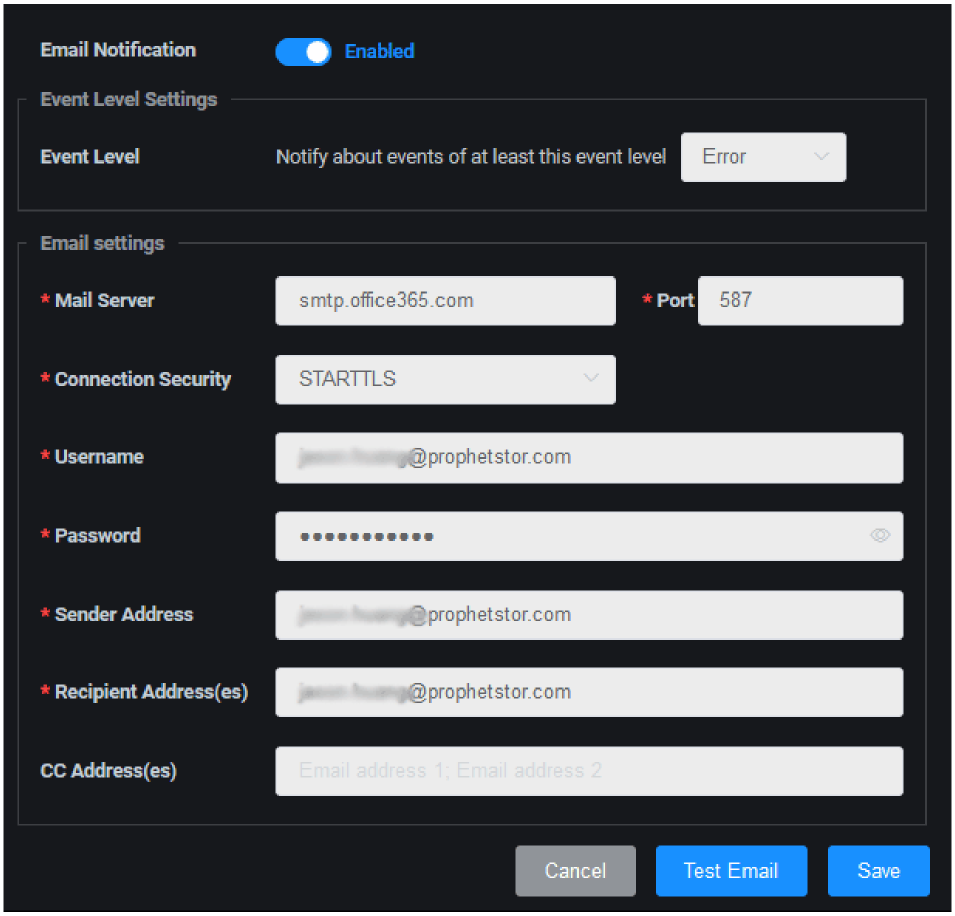
Notification

The *Notification* page allows you to configure email notifications to administrators when system errors and fatal issues occur. To access this page, select *Configuration, System Settings, Notification* tab.

Enable Notification

Follow the steps below to enable notification:

1. Toggle the *Email Notification* icon to *Enabled*.



Email Notification: Enabled

Event Level Settings

Event Level: Notify about events of at least this event level: Error

Email settings

* Mail Server: smtp.office365.com
* Port: 587
* Connection Security: STARTTLS
* Username: [redacted]@prophetstor.com
* Password: ●●●●●●●●●●●●
* Sender Address: [redacted]@prophetstor.com
* Recipient Address(es): [redacted]@prophetstor.com

CC Address(es): Email address 1; Email address 2

Event Level – Select the minimum event level that should trigger notification. Higher levels will also trigger an email. For example, if you select *Error*, fatal events will also trigger an email.

Mail Server - Specify the mail server that should be used to send notification emails.

Port - Specify the mail server port that should be used.

Connection Security – Specify the protocol used by the mail server to secure email transmissions.

Username/Password - Specify the user account that will be used to log into the mail server.

Sender Address - Specify the email account that will be used in the “From” field of emails that are sent.

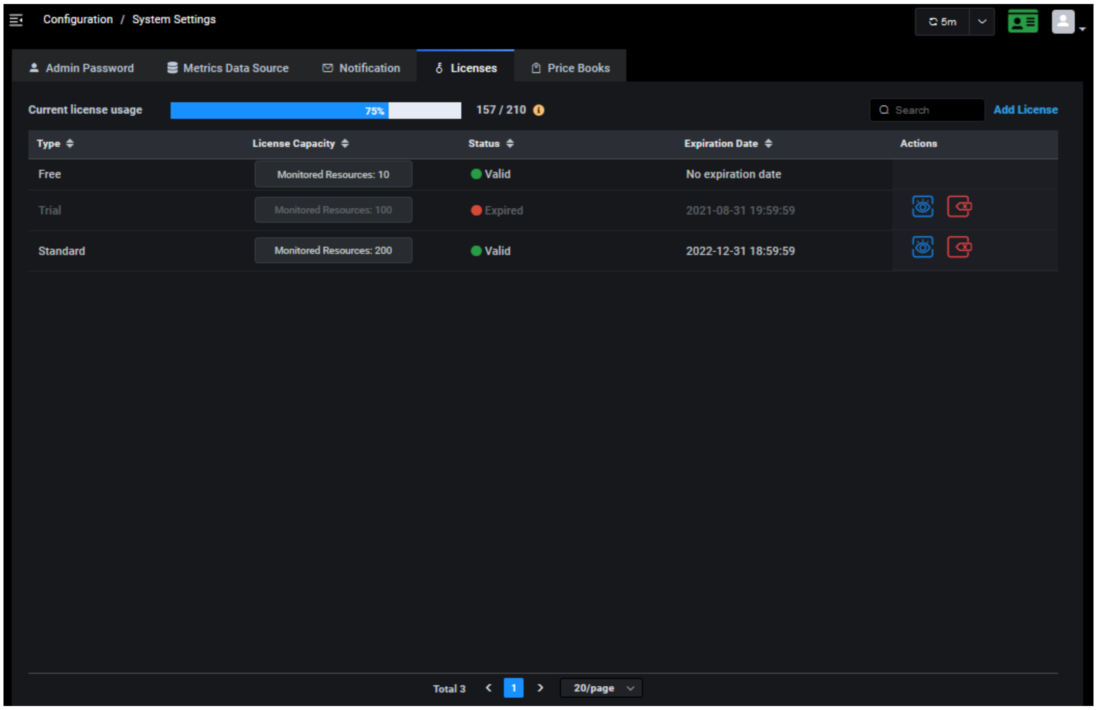
Recipient Address(es) - Specify the email address of the account that will receive emails. This will be used in the “To” field of emails. Separate multiple email addresses with semicolons.

CC Address(es) - Specify any other email accounts that should receive emails. Separate multiple email addresses with semicolons.

2. Click *Test Email* to confirm that all information is correct.
3. Once the test emails are received, click *Save*.

Licenses

To access this page, select *Configuration, System Settings, Licenses* tab. The *Licenses* page displays your current system licenses, including license type, status, and expiration date. It also shows what capacity is included in your license and your current usage.



Current license usage 75% 157 / 210

Type	License Capacity	Status	Expiration Date	Actions
Free	Monitored Resources: 10	● Valid	No expiration date	
Trial	Monitored Resources: 100	● Expired	2021-08-31 19:59:59	
Standard	Monitored Resources: 200	● Valid	2022-12-31 18:59:59	

The *Current license usage* graph displays the percentage of licensed resources being monitored as well as the number of resources being monitored and the total for which you are licensed. For Kubernetes clusters, licensed resources include nodes, namespaces in “Monitoring” state, and configured controllers. For VM clusters, licensed resources include VMs.

There are several license types, *Free, Trial*, and *Standard*. By default, a *Free* license with 10 resources is automatically applied when Federator.ai is installed. A *Trial* license may be provided during Federator.ai evaluation. Once a *Standard* license applied, the *Trial* license expires. The number of licensed resources is cumulative and includes the total number *Free* and *Standard* resources; for trials, it includes the total number of *Free* and *Trial* resources.

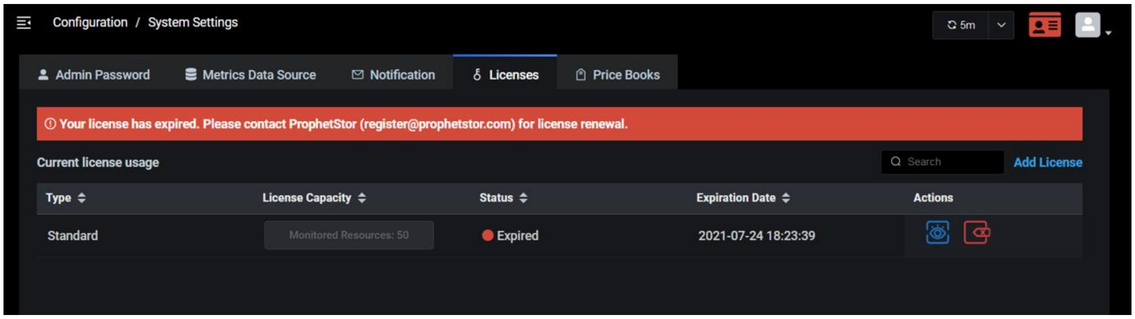
If you reach the number of licensed resources, there is a 30-day grace period, at which time the system prevents you from adding resources and some product functions, such as predictions, recommendations, cost analysis, and autoscaling, will stop. However, data collection will continue until additional license capacity is purchased.

A *Standard* license must be activated within 30 days after install, however a license with a status of *Pending Activation* will still make predictions and recommendations during that period. The status will be *Valid* once the license is activated.

The *Expiration Date* shows when the license expires. If the expiration of a license results in the system exceeding the license limit, predictions, recommendations, cost analysis, and autoscaling will not work.

License Expiration

- Free licenses – Do not expire.
- Trial licenses – Expire on the expiration date displayed. A warning will be displayed when a trial license is expiring within 7 days. There is no grace period for an expired trial license. A trial license will be immediately expired when a standard license is added to the system.
- Standard licenses – Expire on the expiration date displayed. A warning will be displayed when a standard license is expiring within 7 days and will continue during the 30-day grace period after expiration. If a standard license expires and a remaining license does not have enough capacity, predictions and recommendations will stop.



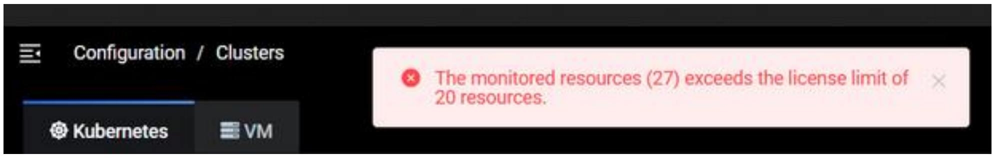
Your license has expired. Please contact ProphetStor (register@prophetstor.com) for license renewal.

Current license usage

Type	License Capacity	Status	Expiration Date	Actions
Standard	Monitored Resources: 50	● Expired	2021-07-24 18:23:39	

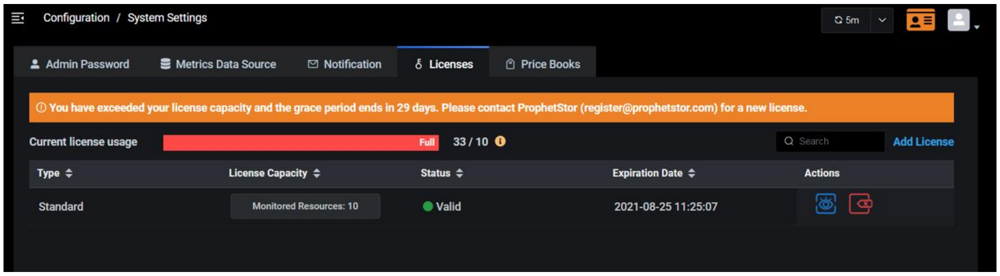
License Limits and Grace Periods

- In most cases, no new monitored resources can be added when there is not enough license capacity.



The monitored resources (27) exceeds the license limit of 20 resources.

- If new cluster nodes are added to a cluster and there is not enough license capacity for new cluster nodes, there will be a 30-day grace period. After the grace period, predictions and recommendations will stop.

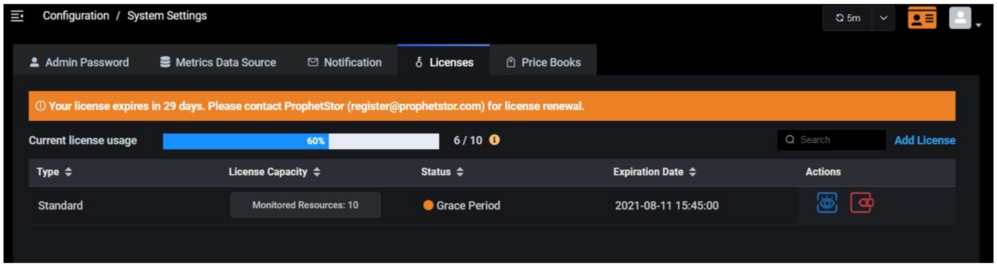


ⓘ You have exceeded your license capacity and the grace period ends in 29 days. Please contact ProphetStor (register@prophetstor.com) for a new license.

Current license usage Full 33 / 10

Type	License Capacity	Status	Expiration Date	Actions
Standard	Monitored Resources: 10	● Valid	2021-08-25 11:25:07	

- There is a 30-day grace period after expiration of a standard license.



ⓘ Your license expires in 29 days. Please contact ProphetStor (register@prophetstor.com) for license renewal.

Current license usage 60% 6 / 10

Type	License Capacity	Status	Expiration Date	Actions
Standard	Monitored Resources: 10	● Grace Period	2021-08-11 15:45:00	

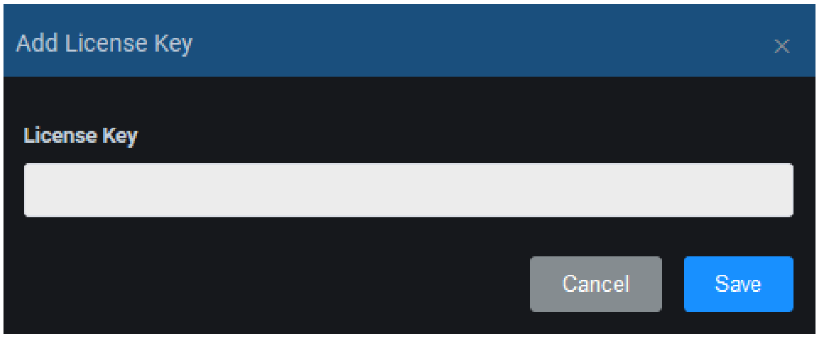
In addition to adding and activating licenses, you can perform the following functions from the *Licenses* page:

Icon	Function
👁	Show license key.
⌫	Remove a license key.

Add a License

Follow the steps below to add and register a license:

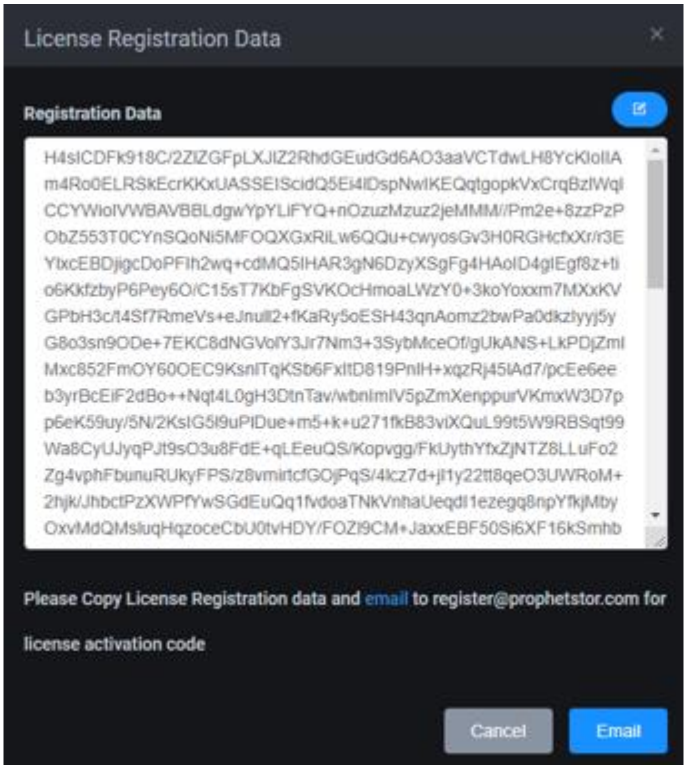
1. Click the *Add License* icon and enter the license key.



2. Click *Add*.

   A trial license is valid immediately after it is added. A standard license needs to be registered in order to be activated.

3. Email the registration data to register@prophetstor.com for a license activation code.



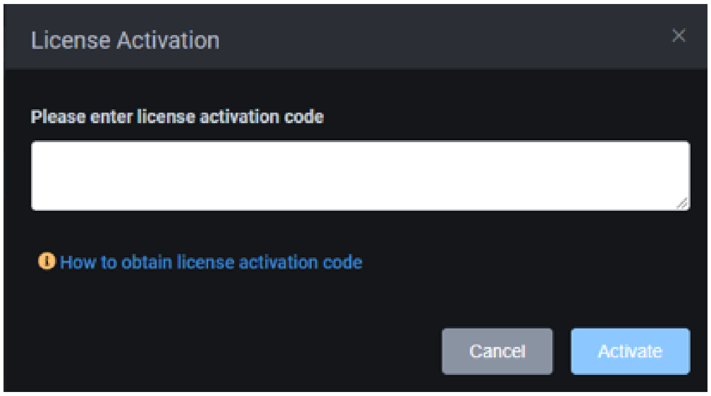
```
H4sICDFk918C/2ZlZGFpLXJlZ2RhdGEudGd6AO3aaVCTdwLH8YcKloIlA
m4Ro0ELRSkEcrKKxUASSElScidQ5Ei4lDspNwIKEQqtgopkVxCrqBzlWql
CCYWioIVWBAVBBLdgwYpYLiFYQ+nOzuzMzuz2jeMMM//Pm2e+8zzPzP
ObZ553T0CYnSQoNi5MFOQXGxRiLw6QQu+cwyosGv3H0RGHcfxXr/r3E
YlxcEBDjigcDoPFIh2wq+cdMQ5IHAR3gN6DzyXSgFg4HAoID4gIEgf8z+ti
o6KkfzbyP6Pey6O/C15sT7KbFgSVKOcHmoaLWzY0+3koYoxxm7MXxKV
GPbH3c/t4Sf7RmeVs+eJnull2+fKaRy5oESH43qnAomz2bwPa0dkzlyyj5y
G8o3sn9ODe+7EKC8dNGVolY3Jr7Nm3+3SybMceOf/gUkANS+LkPDjZml
Mxc852FmOY60OEC9KsnlTqKSb6FxltD819PnlH+xqzRj45lAd7/pcEe6ee
b3yrBcEiF2dBo++Nqt4L0gH3DtnTav/wbnImlV5pZmXenppurVKmxW3D7p
p6eK59uy/5N/2KsIG5l9uPIDue+m5+k+u271fkB83viXQuL99t5W9RBSqt99
Wa8CyUJyqPJt9sO3u8FdE+qLEeuQS/Kopvgg/FkUythYfxZjNTZ8LLuFo2
Zg4vphFbunuRUkyFPS/z8vmirtcfGOjPqS/4kz7d+jl1y22tt8qeO3UWRoM+
2hjk/JhbctPzXWPfYwSGdEuQq1fvdoaTNkVnhaUeqdl1ezegq8npYfkjMby
OxvMdQMsluqHqzoceCbU0tvHDY/FOZl9CM+JaxxEBF50Si6XF16kSmhb
```

Please Copy License Registration data and email to register@prophetstor.com for license activation code
Cancel
Email

Click the *Email* button to launch your email program. Click the *Copy* icon to copy the registration text and paste it into your email before sending.

The license status will now say *Pending Activation*. It must be activated within 30 days.

4. Once you have received the license activation code, click the *Activate License* icon and paste the activation code.



License Activation

Please enter license activation code

How to obtain license activation code

5. Click *Activate*.

Manage Licenses

You can do the following from the *Licenses* page:

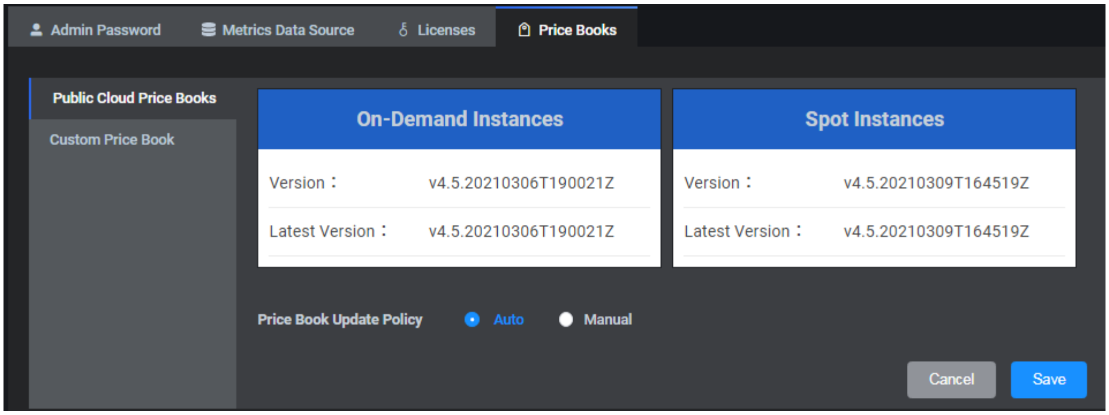
- View the key for your license. To do this, click the *Show License Key* icon.
- Remove the license. To do this, click the *Remove License Key* icon and confirm the removal.

Price Books

To access this page, select *Configuration, System Settings, Price Books* tab.

Public Cloud Price Books

The *Public Cloud Price Books* page displays the versions of the on-demand and spot instance price books that have been collected from the following cloud providers: Amazon Web Services (AWS), Google Cloud, and Microsoft Azure.



On-Demand Instances

Version : v4.5.20210306T190021Z

Latest Version : v4.5.20210306T190021Z

Spot Instances

Version : v4.5.20210309T164519Z

Latest Version : v4.5.20210309T164519Z

Price Book Update Policy ● Auto ○ Manual

The version numbers represent the ProphetStor version of the compiled price books.

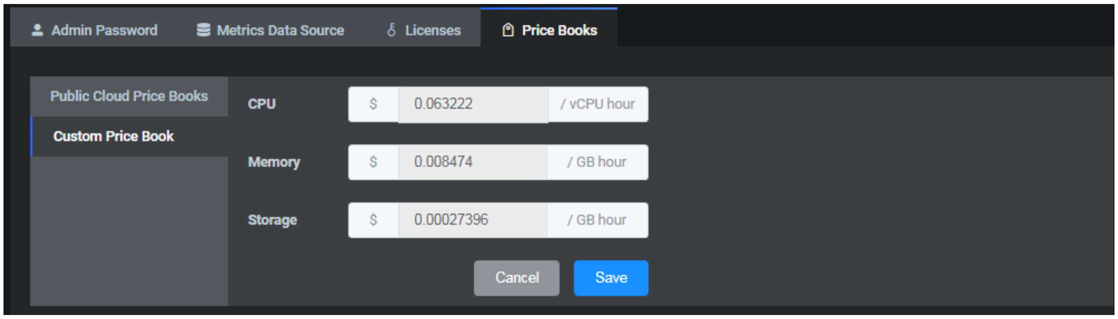
The *Price Book Update Policy* allows you to specify how you want the price books updated. *Auto* regularly checks availability and automatically downloads the latest version, which requires Internet access.

If *Manual* is selected and the *Latest Version* is different from *Version,* Federator.ai is aware of a newer price book but it has not been applied. Click the *Update Price Books* button to update.

Note that the browser needs Internet access in order to display the *Latest Version*. If your computer does not have access, the data must be pushed to system.

Custom Price Book

The *Custom Price Book* page allows you to define your hourly operating costs for CPU, memory, and storage and use these numbers for calculating costs/savings instead of using cloud provider pricing.



CPU	$	0.063222	/ vCPU hour
Memory	$	0.008474	/ GB hour
Storage	$	0.00027396	/ GB hour

When determining your hourly costs, be sure to include electricity, cooling/heating, labor, hardware, etc.

Events

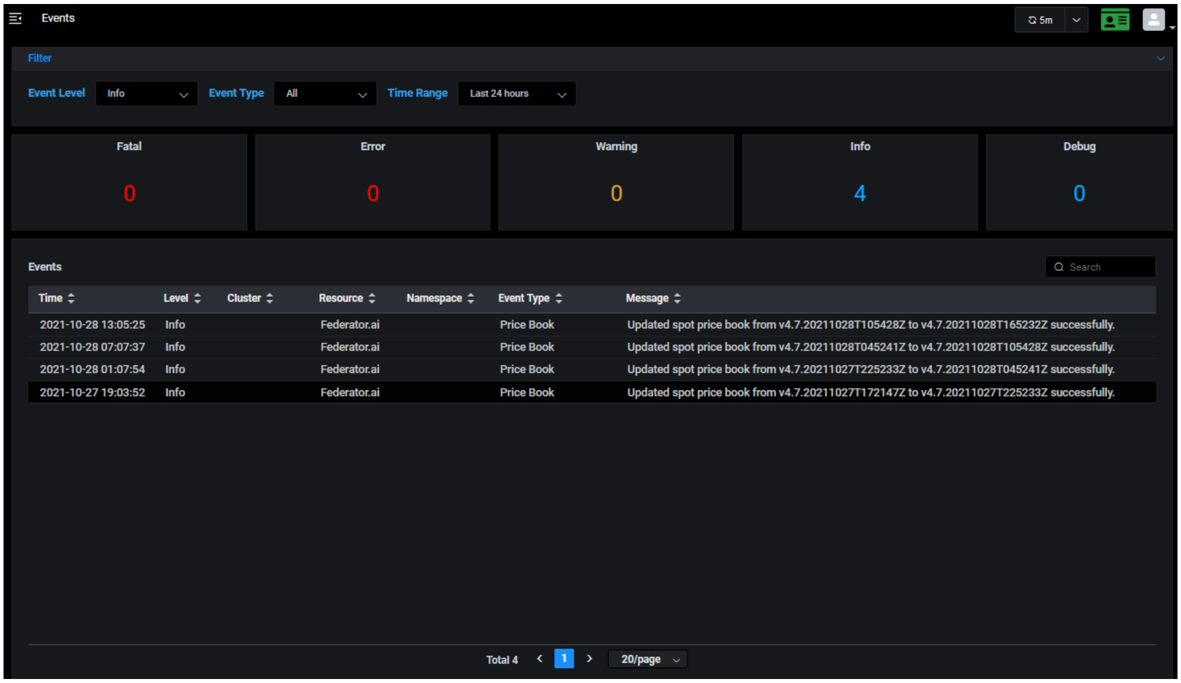
The *Events* page displays all system events that have occurred.

There are five levels of events:

- Fatal - Issues that may stop the system from operating properly.
- Error - Indicates that a failure has occurred.
- Warning - Indicates that something occurred that may require maintenance or corrective action; however, the system is still operational.
- Info - Day-to-day activities, which require no action.
- Debug - Detailed activities used for troubleshooting.

You can filter by the event level, event type, and the time frame to display. For example, if you select the warning level, all warnings, errors, and fatal events will be displayed for the specified time period.

To specify a custom range, select *Custom* under *Time Range* and then specify a date range.



Event Level: Info
Event Type: All
Time Range: Last 24 hours

Fatal	Error	Warning	Info	Debug
0	0	0	4	0

Time	Level	Cluster	Resource	Namespace	Event Type	Message
2021-10-28 13:05:25	Info		Federator.ai		Price Book	Updated spot price book from v4.7.20211028T105428Z to v4.7.20211028T165232Z successfully.
2021-10-28 07:07:37	Info		Federator.ai		Price Book	Updated spot price book from v4.7.20211028T045241Z to v4.7.20211028T105428Z successfully.
2021-10-28 01:07:54	Info		Federator.ai		Price Book	Updated spot price book from v4.7.20211027T225233Z to v4.7.20211028T045241Z successfully.
2021-10-27 19:03:52	Info		Federator.ai		Price Book	Updated spot price book from v4.7.20211027T172147Z to v4.7.20211027T225233Z successfully.

At the bottom of the page, the total number messages being displayed is shown, along with navigation to other pages. You can also determine how many events to show per page.

Related topics:

Terminology

Search/Sort Information in Tables