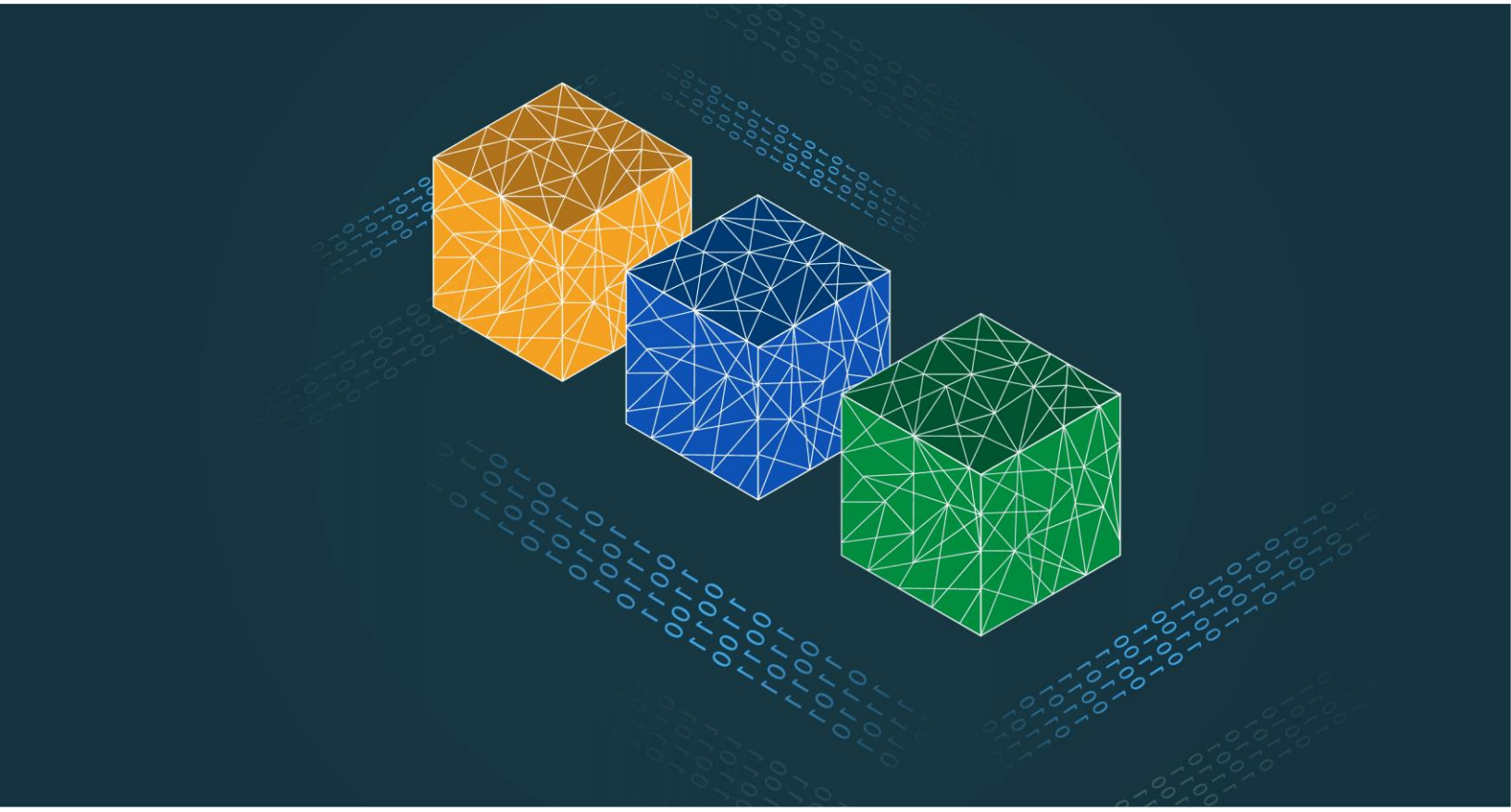
ProphetStor



Federator.ai User Guide

Federator.ai version 4.6.1 User Guide

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

Copyright © 2020-2021 ProphetStor Data Services, Inc. All Rights Reserved.

Federator.ai® is a registered trademark of ProphetStor Data Services, Inc in the United States and other countries.

Kubernetes is a registered trademark of the Linux Foundation, and OpenShift is a trademark of Red Hat, Inc.

All other brand and product names are trademarks or registered trademarks of their respective owners.

6.24.2021

Contents

Overview **4**

Terminology **4**

Getting Started **6**

Access the Federator.ai Portal 6

Setup Wizard 8

Kubernetes Cluster 8

VM Cluster 12

Federator.ai Administration Portal **14**

Portal Sections 14

Portal Icons 14

Common Administration Portal Functions 15

Refresh Statistics 15

License Status 15

User Functions 15

Filter Panel 16

Specify Time Range 16

Search/Sort Information in Tables 17

Show/Hide Metrics in Charts 17

Dashboard **18**

Cluster Workload Prediction 18

Cluster Charts 18

Node/Virtual Machine Chart 20

Namespace Chart (Kubernetes) 20

Application Workload Prediction (Kubernetes) 22

Application Charts 22

Controllers Chart 23

Insight - Cluster Health **25**

Insight - Node Health (Kubernetes) **27**

Planning – Kubernetes or VM Workload Prediction **28**

Managed Nodes Table (Kubernetes) 28

Managed VMs Table (VM) 28

Managed Containers Table (Kubernetes) 29

Workload Prediction Table and Workload Observation and Prediction Charts 29
Workload Prediction Table 30
Workload Observation and Prediction Charts 31
Recommendation - HPA Recommendation (Kubernetes) 33
CPU and memory Charts 33
Optimization – Kafka Consumer (Kubernetes) 34
Number of Replicas Chart 34
Production Rate and Consumption Rate Chart 34
Consumer Lag Chart 35
Consumer Queue Latency Chart 35
CPU and Memory Observation Charts 36
Optimization – Ingress Upstream Services (Kubernetes) 37
Number of Replicas Chart 37
HTTP Request Rate Chart 37
HTTP Response Error Rate Chart 38
Average Response Time Chart 38
Upstream Latency Chart 39
CPU and Memory Observation Charts 39
Cost – Multi-cloud Cost Analysis (Kubernetes and VM Clusters) 40
CPU and Memory Utilization Charts 42
Recommended Cluster Configuration 42
Cost – VM Cost Analysis (VM Clusters and VMs) 44
VM Cost 44
Daily Spend Chart 45
Cumulative Cost Savings Projection Chart 45
Cost – Application Cost Analysis (Kubernetes Applications) 47
Application Cost Savings Analysis 47
Application Costs 47
Cost Trends 48
Cost - Cost Allocation (Kubernetes Namespaces) 50
Current Cluster Configuration 50
Namespace Charts 50
Configuration - Clusters 53
Kubernetes Clusters 53

Add a Kubernetes Cluster 54
Manage Kubernetes Clusters 55
VM Clusters 56
Add a VM Cluster 57
Manage VM Clusters 58
Configuration – Applications 59
Add an Application 59
Generic 60
Kafka Consumer 61
Ingress 62
Manage Applications 64
Configuration – Auto Provisioning 64
Auto Provisioning Scripts 66
Add a Profile 67
Manage Profiles 68
Configuration – System Settings 69
Admin Password 69
Metrics Data Source 69
Notification 70
Enable Notification 70
Licenses 72
Add a License 72
Manage Licenses 73
Price Books 74
Public Cloud Price Books 74
Custom Price Book 74
Events 75

Overview

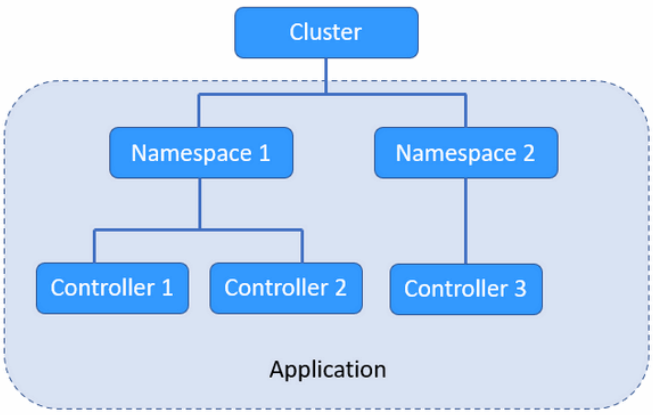
ProphetStor Federator.ai is an AI-based solution that helps enterprises manage and optimize resources for applications on Kubernetes and virtual machines (VMs) in VMware clusters. Using advanced machine learning algorithms to predict application workloads, Federator.ai offers:

- 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
- 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 – In Kubernetes, the ability for the system to automatically increase or decrease containers/pods based on workload demands.

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.

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

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 Applications
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
```

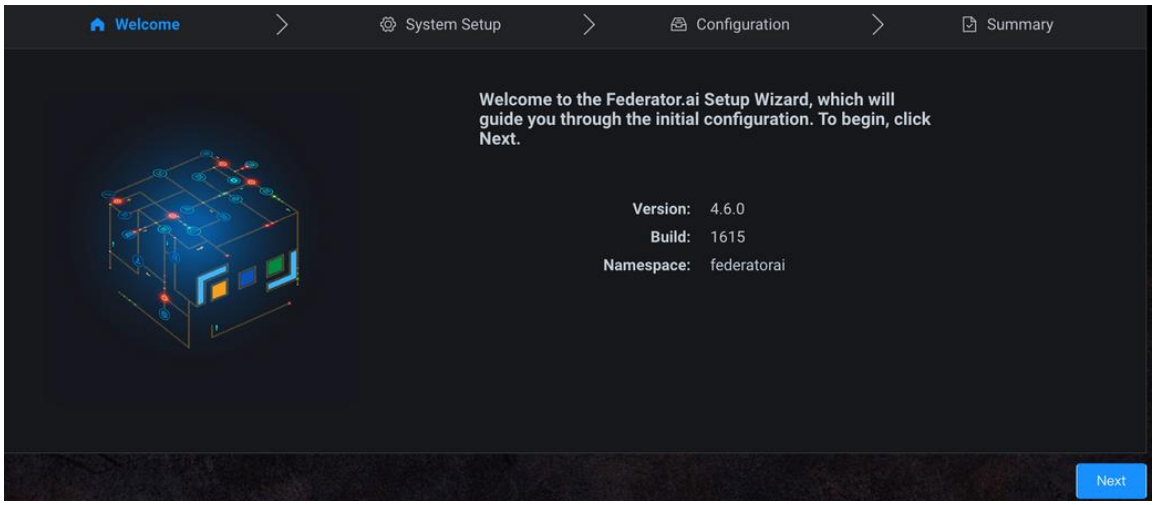
The output will look something like this:

```
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`

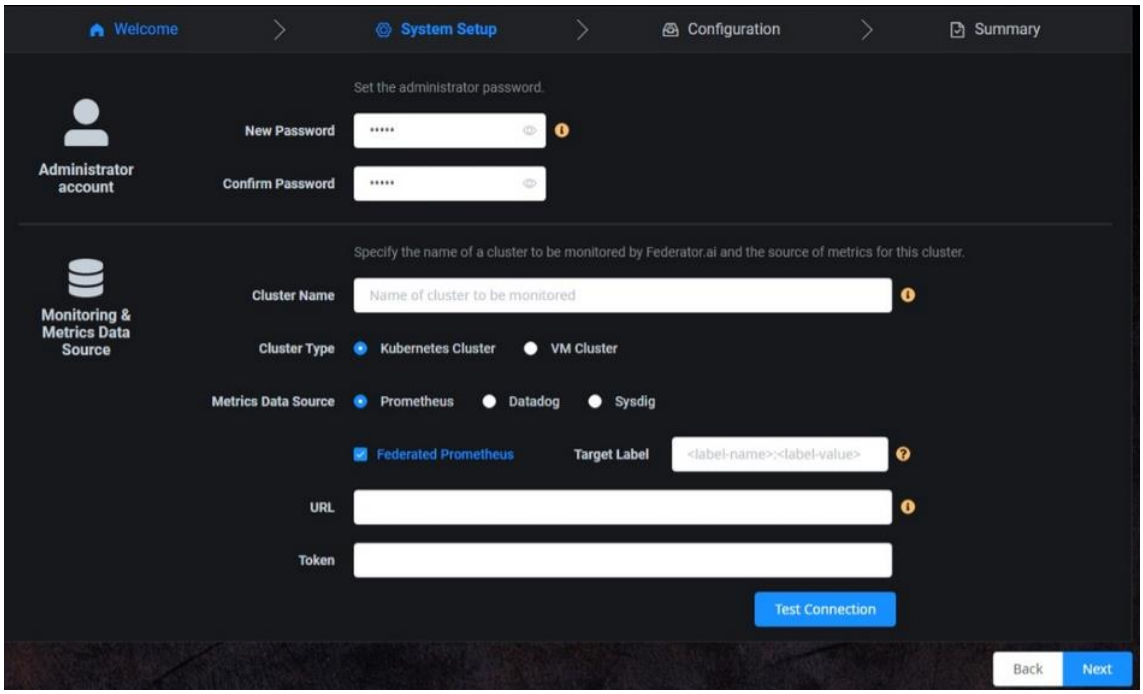
Setup Wizard

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.



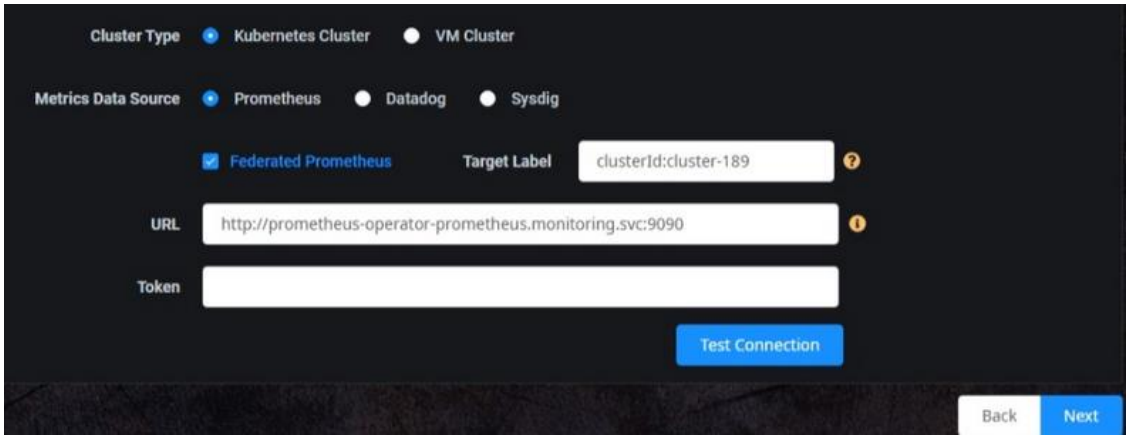
Kubernetes Cluster

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.

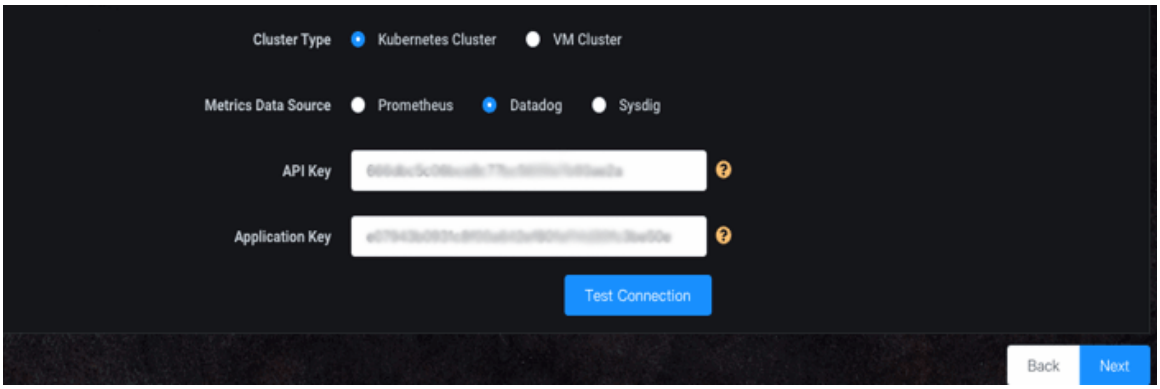


   You must set the administrator password in order to continue but your cluster can be configured later from the *Configuration* section of the portal after completing the setup wizard.

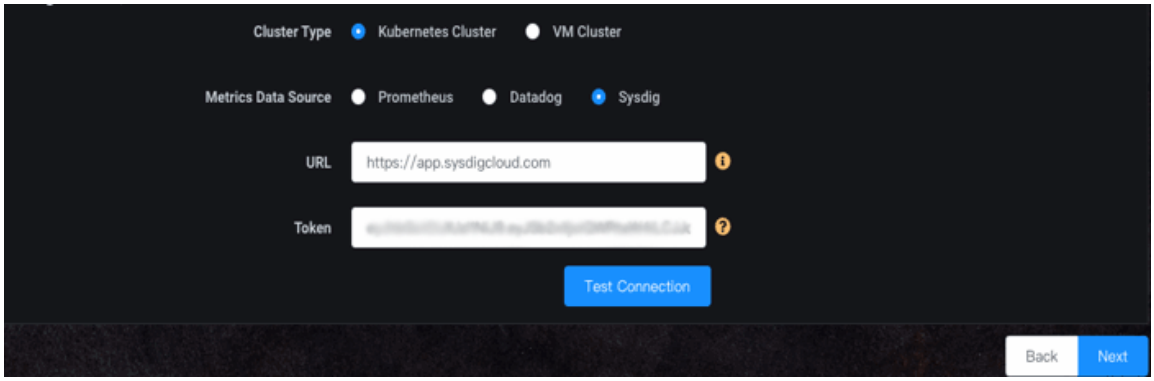
For the Prometheus open-source monitoring system, the URL is required but the token is optional for authentication. Specify if you are using Federation, which is a group of Prometheus servers that send metrics to a centralized Prometheus server. You will need to specify the target label of the centralized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



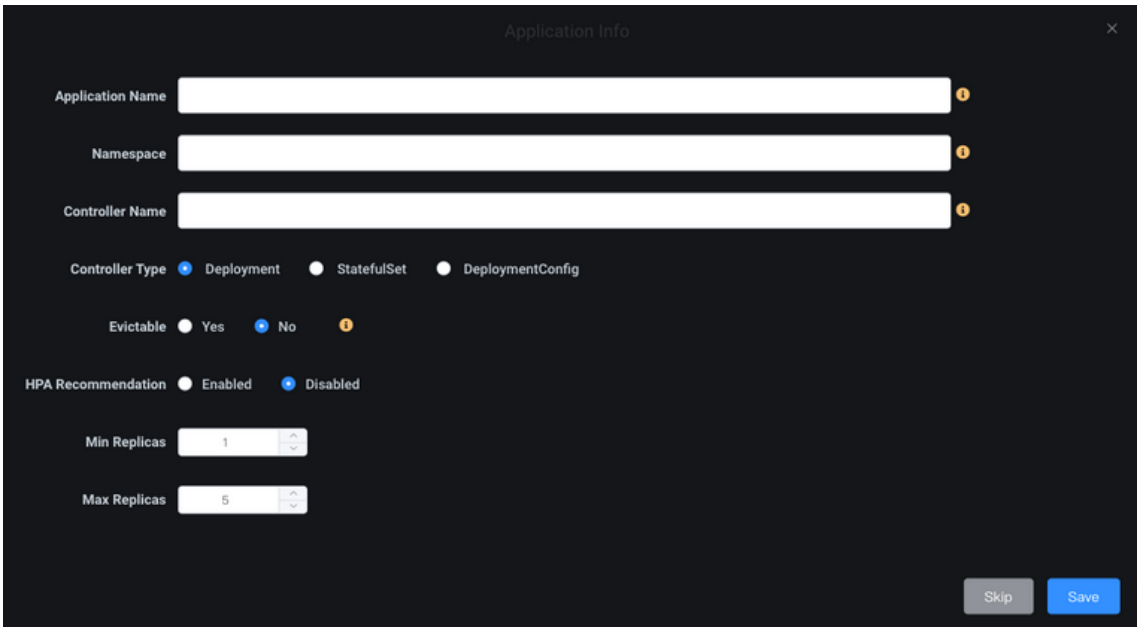
For Datadog, the API key and application key are required for authentication. If needed, you can click on the link to the Datadog website, which is included in the popup help text.



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.



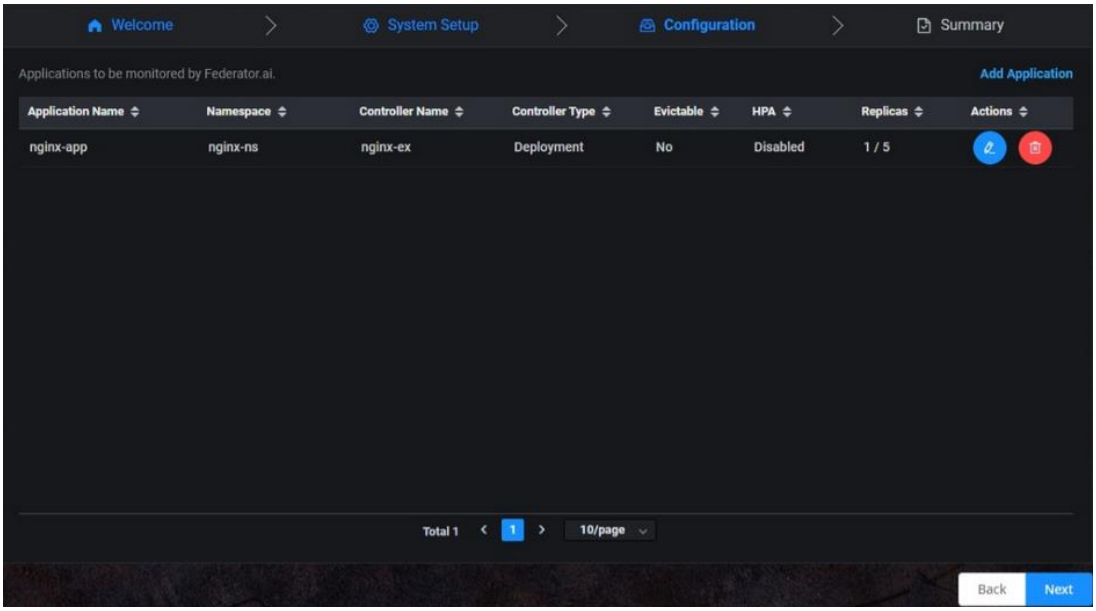
2. Enter information about the first application you want to monitor. You can skip this step and configure applications to be monitored from the *Configuration* section of the portal.



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.

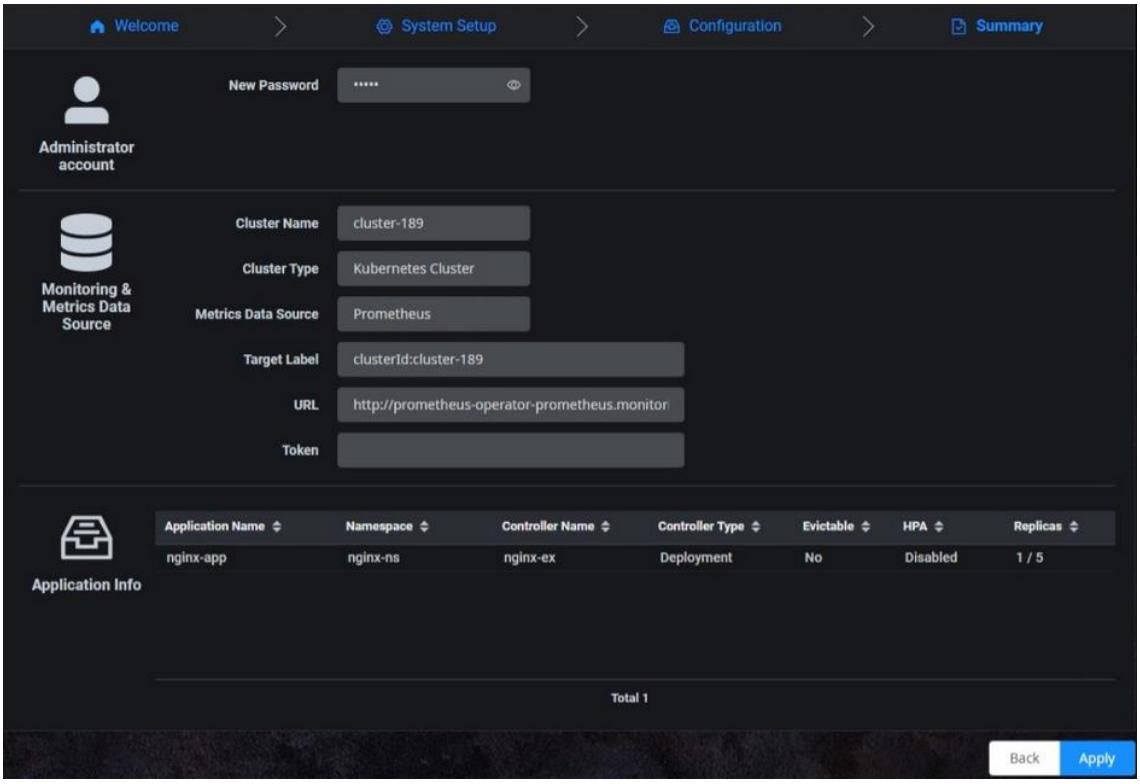
3. Click *Save* to view a list of applications that will be monitored by Federator.ai.



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.

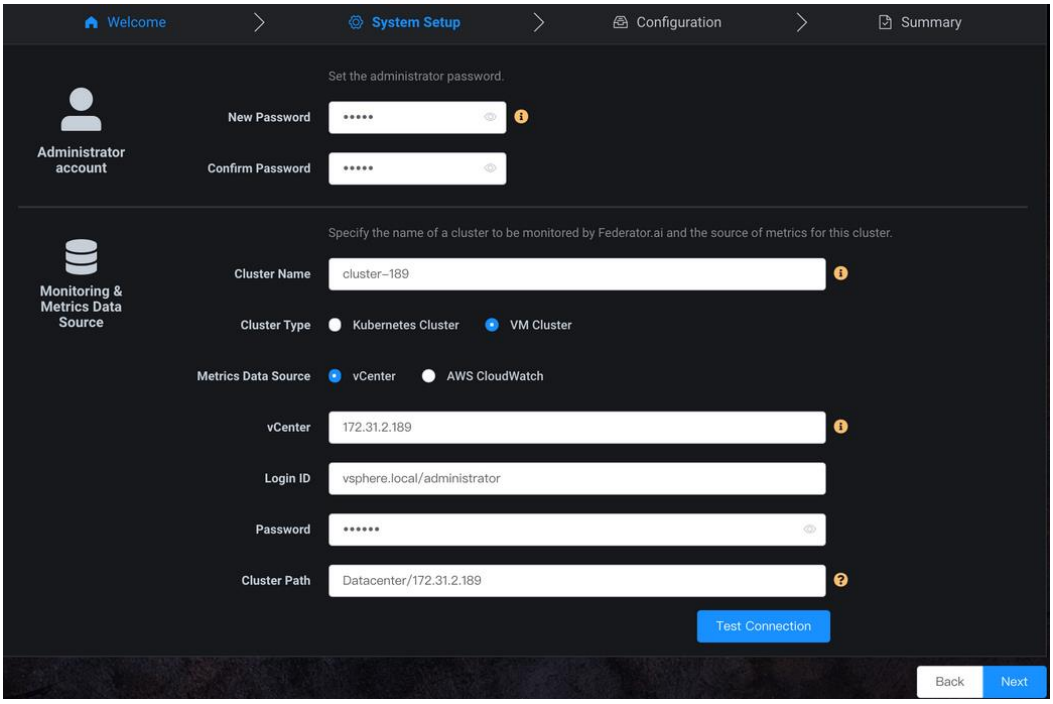
4. Click *Next* to view the *Summary* screen.



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.

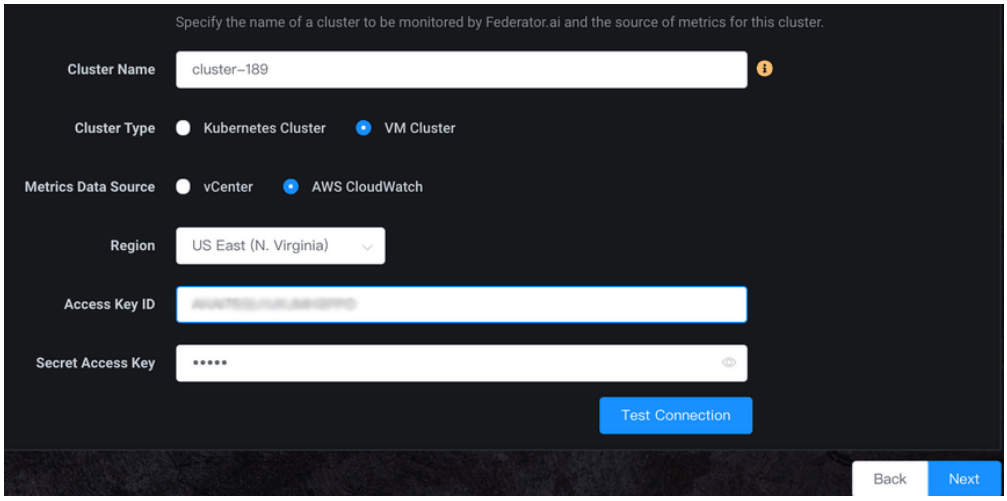


You must set the administrator password in order to continue but your cluster can be configured later from the *Configuration* section of the portal after completing the setup wizard.

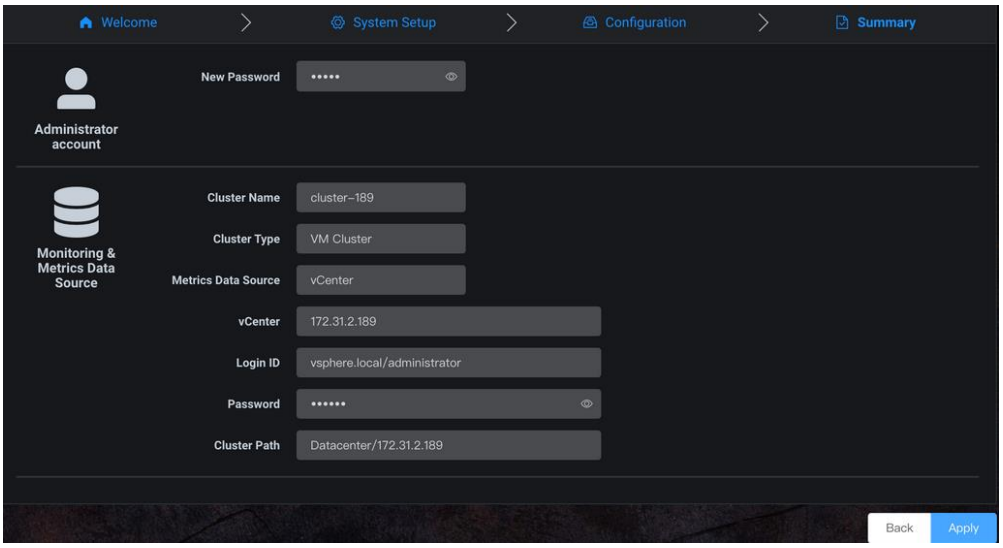
The cluster name must have a maximum of 253 lowercase characters,"-", or "." allowed. The name must start and end with an alphanumeric character.

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.



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.



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
System Configuration

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.
- Insight - Cluster and Kubernetes node health information, including CPU utilization, memory utilization, disk capacity, and VM network throughput.
- Planning – Forecasting tools, including actual CPU and memory usage observations, predicted workload usage, and recommendations for Kubernetes and VM.
- Recommendation - (Kubernetes) HPA recommendations for controllers enabled with autoscaling.
- Optimization - (Kubernetes) Autoscaling predictions for each Kafka individual topic/consumer group and Ingress upstream service.
- Cost – Actual cost and potential savings based on recommendations for clusters, Kubernetes applications, VMs, and Kubernetes namespaces.
- 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, the following icons are used throughout the portal:

Icon	Function
	Kubernetes clusters
	VM clusters. The cluster type is: AWS CloudWatch VM cluster configured with AWS Auto Scaling groups. AWS CloudWatch VM cluster with individual VMs. vCenter cluster.

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

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

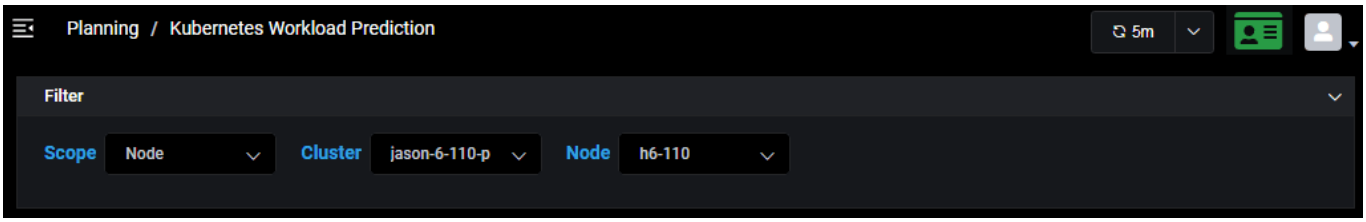
Click the *License* icon at the top right of the dashboard to see Federator.ai license information, including key code, type, CPU cores being monitored, as well as license expiration and status.

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.

Filter Panel

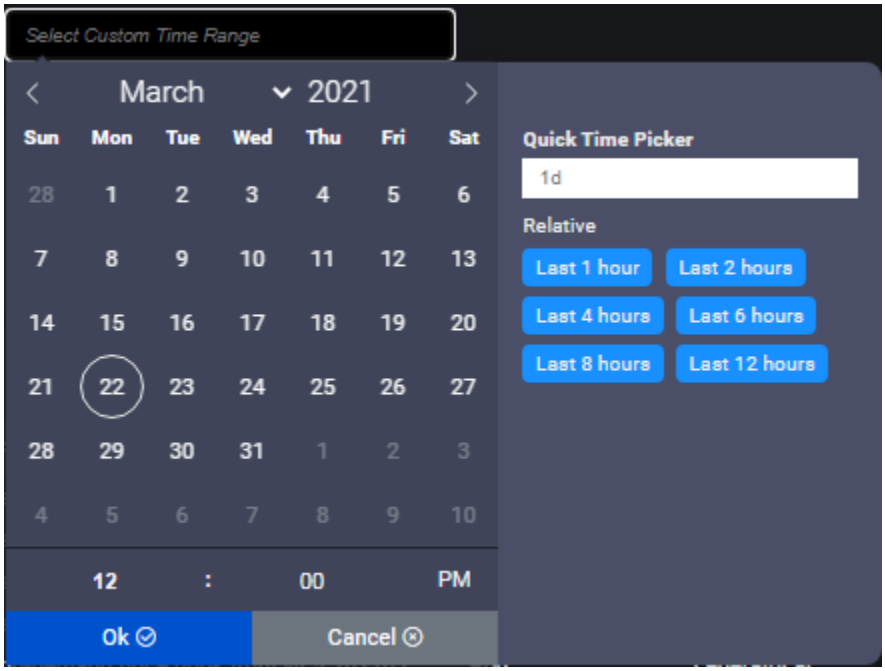
The *Filter* panel appears on most pages and allows you to filter the display of information. Depending upon the page and cluster type, you may be able to select a scope/resource type, cluster, application, namespace, controller, time range, etc.



Specify Time Range

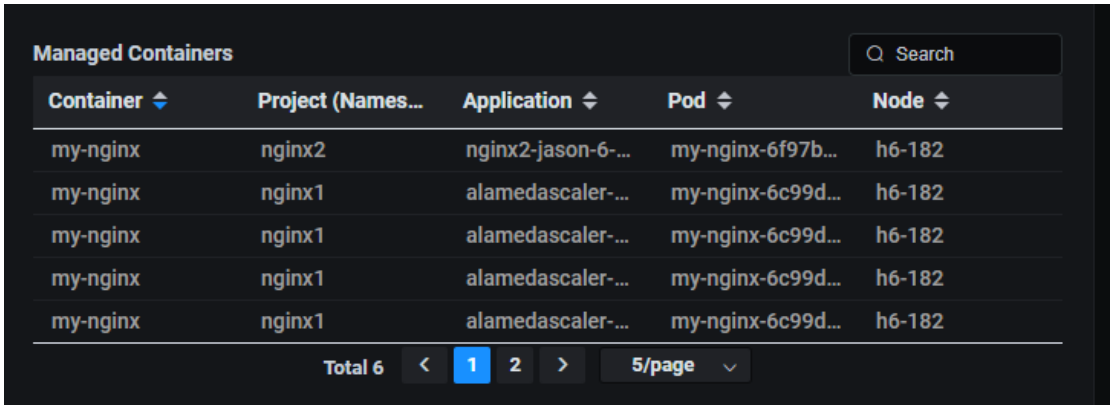
When a chart allows you to specify a time range to display data, you can select a predefined time frame (e.g., last 1 hour, last 24 hours) from the drop-down box or you can specify a custom time range via one of the following methods:

- 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.



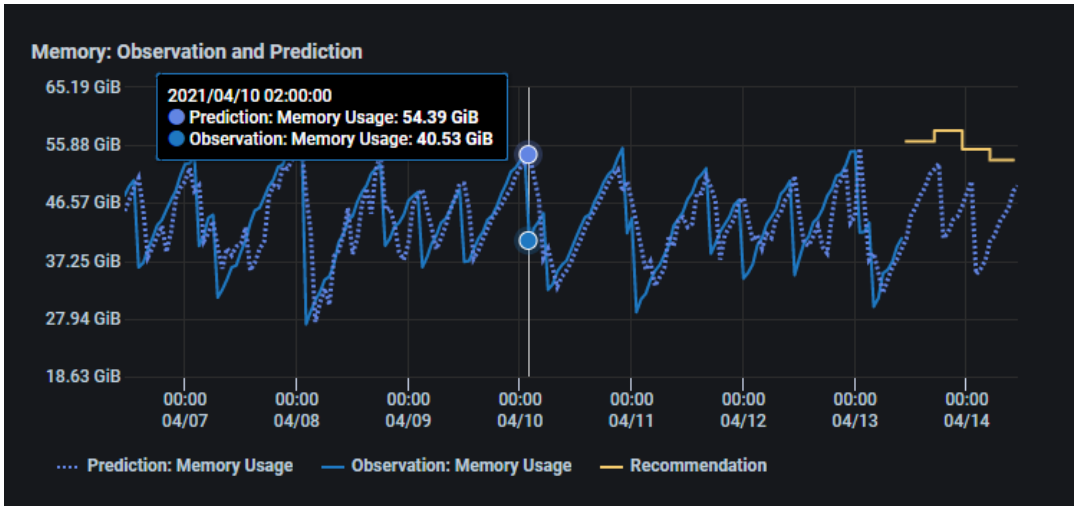
Search/Sort Information in Tables

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).



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.



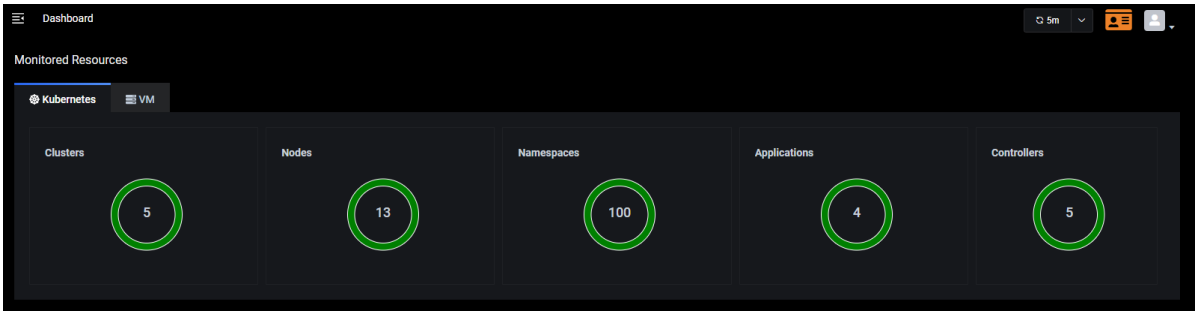
Related topics:

Common Administration Portal Functions

Dashboard

Dashboard

The *Dashboard* displays the number of monitored resources in Kubernetes clusters or VM clusters, as well as cluster and Kubernetes application workload predictions and recommendations. Select the *Kubernetes* or *VM* tab to view workload predictions and recommendations for resources in each type of cluster.



Cluster Workload Prediction

Select a cluster from the drop-down list and select the timeframe (daily, weekly, or monthly) to display CPU and memory observations, predictions, and recommendations.

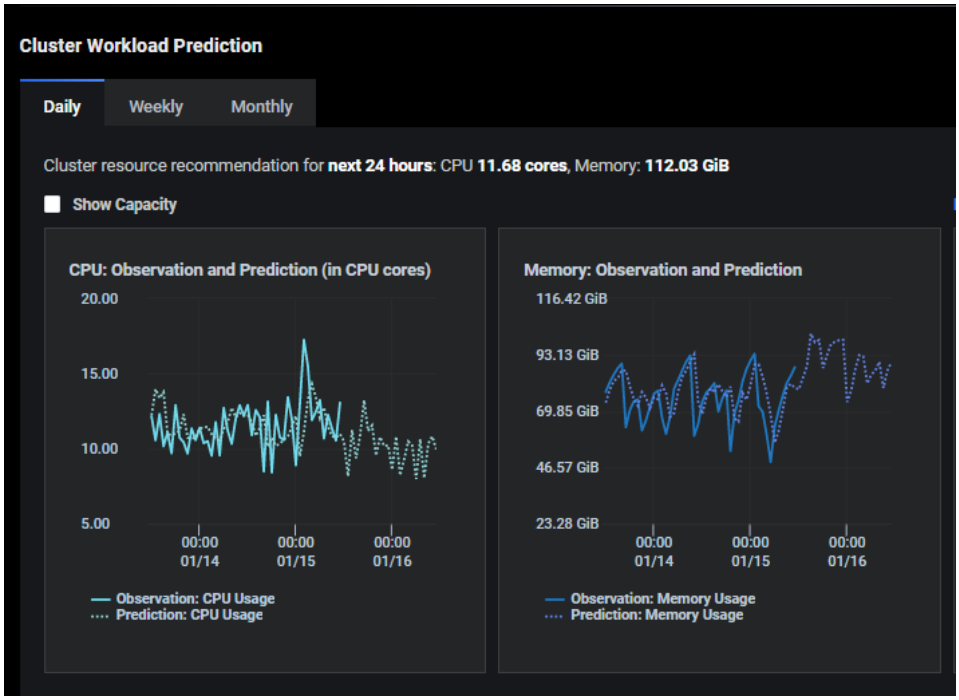
The first two charts display information for the selected cluster; the chart on the right toggles the display of information for nodes and namespaces of the selected Kubernetes cluster. For a VM cluster, the chart on the right displays information for the VMs in the selected cluster.

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

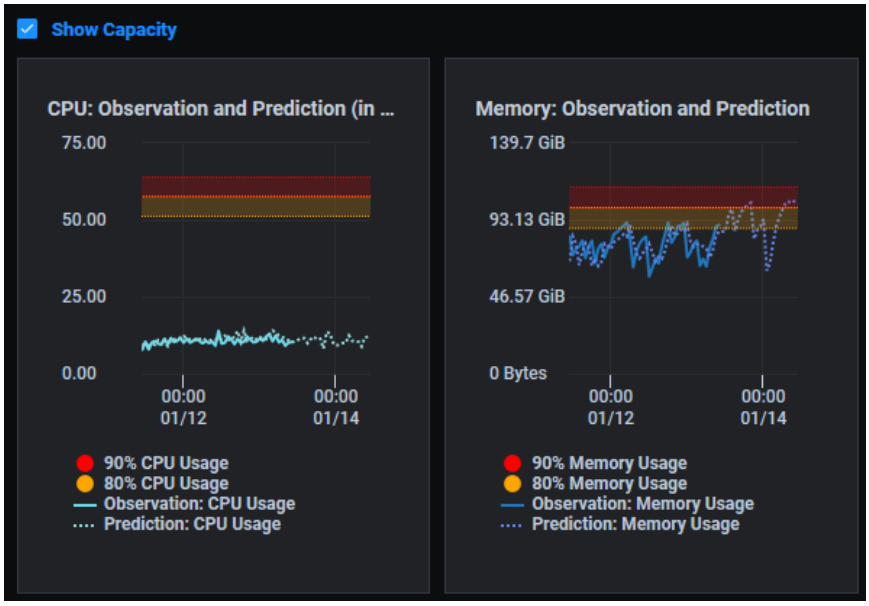
Cluster Charts

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 Node/Namespace chart accordingly.



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.

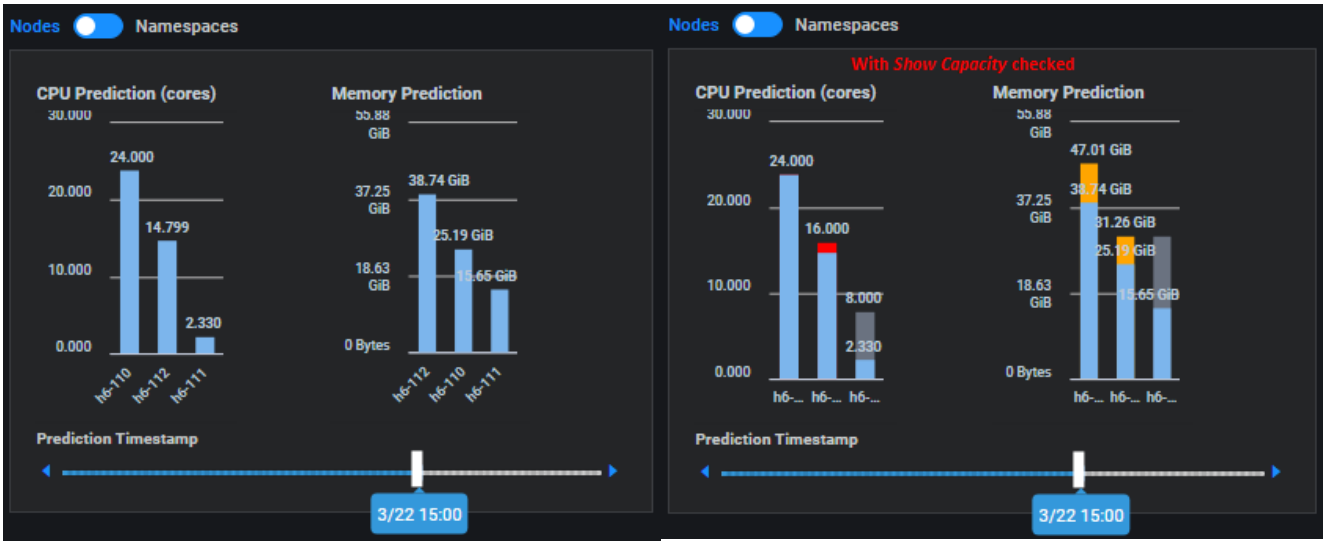


Node/Virtual Machine Chart

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.

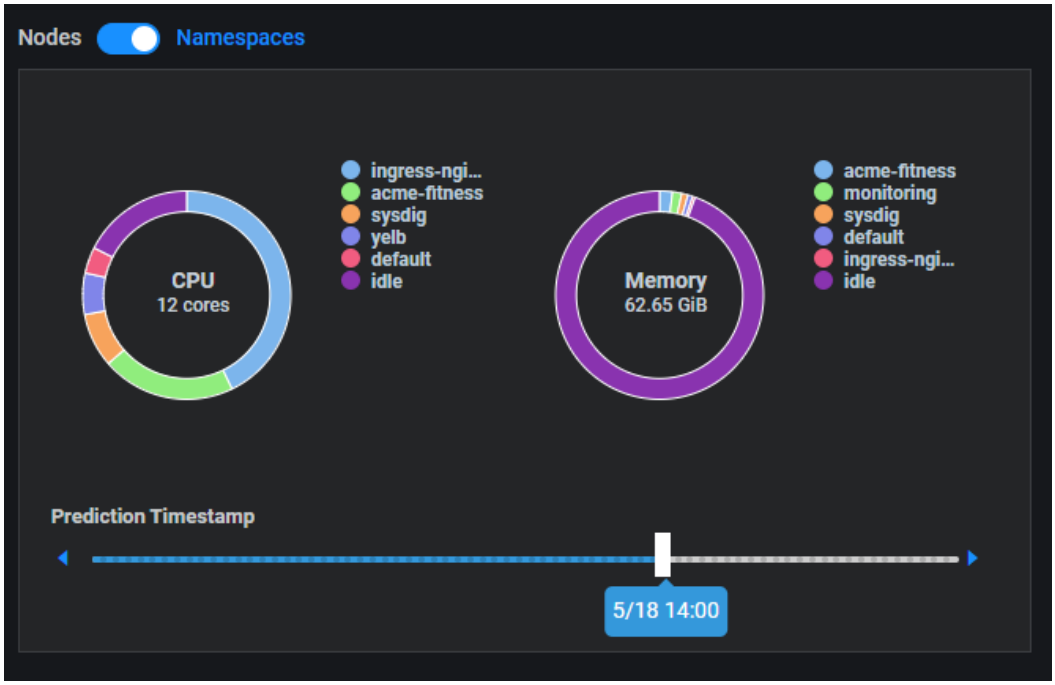


Namespace Chart (Kubernetes)

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
12 cores
ingress-ngi...
acme-fitness
sysdig
yelb
default
idle
Memory
62.65 GiB
acme-fitness
monitoring
sysdig
default
ingress-ngi...
idle
Prediction Timestamp
5/18 14:00

Application Workload Prediction (Kubernetes)

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

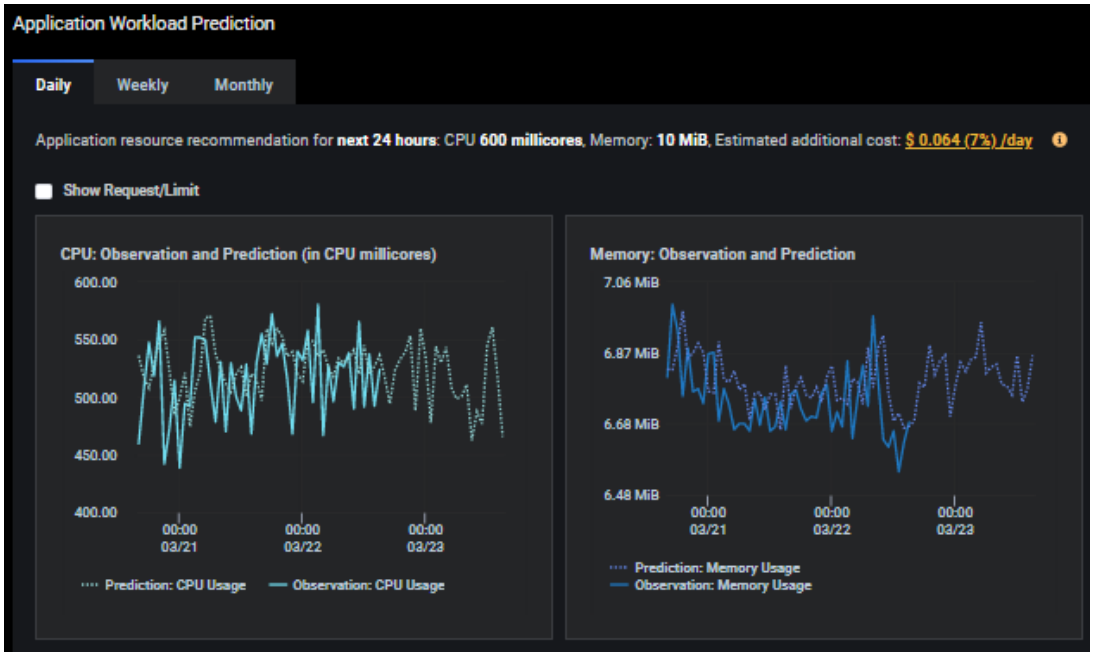
The first two charts display information for the selected application; the chart on the right displays information for controllers (generic application) or consumer groups (Kafka).

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 *Application Cost Analysis* page for more details.

Application Charts

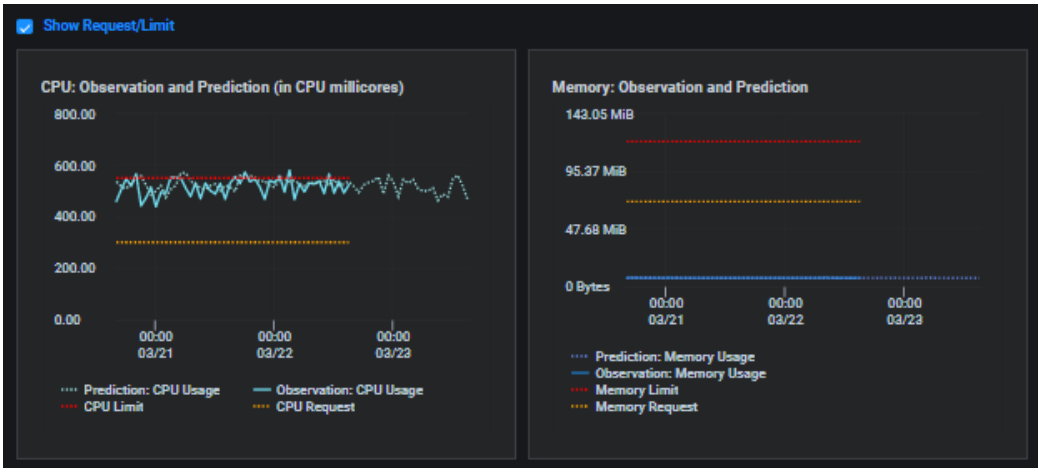
The application charts display CPU and memory observations, predictions, and recommendations for the application.

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.



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.

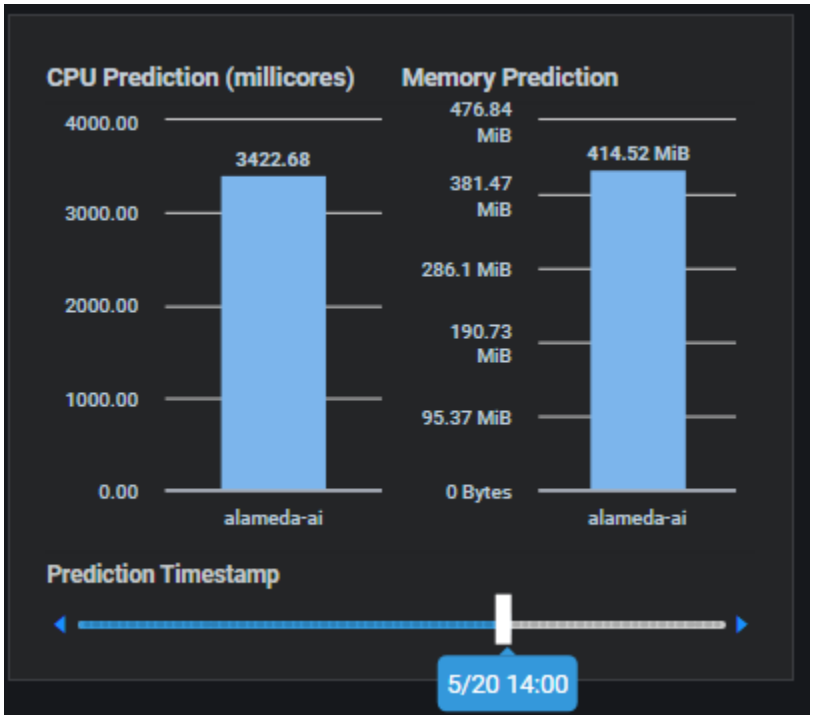
Generally, memory is a hard limit, but CPU is more *stretchable*. This is a useful way to see if you are over-provisioned or under-provisioned for your application.



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.



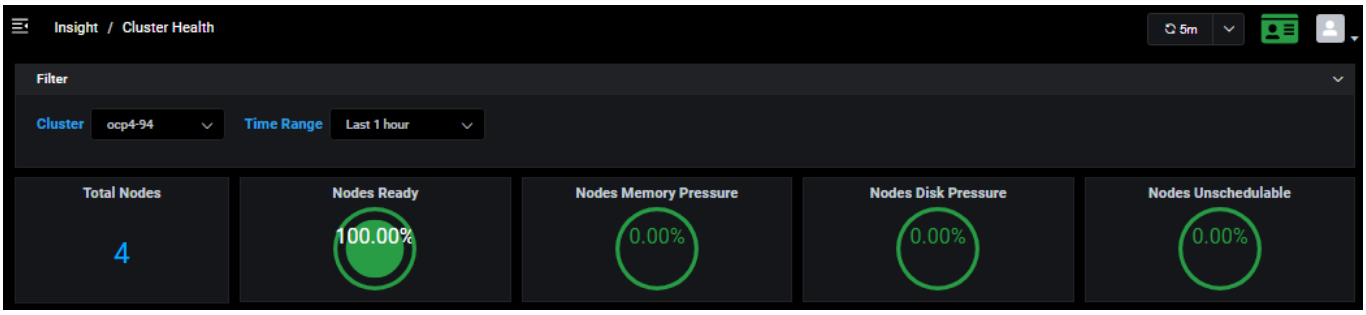
Related topics:

Common Administration Portal Functions
Refresh Statistics
License Status
User Functions
Search/Sort Information in Tables
Show/ Hide Information in Charts
Filter Panel

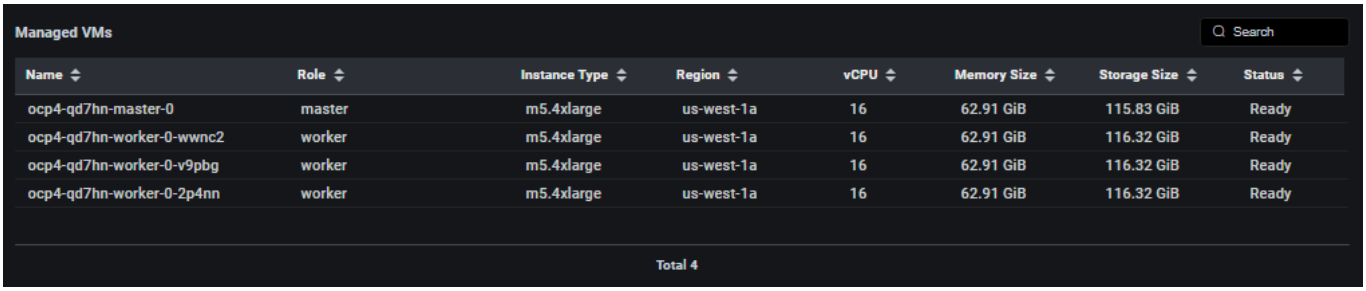
Insight - Cluster Health

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.



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.



Kubernetes cluster

VM cluster

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

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

Related topics:

Search/Sort Information in Tables

Terminology

Insight - 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.

Related topics:

Search/Sort Information in Tables

Show/ Hide Information in Charts

Terminology

Planning – Kubernetes or VM Workload Prediction

The *Workload Prediction* page displays actual CPU and memory usage observations, predicted usage, and recommendations.

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

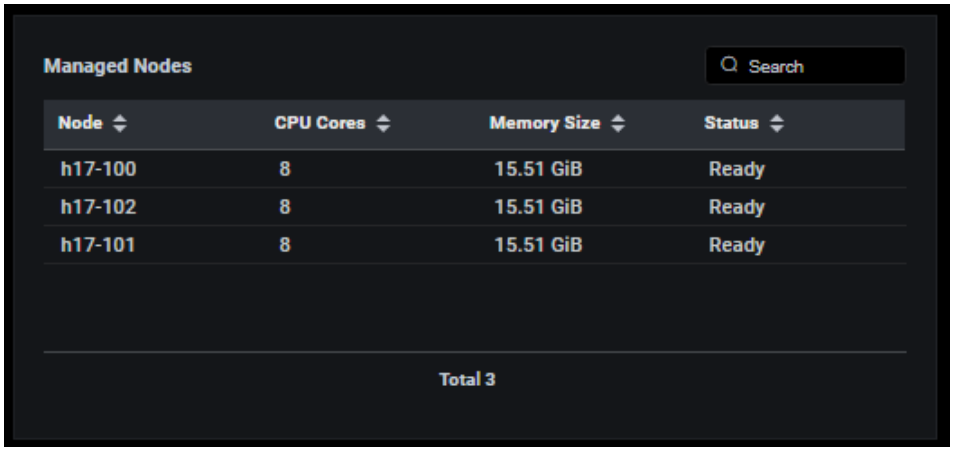
For VM, Federator.ai monitors resource usage for monitored clusters and VMs and provides workload prediction and recommendations at each level.

With the 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.

In the *Filter* panel, select the level of information you want to display. When you select a Kubernetes namespace, the namespace status will be displayed.

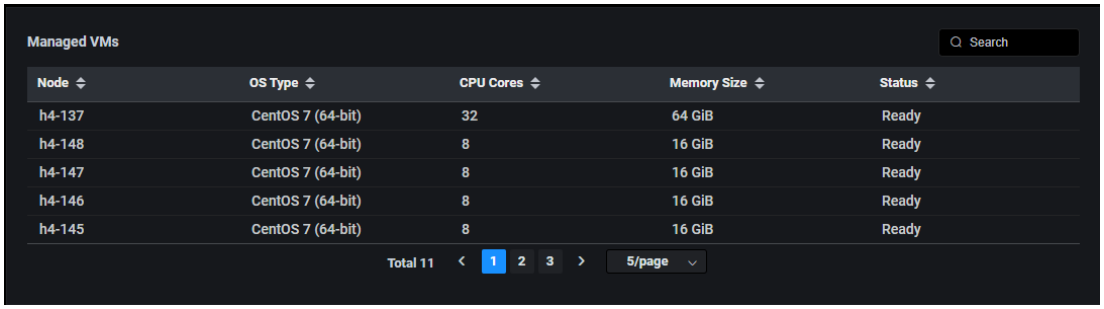
Managed Nodes Table (Kubernetes)

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.



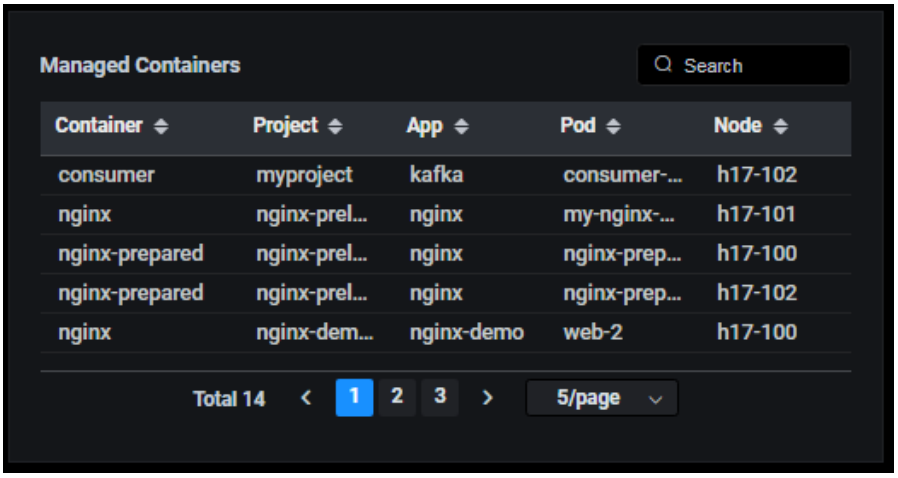
Managed VMs Table (VM)

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 Containers Table (Kubernetes)

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



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:

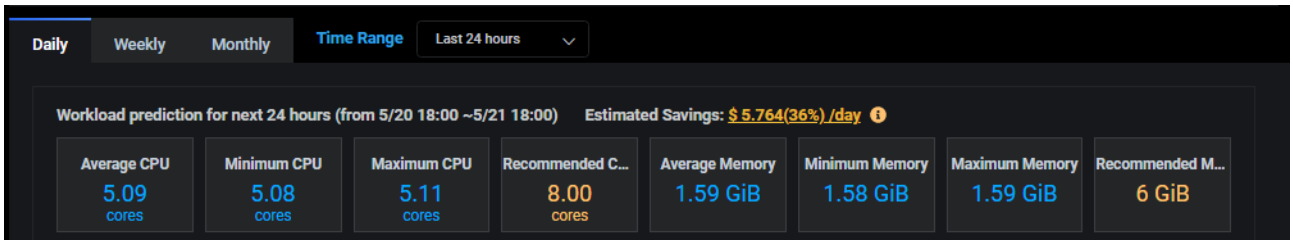
- Daily – Predicts CPU and memory usage every hour for the next 24 hours.
- Weekly – Predicts CPU and memory usage every 6 hours for the next 7 days.
- Monthly – Predicts CPU and memory usage every day for the next 30 days.

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

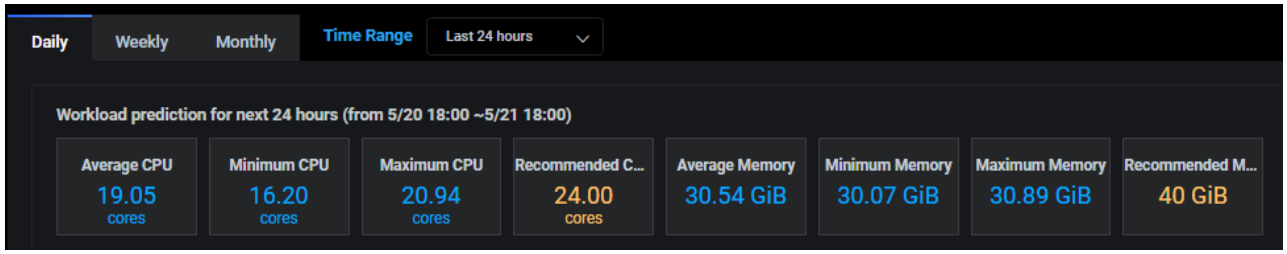
Workload Prediction Table

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:

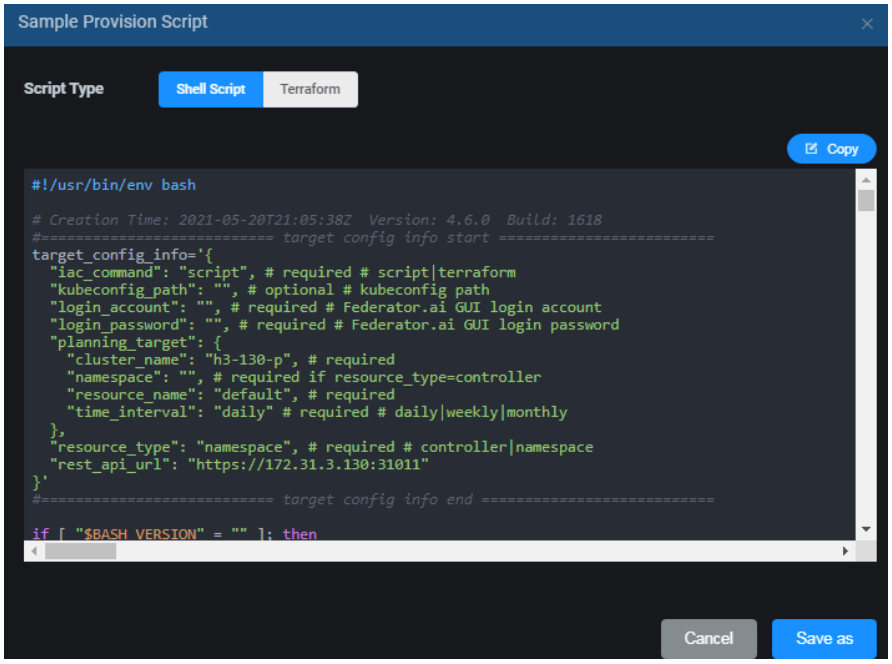


Kubernetes:



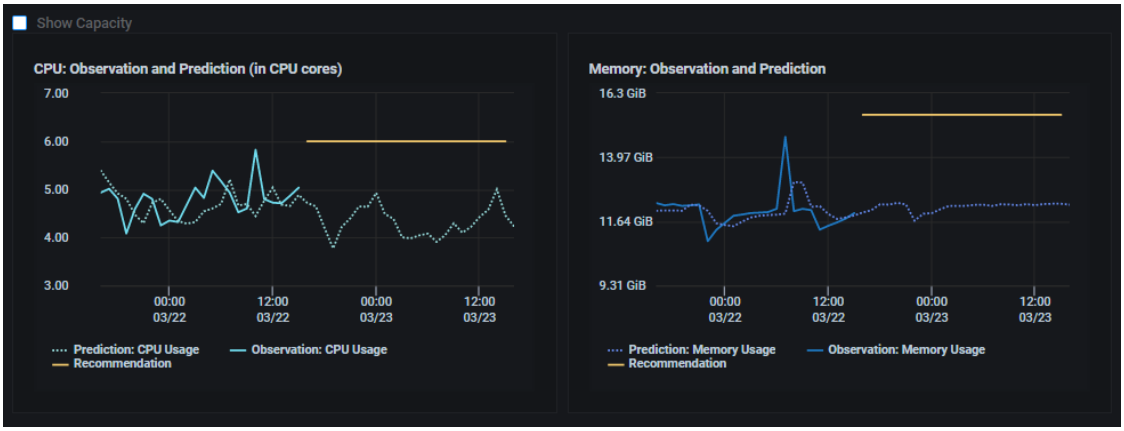
If you are displaying information for a Kubernetes namespace and the status is anything but *Monitoring*, this section will provide more information. For example, you will see the message, "Workload prediction is not configured for this namespace" or "Not enough information for predictions" for newly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



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.

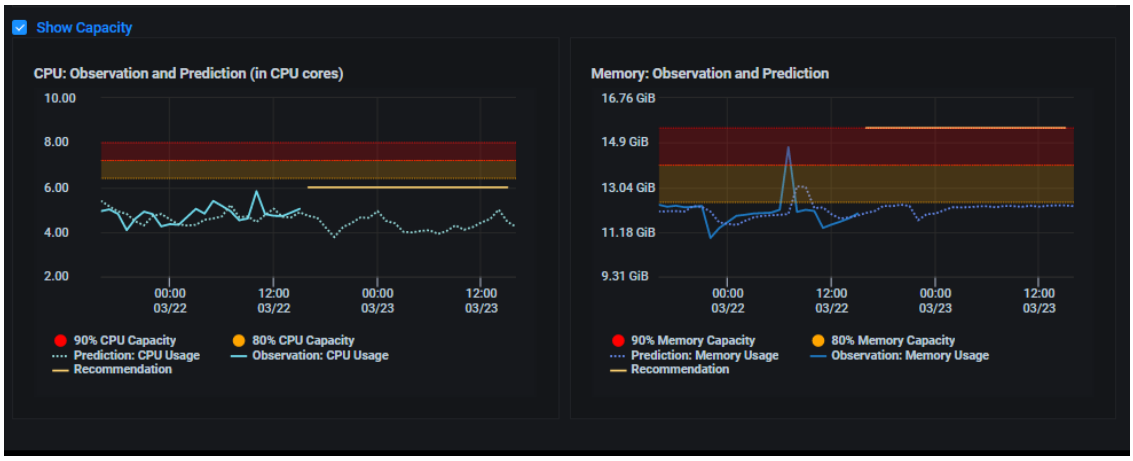
Workload Observation and Prediction Charts



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.



Related topics:

Terminology
Search/Sort Information in Tables
Show/Hide Information in Charts
Cluster Configuration
Auto Provisioning

Recommendation - HPA Recommendation (Kubernetes)

The *HPA Recommendation* 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 auto-scaled 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.

Related topics:

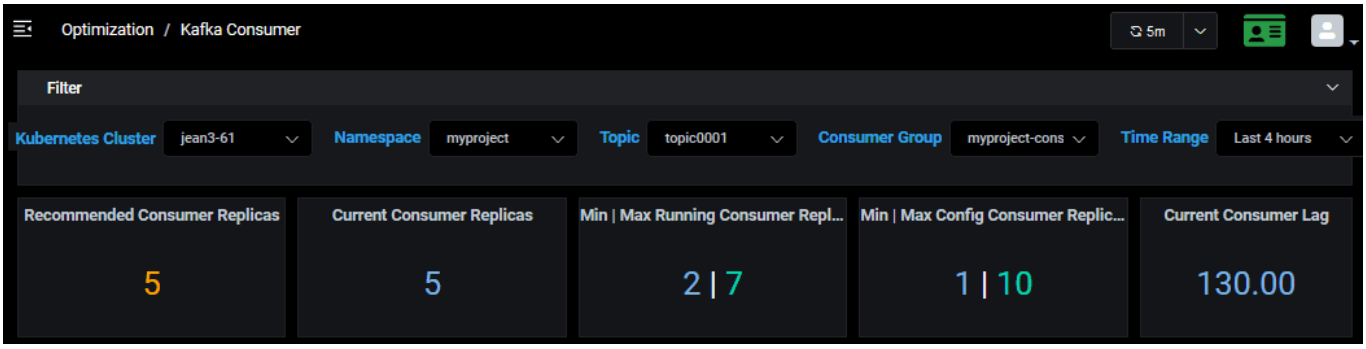
Common Administration Portal Functions
Terminology
Configuration
Search/Sort Information in Tables
Show/Hide Information in Charts
Configure Applications

Optimization – 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) offers a better indicator of workload and performance, and is monitored by Federator.ai. Using message production rate predictions, Federator.ai auto-scales the number of Kafka consumer pods to fit the workload and optimize performance.

If you have configured Federator.ai to monitor and auto-scale 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 auto-scale consumers for each individual topic/consumer group. Refer to the Add an Application section for information about how to configure Kafka consumer monitoring and auto-scaling.

In the *Filter* panel, 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.



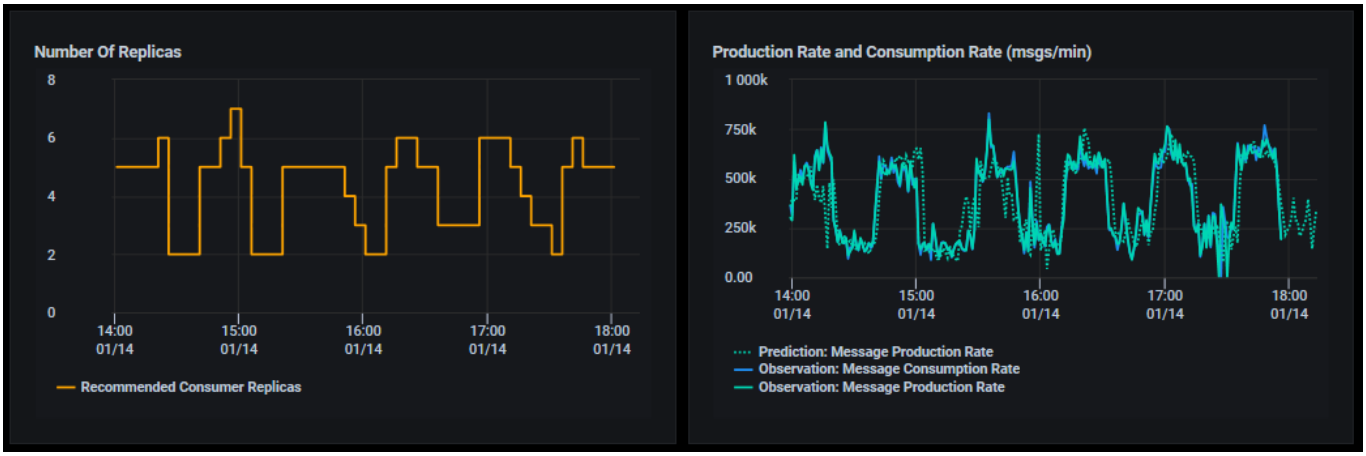
The number of recommended and current consumer replicas is displayed along with the minimum and maximum number of running and configured replicas and the current consumer lag.

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).

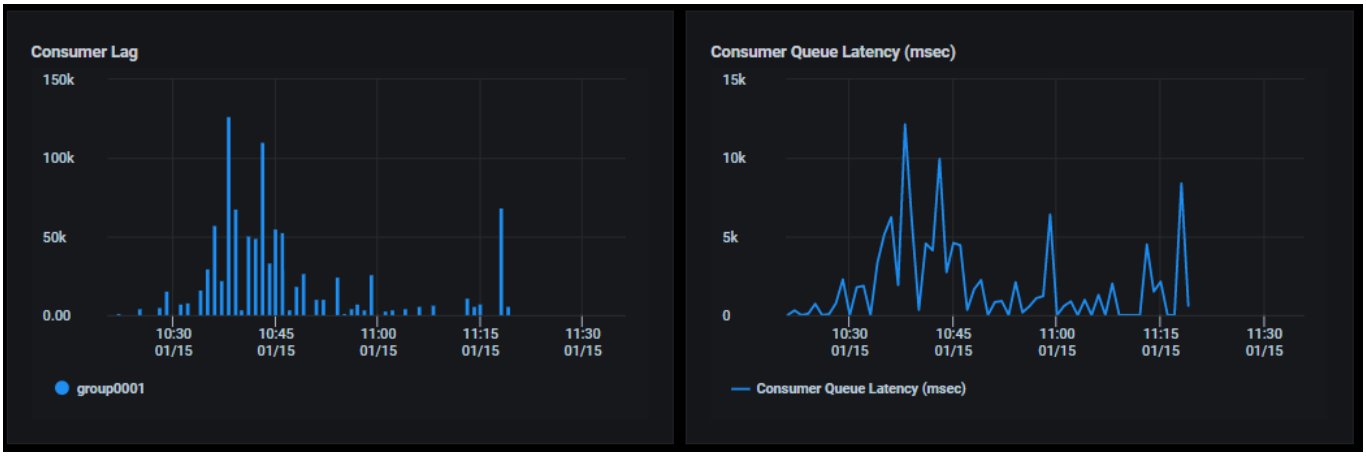


Consumer Lag Chart

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.

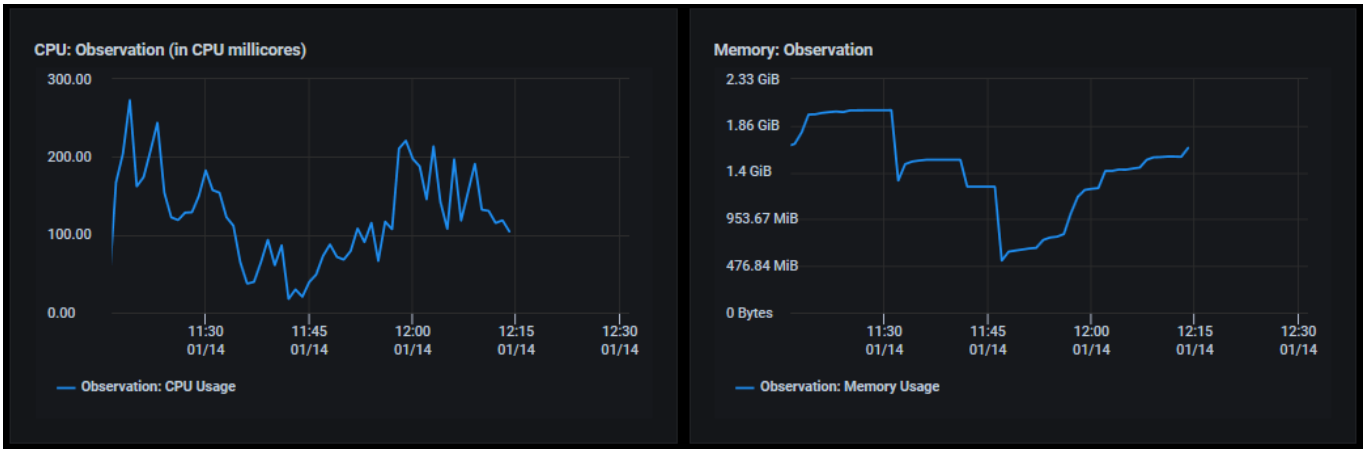
Consumer Queue Latency Chart

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:

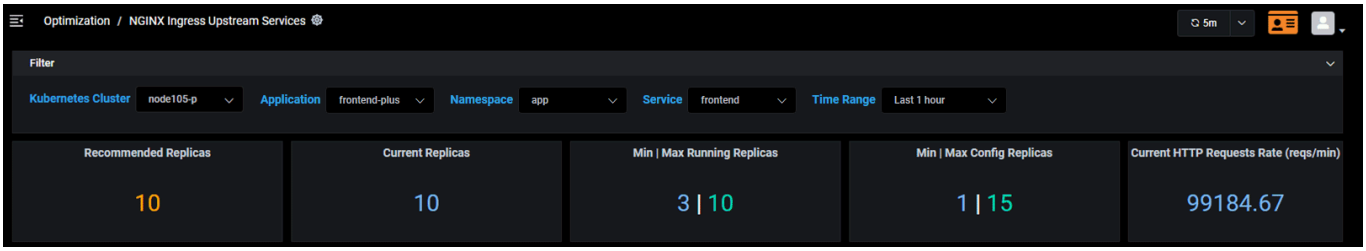
Common Administration Portal Functions
Terminology
Show/Hide Information in Charts
Setup Wizard
Configure Applications

Optimization – 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 auto-scales the number of services to fit the workload and optimize performance.

If you have configured Federator.ai to monitor and auto-scale 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 auto-scale for each individual service. Refer to the Add an Application section for information about how to configure Ingress upstream services for auto-scaling.

In the *Filter* panel, 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.



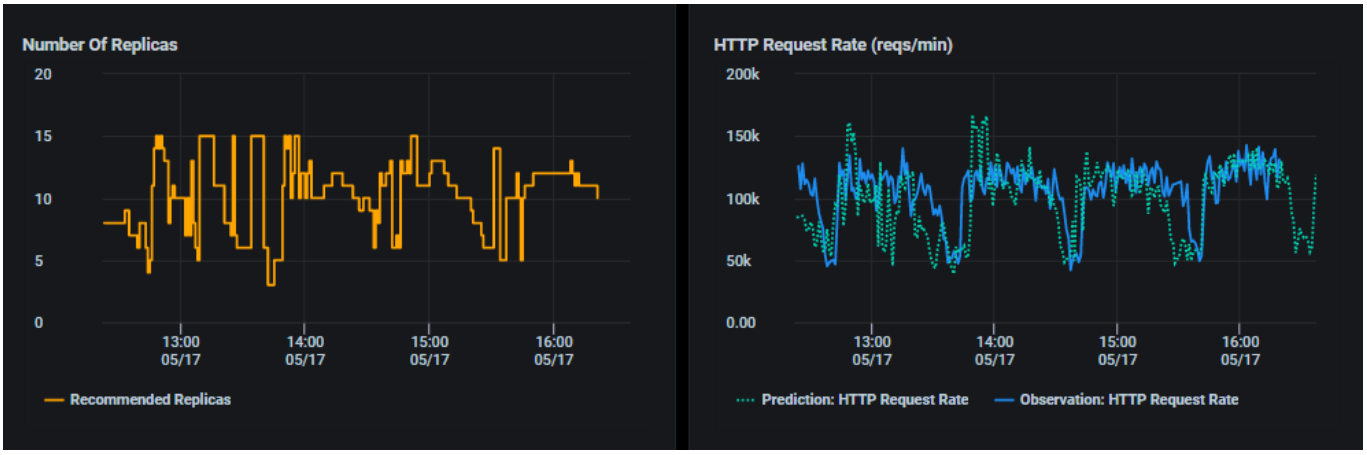
The number of recommended and current replicas is displayed along with the minimum and maximum number of running and configured replicas and the current HTTP request rate.

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).

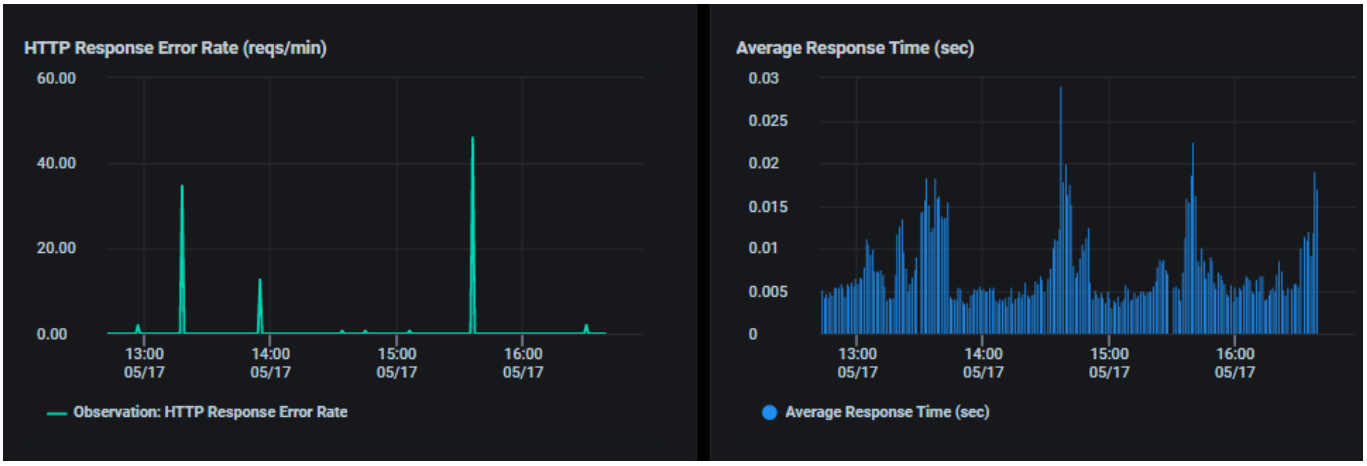


HTTP Response Error Rate Chart

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.

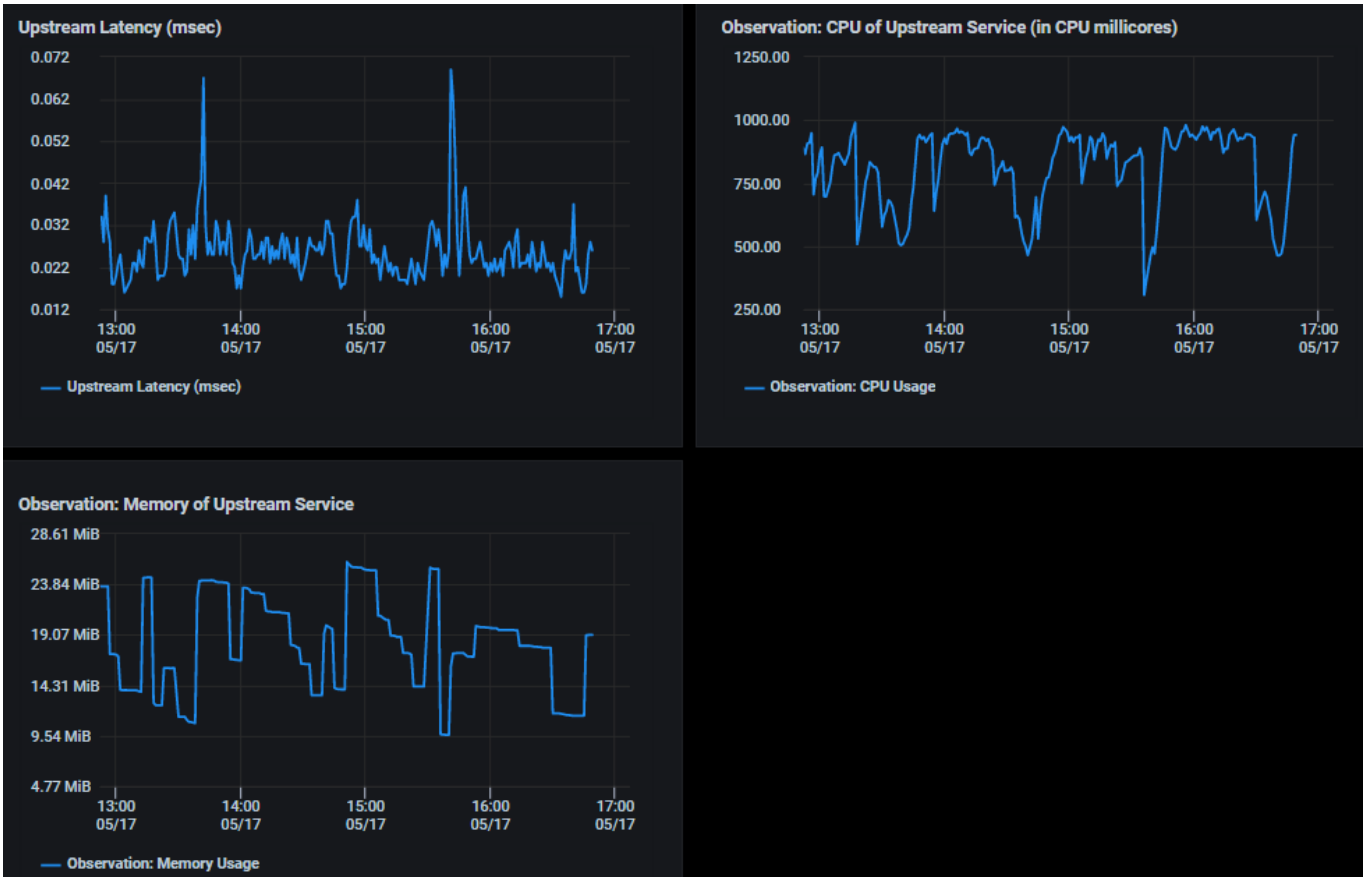


Upstream Latency Chart

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.

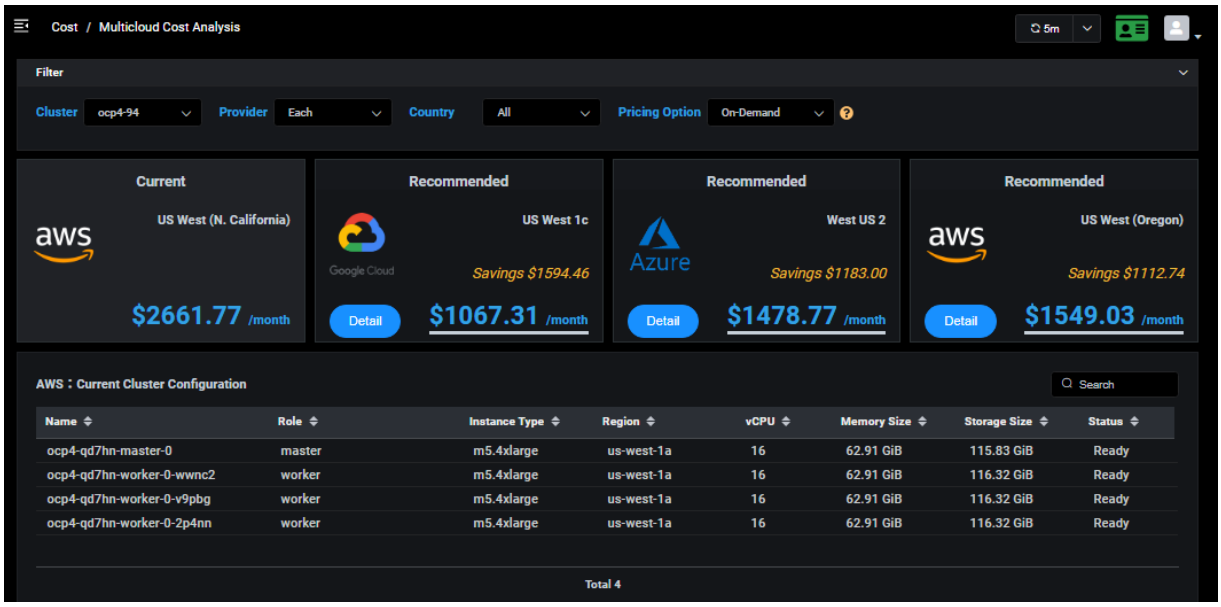


Related topics:

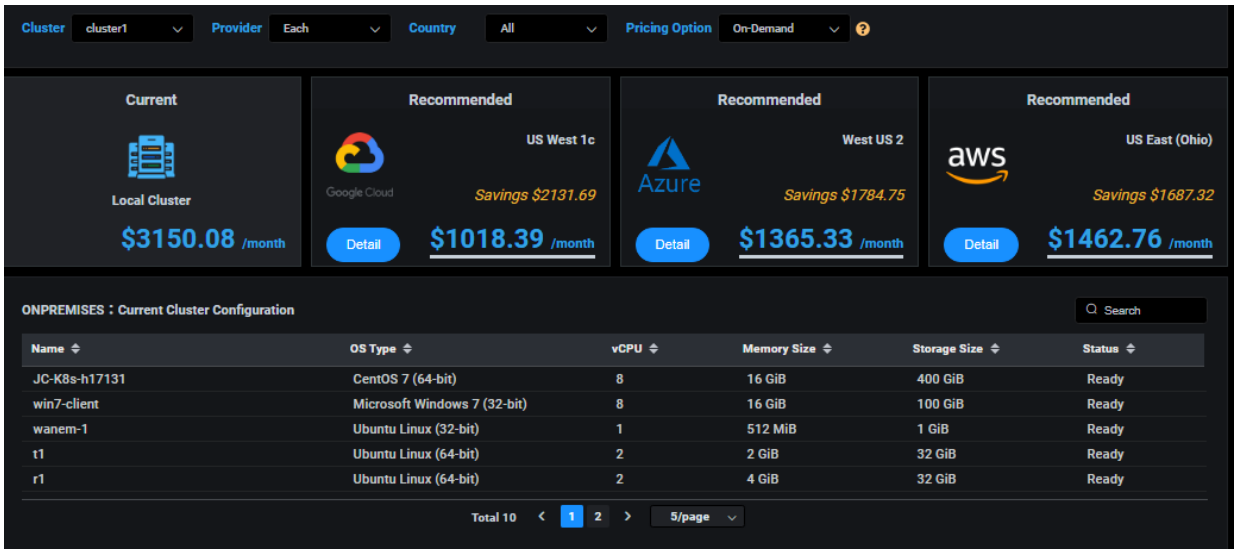
Common Administration Portal Functions
Terminology
Show/Hide Information in Charts
Setup Wizard
Configure Applications

Cost – Multi-cloud Cost Analysis (Kubernetes and VM Clusters)

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.



Spending with current cloud provider



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 or operating system (OS) for your local cluster, 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.

CPU and Memory Utilization Charts

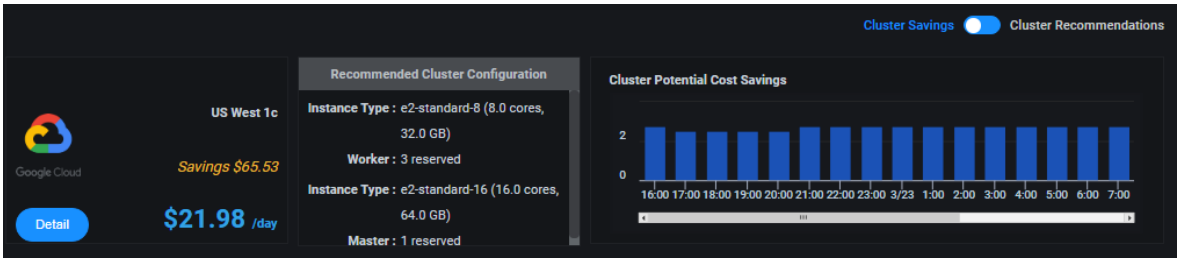
The *CPU Utilization* and *Memory Utilization* charts display utilization information for all nodes/VMs in the selected cluster based on your daily, weekly, or monthly workload.

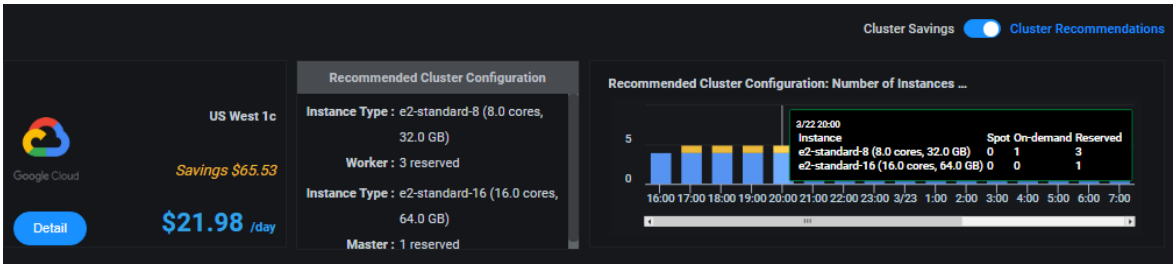
The solid lines represent the observed actual usage while the dotted lines show the historical and future predicted usage. The predicted usage is used in the analysis for cluster configuration recommendations.

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

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 *CPU Utilization* and *Memory Utilization* charts, the recommendation 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.





Use the toggle to display *Cluster Savings* or *Cluster Recommendations*.

Cluster Savings shows how much you can save by following the recommended configuration.

Cluster Recommendations displays a *Time Series* chart that shows the recommended instances (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. The recommendations for each data point include the number of on-demand, reserved, or spot instances (for Kubernetes), depending on the pricing option selected.

For daily predictions, data is displayed in the charts for each hour. For weekly predictions, data is displayed for every six hours. For monthly predictions, data is displayed for each day.

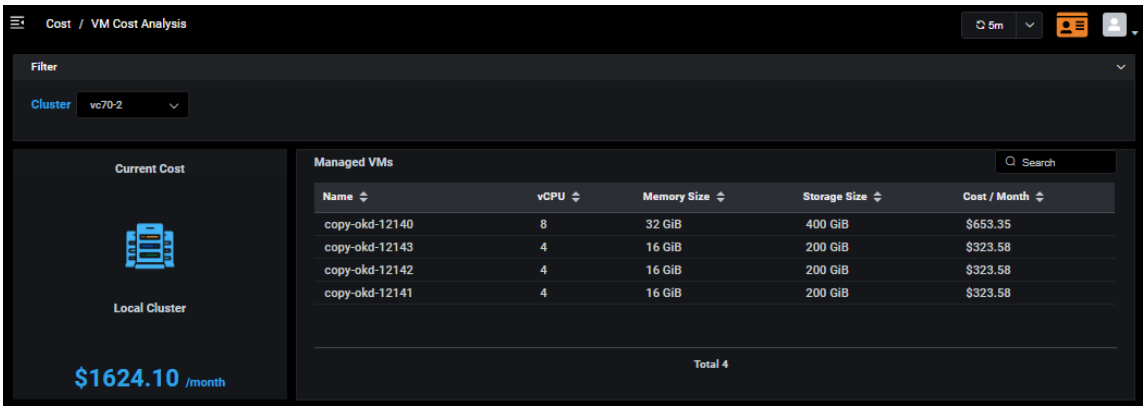
In the case of a VM cluster, the recommended cluster configuration will display the recommended instance type for each individual VM.

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
Price Books

Cost – VM Cost Analysis (VM Clusters and VMs)

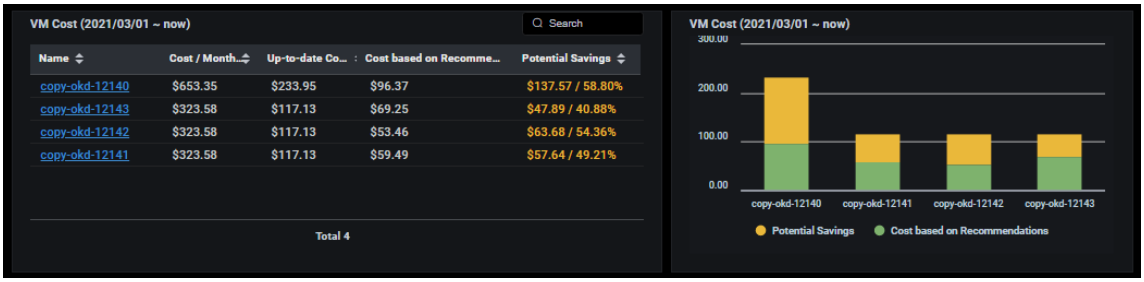
The *VM Cost Analysis* page displays your cost for an individual cluster and for each managed VM for the current month and shows your potential cost/savings based on usage recommendations for the next 30 days.



The *Current Cost* box displays the current monthly cost for the cluster and the *Managed VMs* table displays a list of managed VM in the cluster with the number of CPU cores, memory size, storage size, and monthly cost for each VM.

VM Cost

This section shows the usage and cost for each VM along with potential savings based on recommendations. The recommendations used in the calculations can be viewed on the *Planning / VM Workload Prediction* page. Be sure to check *Show Capacity* to see the maximum CPU and memory usage limits for the VM, which shows if resources are over-provisioned.



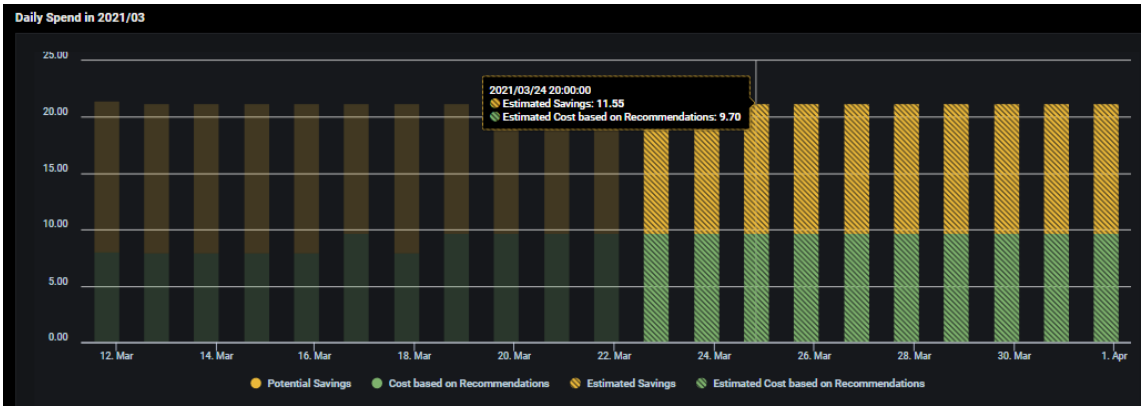
The table on the left shows the full monthly cost, the cost from the beginning of the current month to now, what that cost would be based on the recommendations, and the potential savings for each VM. Click on a VM to update the charts below.

The bar chart on the right presents a graphical view of the cost based on the recommendations and the potential savings from the beginning of the current month to now. The green area of each bar represents your monthly cost based on the recommendations and the yellow represents your estimated monthly potential savings based on the recommendations.

Daily Spend Chart

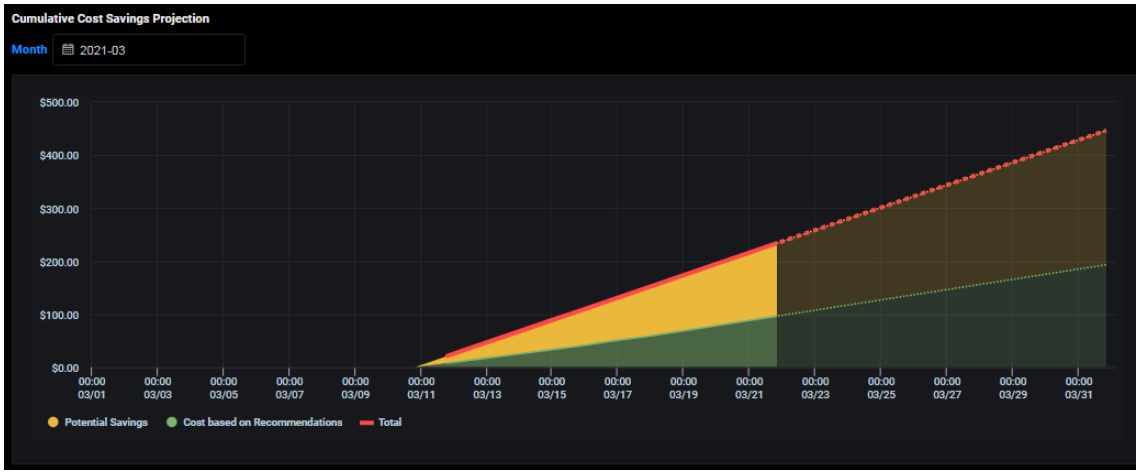
This chart displays your daily spending for the selected VM from the start of the month to end of the month. The green area of each bar represents your daily cost based on the recommendations (actual or estimated for the future) and the yellow represents your daily potential savings based on the recommendations.

Move your cursor to a specific day to see detailed information.



Cumulative Cost Savings Projection Chart

This chart displays the cumulative savings for the VM selected from the start of the month and going forward. The default is the current month.



The green section represents the cost based on the recommendations, the yellow section represents estimated potential savings, and the red dotted line represents the total.

Related topics:

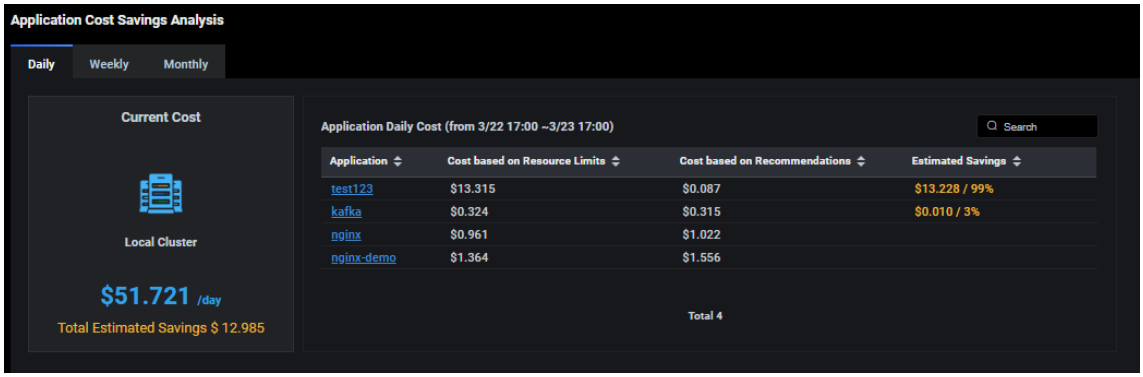
Multicloud Cost Analysis
Workload Prediction Page
Workload Observations and Predictions
Common Administration Portal Functions

Terminology
Search/Sort Information in Tables
Show/Hide Information in Charts
Price Books

Cost – Application Cost Analysis (Kubernetes Applications)

The *Application Cost Analysis* page displays your cost for each monitored Kubernetes application in a selected cluster and shows your potential cost/savings based on usage recommendations. The recommendations used in the calculations can be viewed on the *Planning / Kubernetes Workload Prediction* page.

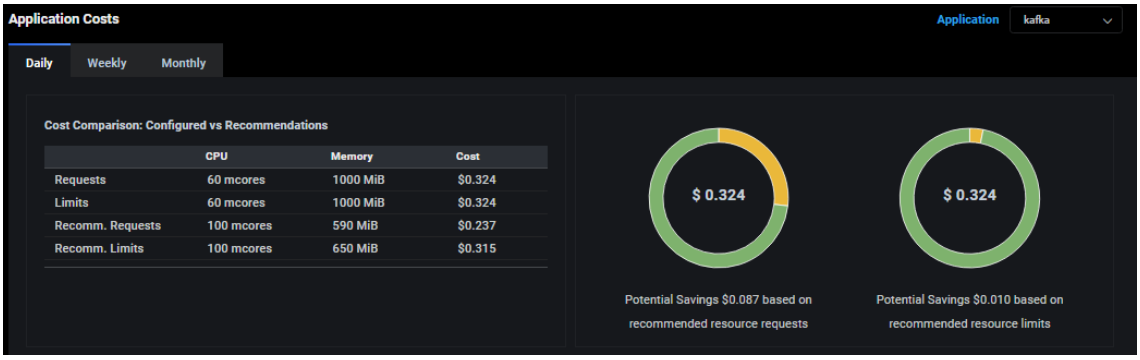
Application Cost Savings Analysis



The *Current Cost* box displays the current monthly cost for the cluster along with the savings for all applications. The *Application Daily Cost* table displays the cost based on current resource limits, what that cost would be based on the recommendations, and the potential savings for each application in the cluster. Click on an application to update the charts below.

Information can be viewed daily, weekly, or monthly.

Application Costs



Select an application and day/week/month to display a cost comparison between your current Kubernetes application CPU and memory requests and limits and the recommended requests and limits.

The charts on the right present a graphical view of the difference between the current and recommended requests and limits as well as the estimated savings for the selected application. Move your cursor over the chart to see detailed information.

Cost Trends

This chart displays the trends over time. Use the sliders to see requests or limits for the current selected time period (day/week/month) or cumulatively through the end of the selected time period. The total savings for the period are shown above the chart.

The green area of each bar represents your cost based on the recommendations (actual or estimated for the future) and the yellow represents your potential savings based on the recommendations.

Move your cursor over the chart to see detailed information for a data point.

If auto provisioning is used, you will see the adjusted costs and actual savings after auto provisioning is applied.

In this example, the dotted yellow line is the original cost before auto provisioning was applied and the blue line is the cost calculation based on the resource limit. You can see how it drops after the recommendations are applied.

Related topics:

Multicloud Cost Analysis
Cost Allocation Kubernetes Namespaces
Workload Prediction Page
Workload Observations and Predictions
Common Administration Portal Functions
Terminology

Search/Sort Information in Tables
Show/Hide Information in Charts
Auto Provisioning
Price Books

Cost - Cost Allocation (Kubernetes Namespaces)

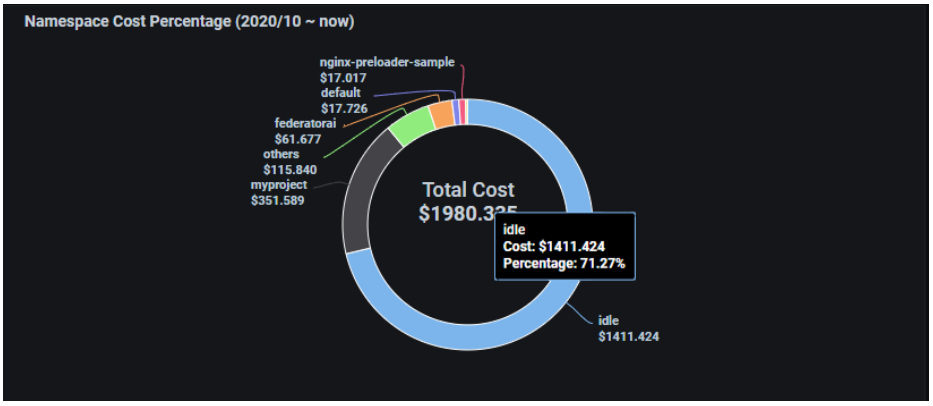
The *Cost Allocation* page provides an in-depth cost analysis for individual Kubernetes namespaces of a cluster hosted by the current cloud service provider. With the information displayed on this page, you can compare the cost of each namespace and determine the estimated cost of each namespace based on the workload prediction. You can also see the percentage of cost of the system spent while the system is idle. A high percentage of cost while the system is idle indicates that the cluster is over-provisioned.

Current Cluster Configuration

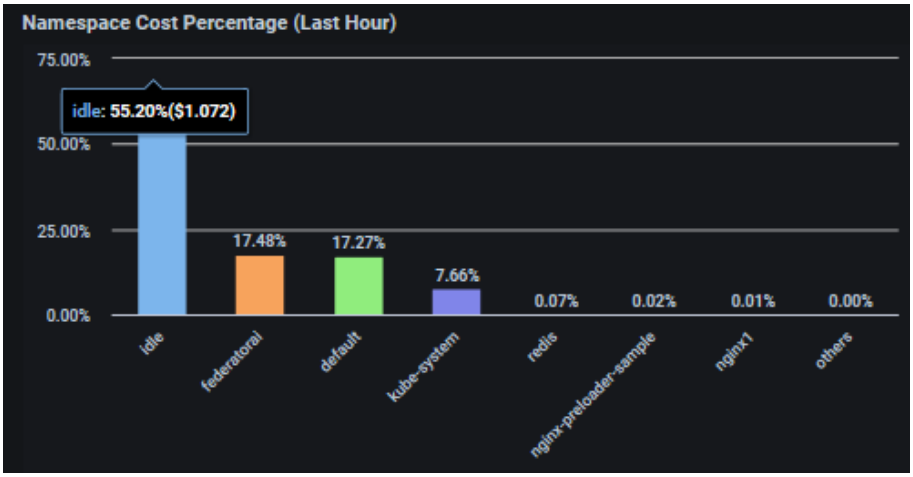
The existing configuration with your provider is displayed. This includes the role (master, worker), instance types being used, region, number of CPUs, memory size, and storage space.

Namespace Charts

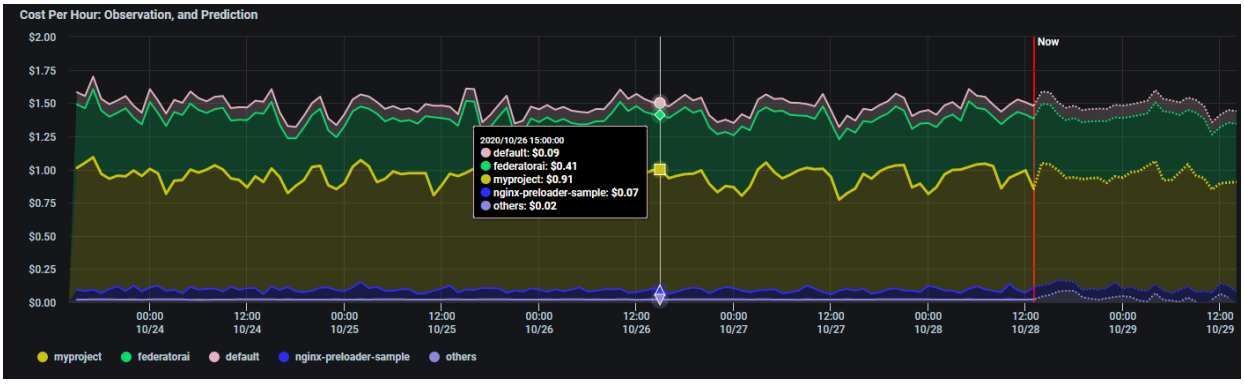
The *Namespace Cost Percentage (month)* chart displays this month's actual percentage of cost to-date for each namespace and when the system is idle. The *others* tag represents the unmonitored namespaces in a cluster. Move your cursor over each section to show the percentage of total cost for a namespace or when the system is idle.



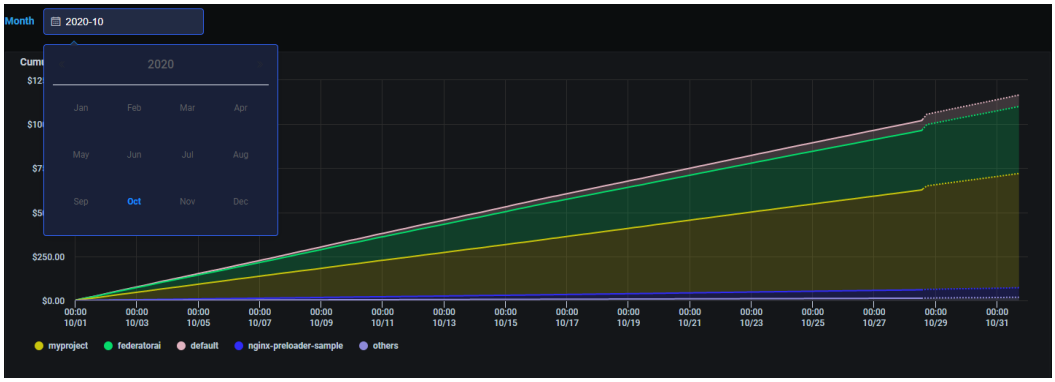
The *Namespace Cost Percentage (Last Hour)* chart displays the actual cost for each namespace and when system was idle during the past hour. The *others* tag represents the unmonitored namespaces in a cluster. Move your cursor over each bar to show the cost of a namespace or when system is idle.



The *Cost Per Hour* chart displays hourly costs for each namespace during the past five days and the predicted hourly cost for the next 24 hours. The solid lines represent the observed actual cost while the dotted line shows the future predicted cost based on the namespace resource usage prediction. The *others* tag represents the unmonitored namespaces in a cluster. Move your cursor to a specific date/hour to see the detailed cost of each namespace.



The *Cumulative Cost Projection* chart displays a cumulative cost from the beginning of the month and a cost projection for this month for each namespace. The solid lines represent the observed actual cost while the dotted lines show the future predicted cost based on the namespace resource usage prediction. The *others* tag represents the unmonitored namespaces in a cluster. Click on the calendar icon to choose the cost analysis for a specific month.



Related topics:

Multicloud Cost Analysis
Application Cost Analysis
Common Administration Portal Functions
Terminology
Search/Sort Information in Tables
Show/Hide Information in Charts
Price Books

Configuration - Clusters

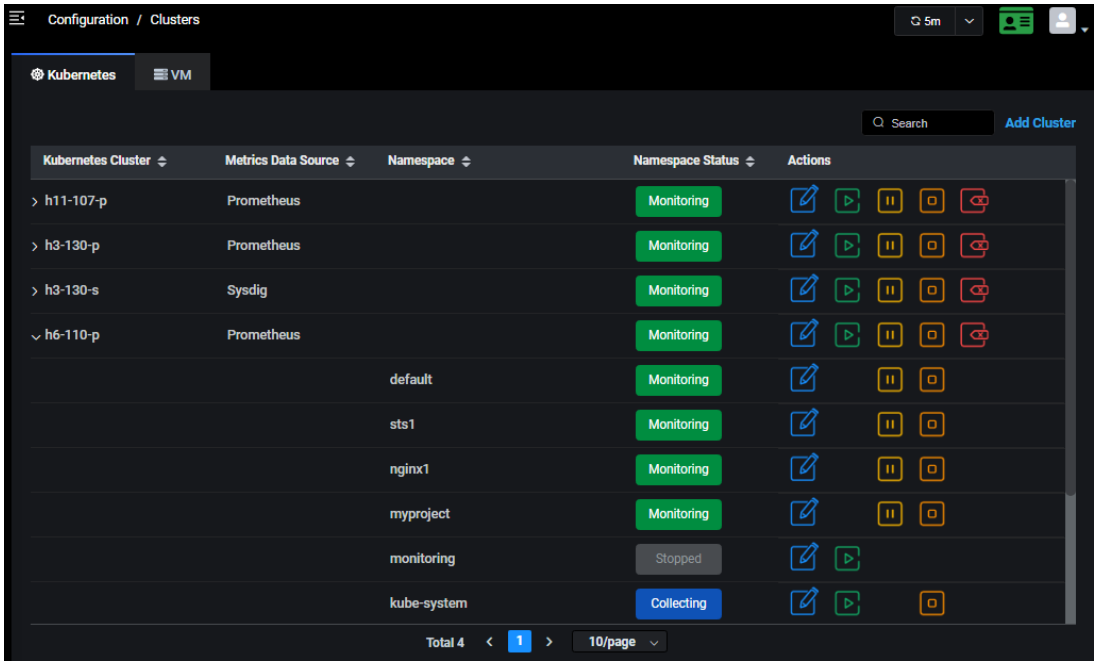
The *Clusters* page displays all Kubernetes or VM clusters being monitored by Federator.ai. You can add, manage, and remove clusters from this page.

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. A namespace will also collect data if it was added into a cluster that does not automatically monitor for new namespaces.

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.



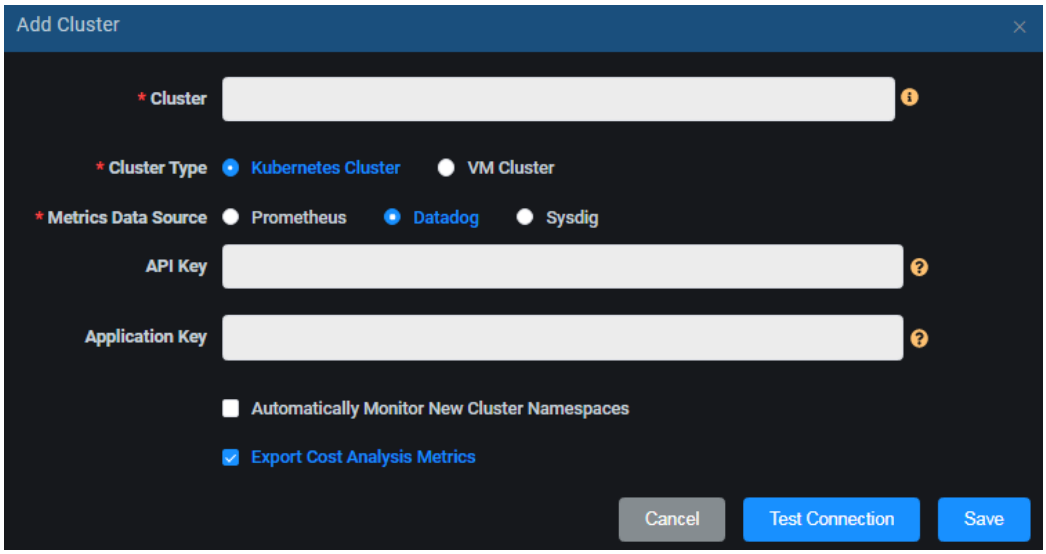
In addition to adding a cluster, you can perform the following functions for existing clusters and namespaces:

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*.



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. 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
  ```

Automatically Monitor New Cluster Namespaces – Specify if you want to automatically monitor new namespaces in this cluster. If unchecked (default), you must manually specify which namespaces to monitor.

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.

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:

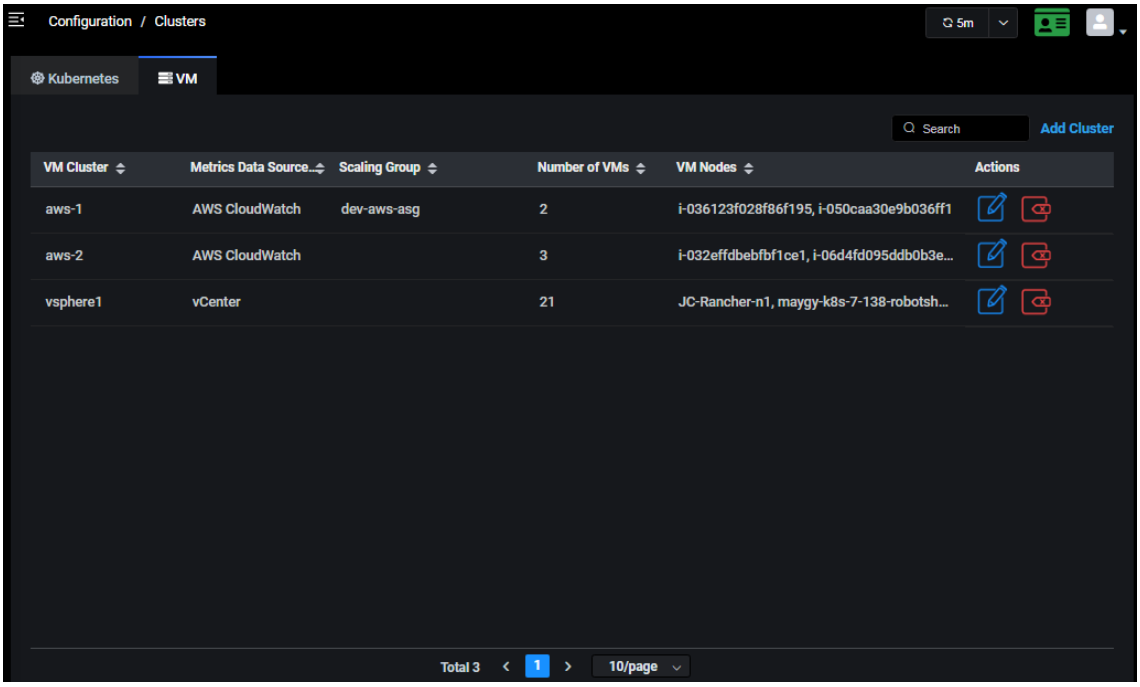
- Edit cluster settings – Manage monitoring of new namespaces and export of cost analysis metrics (Datadog), configure Prometheus Federation (Prometheus). To do this, click the *Edit Cluster* icon.
- Edit namespace settings – Change monitoring status and configure auto provisioning for the namespace. 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.

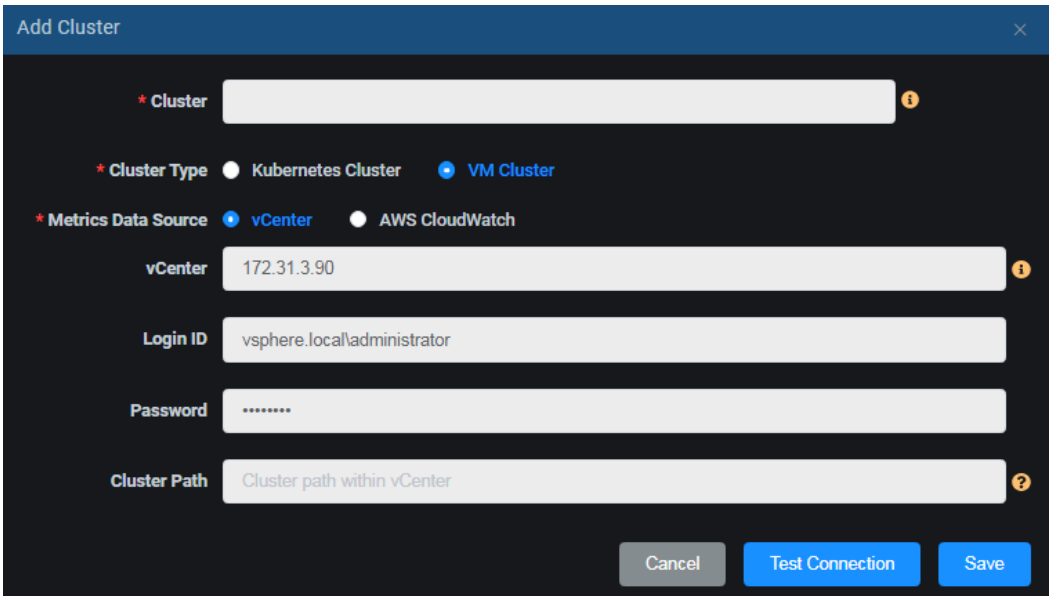


In addition to adding a cluster, you can perform the following functions for existing clusters:

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*.



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.

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.

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

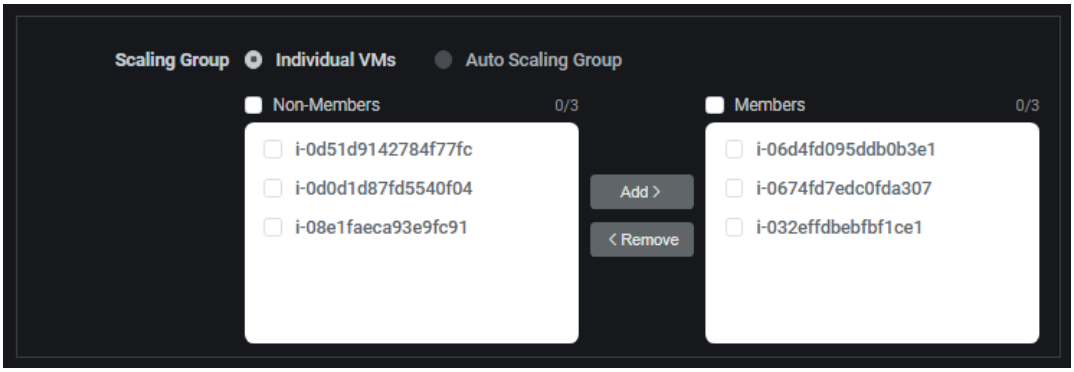
3. Click *Save* when the system can connect to the cluster.

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.

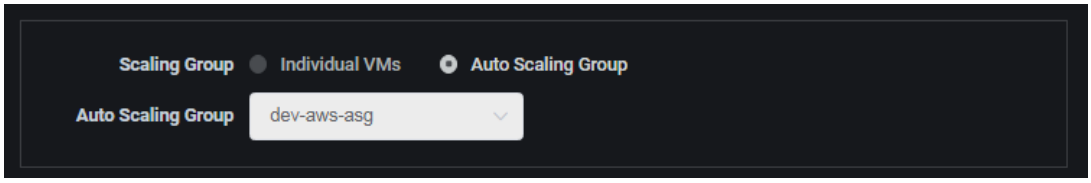
Manage VM Clusters

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.



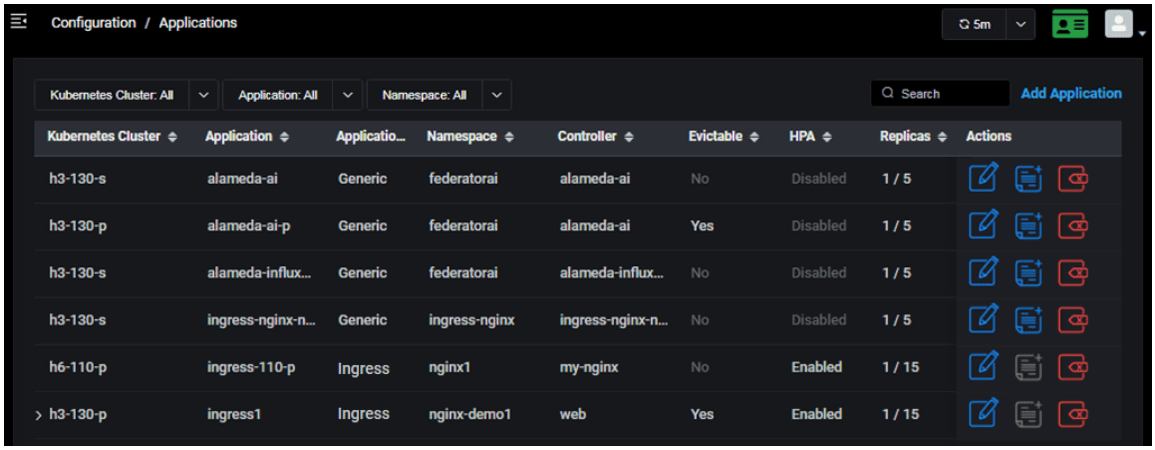
If you have Auto Scaling groups configured in AWS, select a group to monitor.



- 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.

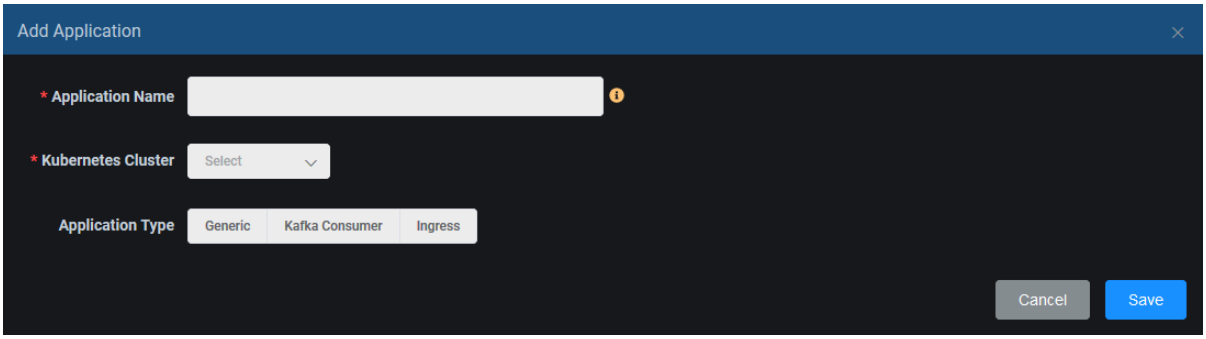


In addition to adding an application, you can perform the following functions for existing applications:

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

Add an Application

1. On the *Configuration* / *Applications* page, click *Add Application*.



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.

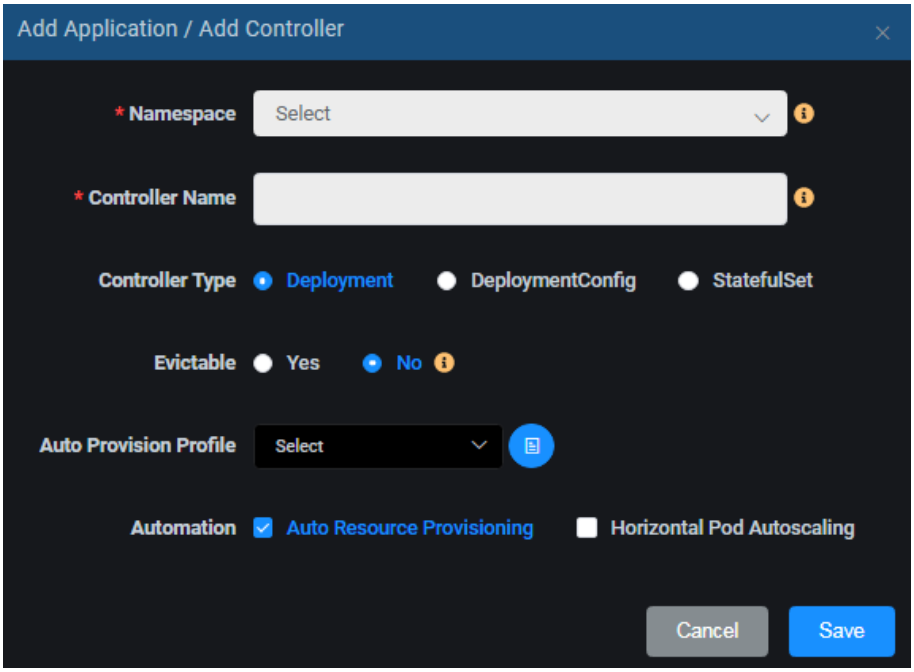
Kubernetes Cluster – Select an existing cluster where this application resides.

Application Type – Select the type of application:

- *Generic* - You can configure auto provisioning or HPA for controllers.
- *Kafka Consumer* – You can configure HPA for consumer groups.
- *Ingress* – You can configure HPA for upstream HTTP services using Ingress NGINX or NGINX Ingress Controller. Ingress NGINX is the community supported version and NGINX Ingress Controller is a commercial version by F5. Federator.ai currently only supports NGINX Ingress Controller in Kubernetes clusters that use Prometheus as the metrics data source.

2. Click to add controllers (generic) or consumer groups (Kafka Consumer) or upstream services (Ingress).

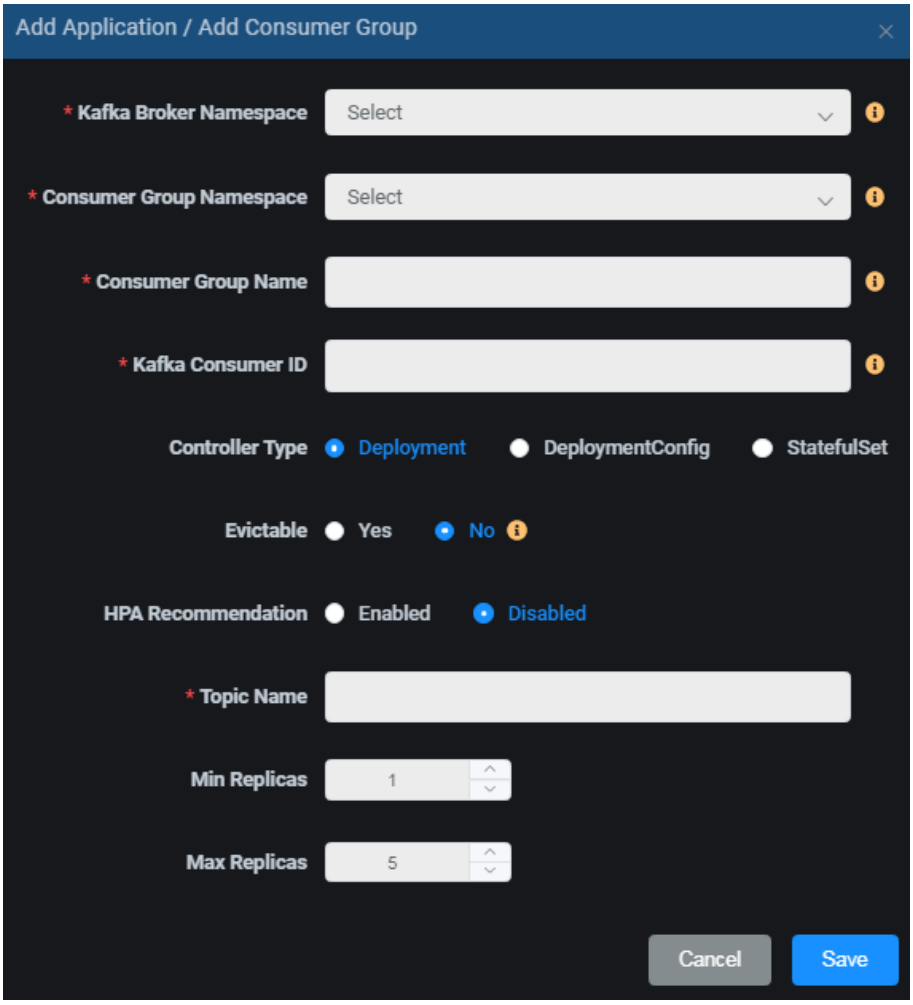
Generic



- *Namespace* – Select the Kubernetes namespace where the controller is deployed.
- *Controller Name* – Specify 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.
- *Auto Provisioning Profile* – Indicate if you want to select an auto provisioning 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*: *Auto Provisioning* - 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.

Kafka Consumer

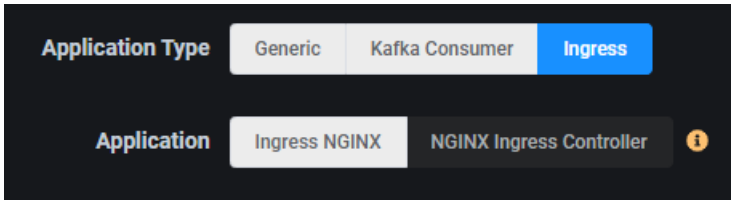


- *Kafka Broker Namespace* – Select the namespace for the Kafka broker that receives and stores messages from producers and allows consumers to fetch the messages.
- *Consumer Group Namespace* – Select the namespace of the consumer group used for checking the offset on topics and processing messages.
- *Consumer Group Name* – Specify the name of the Kafka consumer group used for checking the offset on topics and processing messages.

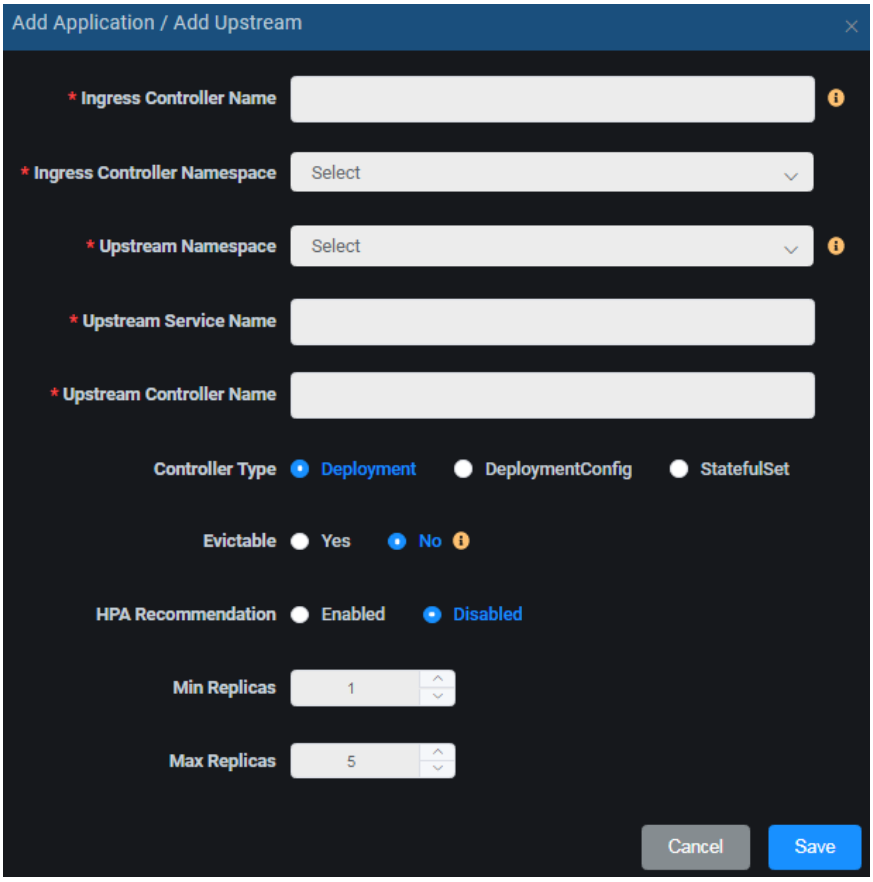
- *Kafka Consumer ID* – Specify the unique ID of the Kafka consumer group for the set of consumers within the same consumer group.
- *Controller Type* – Specify the controller type of the consumer group. Supported controller types are *Deployment, DeploymentConfig* (OpenShift only), and *StatefulSet.*
- *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, 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. If disabled, Federator.ai still monitors the message production rate but it will not auto scale Kafka consumer pods.
- *Topic Name* – Indicate the topic (where records are stored and published) that will be processed by consumers.
- *Mix/Max Replicas* – Specify the minimum and maximum number of consumers to be auto-scaled.

Ingress

When *Ingress* is selected as the *Application Type*, you can select *Ingress NGINX* or *NGINX Ingress Controller* for the application. Ingress NGINX is the community-supported version and NGINX Ingress Controller is a commercial version by F5. The NGINX Ingress Controller option is only enabled when you select a Kubernetes cluster that uses Prometheus as the metrics data source.



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.



- *Ingress Controller Name* – Specify the deployment name of the Ingress controller. Click the *Info* button to see a diagram showing the components of an Ingress system.
- *Ingress Controller Namespace* – Select the namespace of the Ingress controller.
- *Upstream Namespace* – Specify the namespace of the upstream service. Click the *Info* button to see a diagram showing the components of an Ingress system.
- *Upstream Service Name* – Specify the name of the upstream service to be scaled.
- *Upstream Controller Name* – Specify the deployment name of the controller that is used to scale the upstream service.
- *Controller Type* – Specify the Ingress controller type. Supported controller types are *Deployment, DeploymentConfig* (OpenShift only), and *StatefulSet.*
- *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, HTTP services are monitored, and the number of services is automatically increased/decreased based on the workload. If disabled, Federator.ai still monitors the HTTP services but it will not auto scale them.
- *Mix/Max Replicas* – Specify the minimum and maximum number of services to be auto-scaled.

3. Click *Save*.
4. Continue adding controllers, consumer groups, or upstream HTTP services and click *Save* when done.

Manage Applications

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

- Add, edit, or remove a controller, consumer group, or upstream HTTP service. To do this, click the *Edit Application* icon and click the appropriate icon:

Icon	Function
	Edit a controller, consumer group, or upstream HTTP service.
	Remove a controller, consumer group, or upstream HTTP service.

- 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

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.

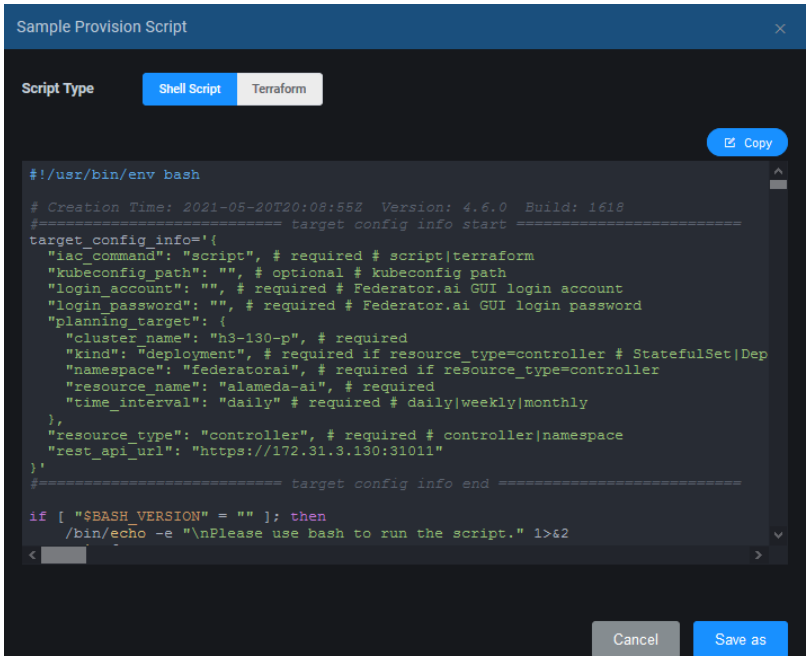
For each profile, you will see the frequency of the recommendations, CPU and memory adjustments, auto provisioning schedule, and which controllers and namespaces are using the profile. Purple represents a controller and blue represents a namespace.

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.

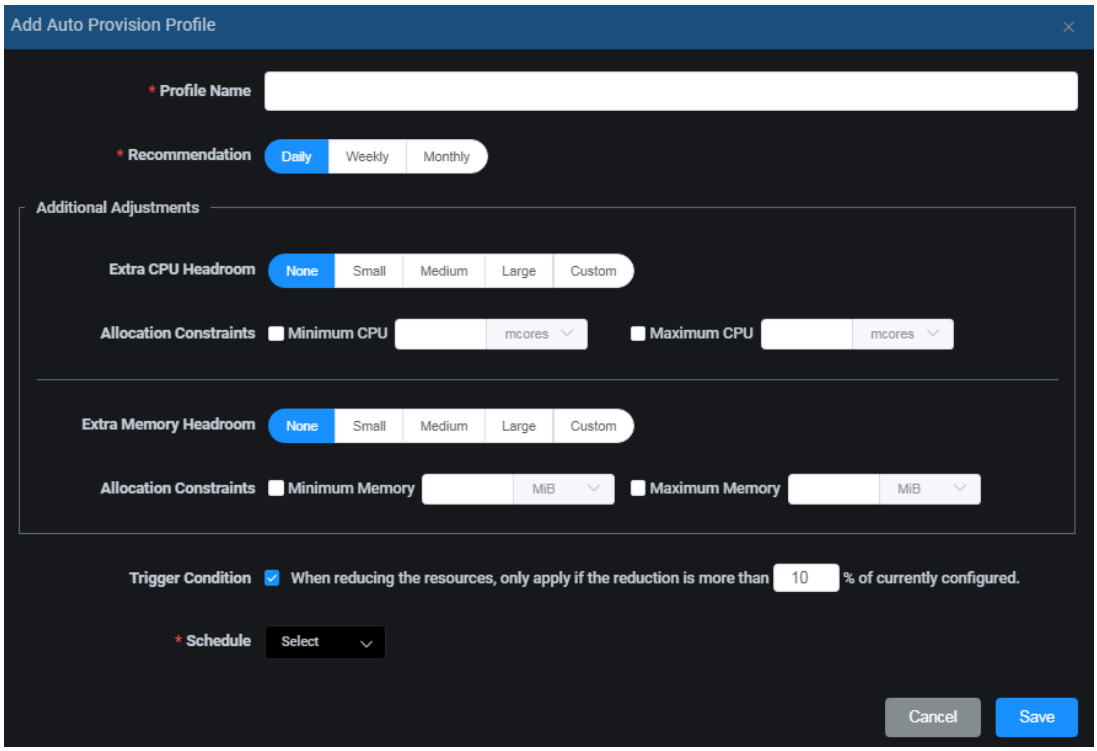
You can find the scripts, save the scripts locally, and copy these scripts via the following pages: *Configuration / Applications, Configuration / Clusters* (when you edit a namespace), or *Planning / Kubernetes Workload Prediction* (when you are viewing information for controllers or namespaces).



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*.



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 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.
- 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.

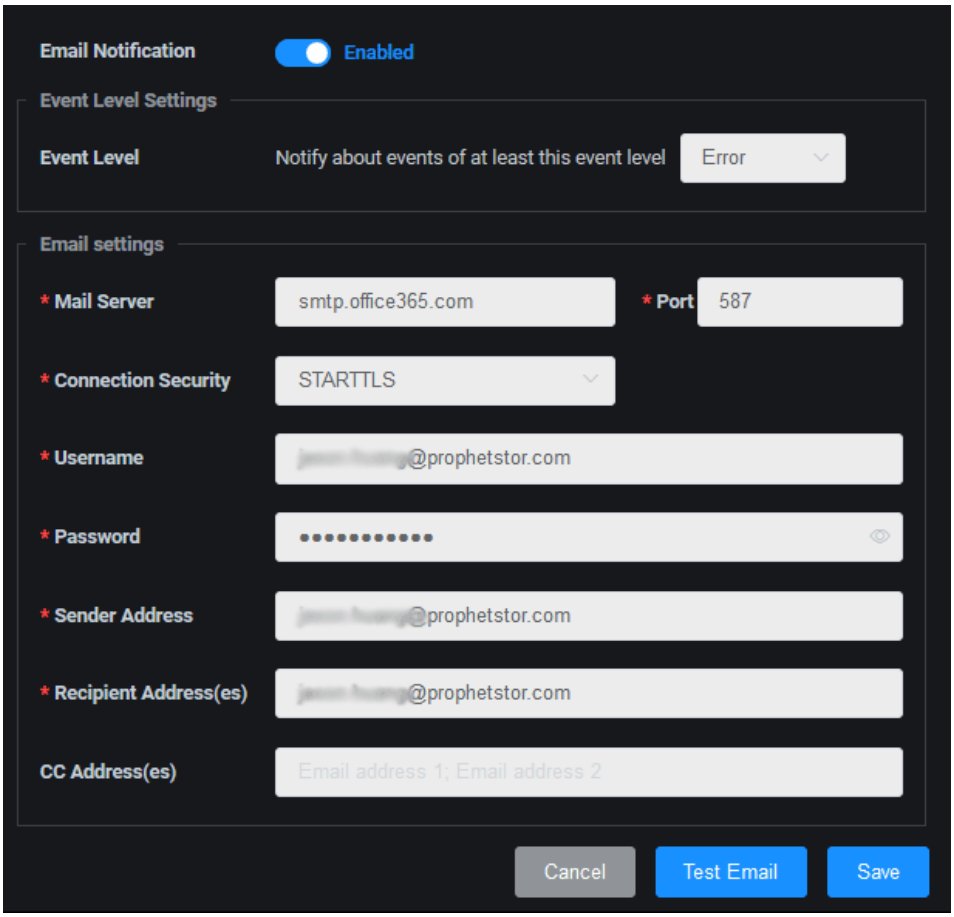
Notification

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*.



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 Connection* 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 license, including license type, status, and expiration date. It also shows what is included in your license.

In addition to adding a license, you can perform the following functions:

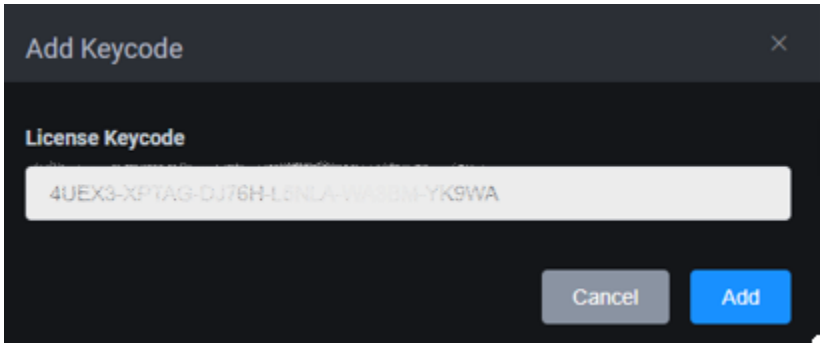
Icon	Function
	Activate a license.
	Show license keycode.
	Remove a license.

Add a License

There is one license per system; you must delete the existing license before adding a new one.

Follow the steps below to add and register a license:

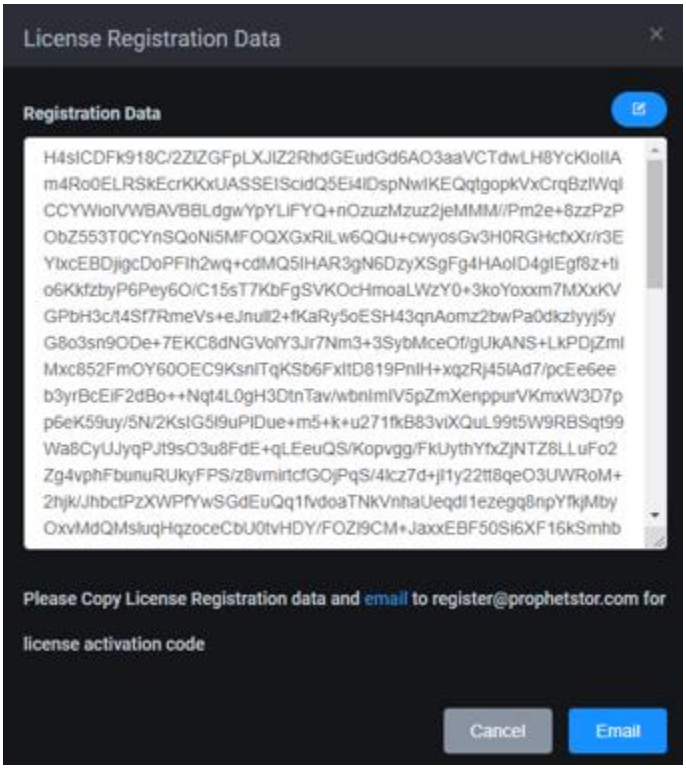
1. Click the *Add License* icon and enter the keycode.



2. Click *Add*.

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

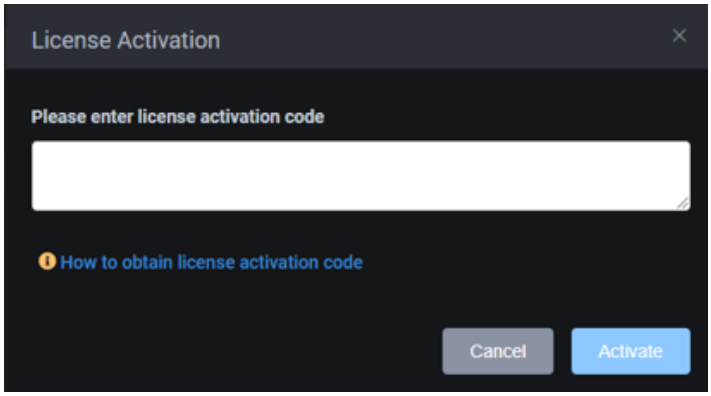
3. Email the registration data to register@prophetstor.com for a license activation code.



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.



5. Click *Activate*.

Manage Licenses

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

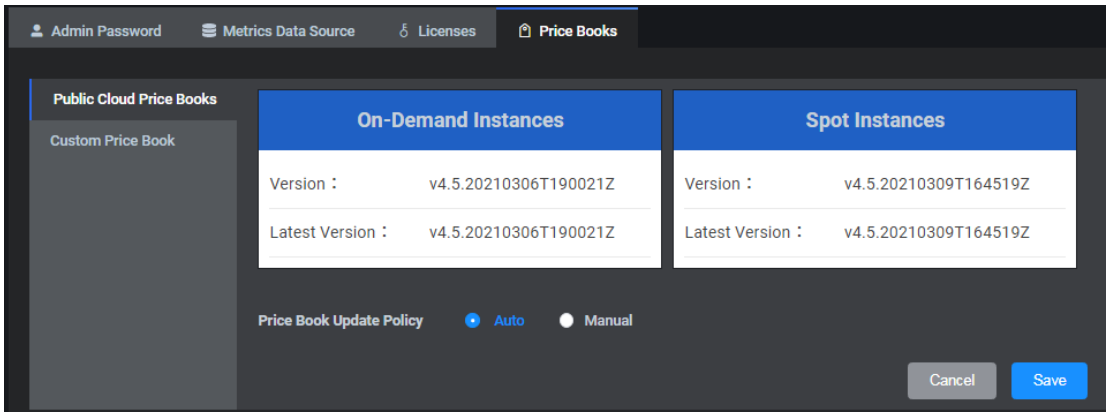
- View the keycode for your license. To do this, click the *Show License Keycode* icon.
- Remove the license. To do this, click the *Remove License* icon and confirm the removal.

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.



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

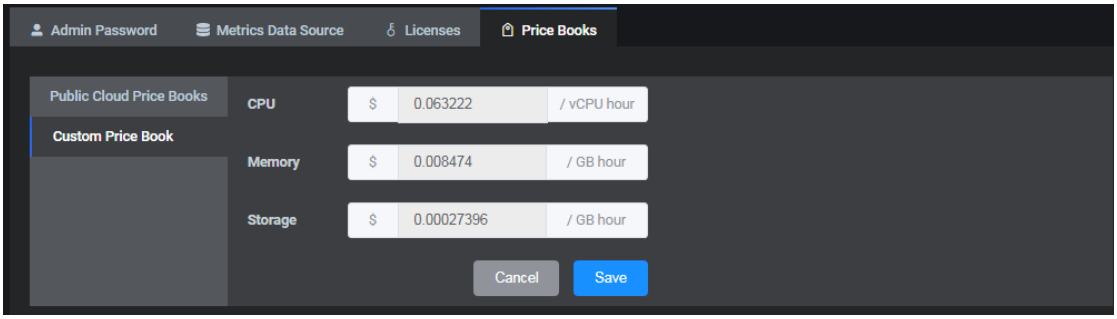
The *Price Book Update Policy* allows you to specify how you want the price books updated. *Auto* regularly checks availability and automatically downloads the latest version, which requires Internet access.

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.



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

Events

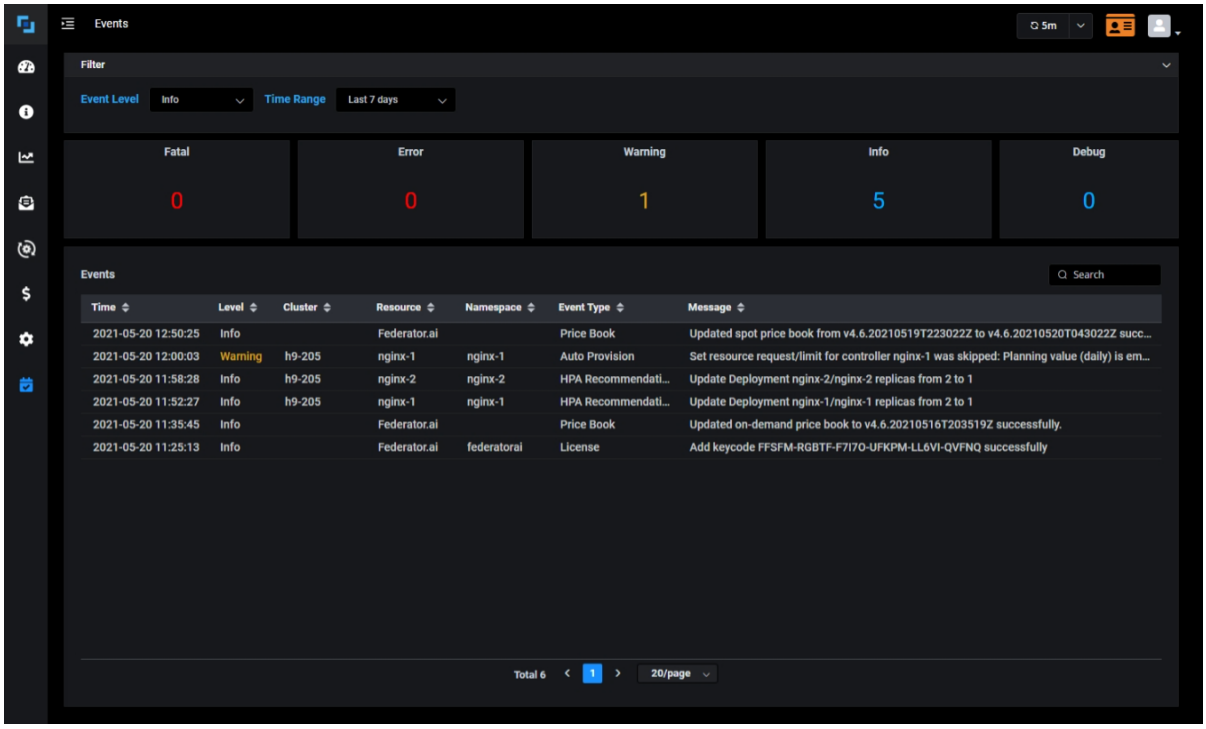
The *Events* page displays all system events that have occurred.

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 select the event level, which includes all levels above, 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.



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