



Federator.ai for Datadog

User Guide

Federator.ai version 4.3 for Datadog 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 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.

Datadog is a registered trademark of Datadog Corporation, 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.

10.21.2020

Contents

Overview	3
Datadog Integration Workflows	3
ProphetStor Dashboards in Datadog	4
Integrated Monitors	5
ProphetStor Federator.ai Cluster Overview Dashboard	6
Cluster Resource Usage Predications and Recommendations	6
Node Resource Utilization History.....	7
ProphetStor Federator.ai Application Overview Dashboard	8
Application Workload Predictions/Resource Recommendations.....	8
Application Resource Utilization History	9
Autoscaling/HPA Recommendations	9
Current/Desired/Recommended Replicas	10
CPU Usage/Request/Limit vs. Recommended CPU Limit	10
Memory Usage/Request/Limit vs. Recommended Memory Limit	10
CPU Usage/Limit Utilization	10
ProphetStor Federator.ai Kafka Overview Dashboard	11
Recommended Replicas vs Current Desired Replicas	12
Production vs Consumption vs Product Prediction	12
Kafka Consumer Lag	12
Consumer Queue Latency	12
Deployment Memory Usage	12
Deployment CPU Usage	12
ProphetStor Federator.ai Cost Analysis Overview Dashboard	13
Multi-cloud Cost Analysis	13
Namespace Cost Allocation Charts.....	14
Metrics	15
Common Metrics	15
Resource recommendation for an application for autoscaling purposes.....	15
Rolled-up CPU usage metrics by Federator.ai for a Kubernetes node.....	16
Rolled-up memory metrics by Federator.ai for a Kubernetes node	17
Rolled-up CPU usage by Federator.ai for a Kubernetes controller	18
Rolled-up memory metrics by Federator.ai for a Kubernetes controller	19

Health status of Federator.ai integration.....	20
Generic Application Metrics for Predictions – Nodes.....	21
Workload predictions for a node	21
Average value of predictions for a node	22
Minimum value of predictions for a node.....	23
Maximum value of predictions for a node	24
Generic Application Metrics for Predictions – Controllers	25
Workload predictions for a specific controller.....	25
Average value of predictions for a specific controller	26
Minimum value of predictions for a specific controller	27
Maximum value of predictions for a specific controller	28
Generic Application Metrics for Resource Planning – Nodes	29
Resource planning for a node	29
Resource planning for a controller.....	30
Kafka Metrics.....	31
Workload predictions for Kafka.....	31
Kafka broker offset rate.....	32
Kafka consumer offset rate	32
Cost Analysis Metrics.....	33
Cost of a recommended cloud instance.....	33
Cost analysis for a cloud instance	34
Cost allocation for a namespace	35
Cost prediction for a namespace.....	36
Cost percentage prediction for a namespace	36
Datadog Federator.ai Metrics List and Updating Interval	37

Overview

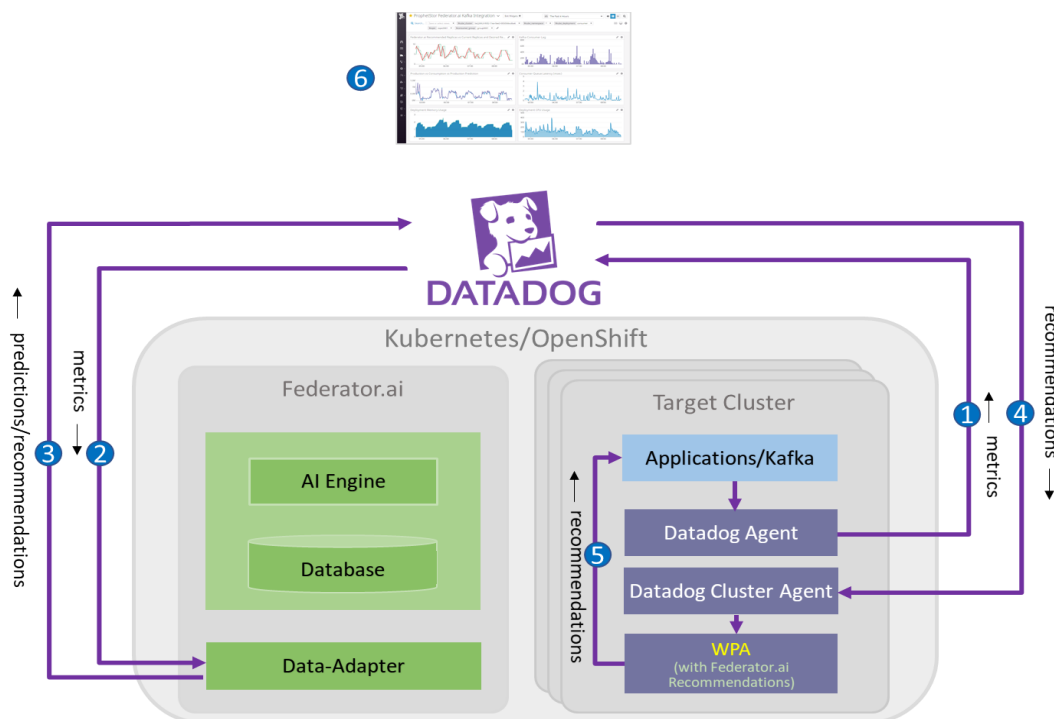
ProphetStor Federator.ai is an AI-based solution that helps enterprises manage, optimize, and auto-scale resources for any applications on Kubernetes. Using advanced machine learning algorithms to predict application workloads, Federator.ai scales the appropriate amount of resources at the right time for optimized application performance. Federator.ai offers:

- AI-based workload prediction for Kafka and other applications
- Resource recommendations based on workload prediction, application, Kubernetes, and other related metrics
- Automatic scaling of application containers through Datadog Watermark Pod Autoscaler (WPA)

Datadog Integration Workflows

Datadog provides monitoring for servers, applications, and services. With Datadog, enterprise customers can monitor their application workloads and get visibility into Kubernetes clusters. Federator.ai integration with Datadog aggregates metrics and events and provides resource predictions/recommendation for Kubernetes deployments and application-aware acceleration and optimization for Kafka.

The following diagram shows how application metrics are used by Federator.ai to predict workloads and to automatically scale applications for better performance:



1. The Datadog Agent sends cluster and application metrics to Datadog Services.
2. The Federator.ai Data-Adapter queries cluster and application metrics from Datadog Services and forwards to the Federator.ai AI engine.

3. The Federator.ai Data-Adapter posts its predictions, recommendations, and plans to Datadog Services.
4. The Datadog Cluster Agent gets the predictions, recommendations, and plans from Datadog Services.
5. WPA applies the plans and auto-scales applications.
6. The Datadog dashboard displays cluster and application metrics as well as predictions, recommendations, and plans from Federator.ai.

ProphetStor Dashboards in Datadog

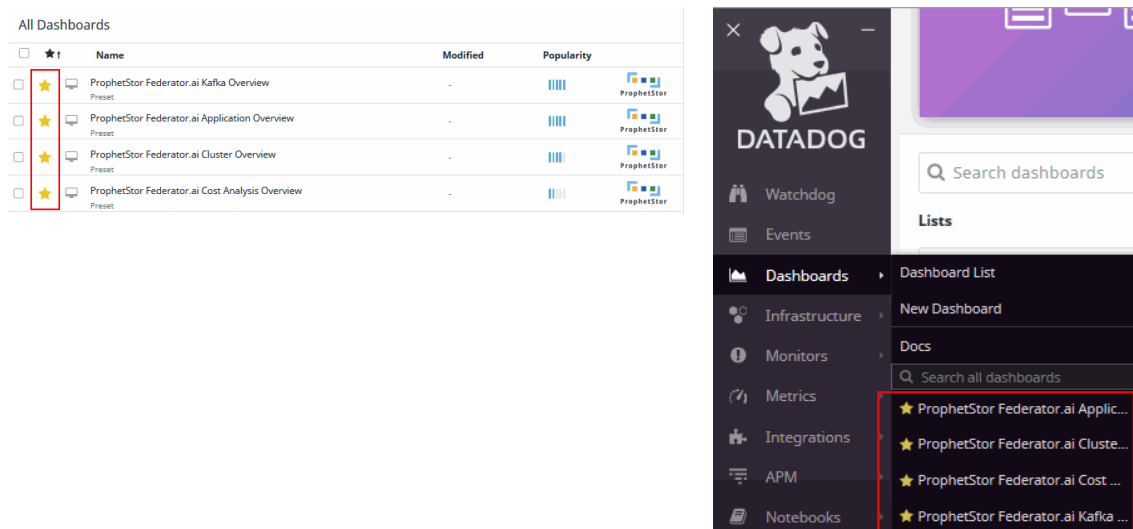
Note that Federator.ai must be installed and configured for Datadog before you can access the ProphetStor dashboards from Datadog. Refer to your *Federator.ai Installation Guide* for more details.

ProphetStor Federator.ai has four dashboards available in Datadog:

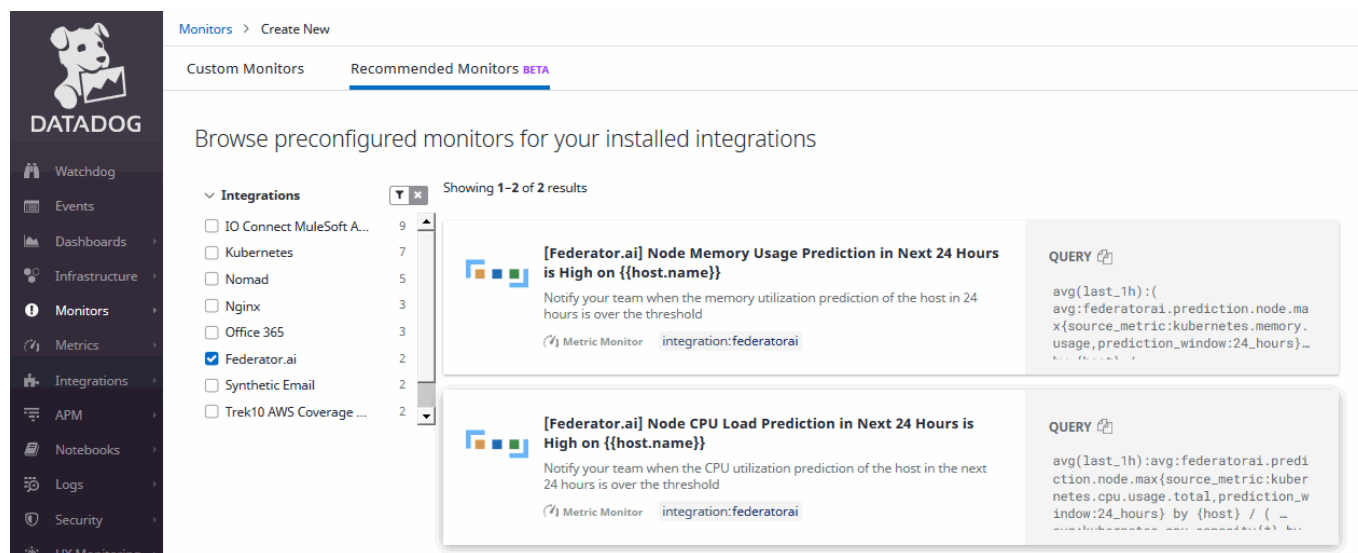
- ProphetStor Federator.ai Cluster Overview - Resource usage predictions and recommendations for clusters and nodes and historical utilization.
- ProphetStor Federator.ai Application Overview - Predicted CPU and memory usage and recommendations for each application.
- ProphetStor Federator.ai Kafka Overview - Usage information and recommendations about Kafka consumer replicas.
- ProphetStor Federator.ai Cost Analysis Overview

To access these dashboards from Datadog, select *Dashboards* → *Dashboard List*. The ProphetStor dashboards are listed and can be easily located by the ProphetStor logo. Click the name of the dashboard you want to view to display that dashboard.

For easier access to the dashboards, click the yellow star next to each ProphetStor Federator.ai dashboard so that they appear on under *Dashboards* in the navigation pane.



Integrated Monitors



The screenshot shows the Datadog 'Monitors' page. On the left is a navigation sidebar with the Datadog logo and various tool icons. The main content area is titled 'Monitors > Create New' and has two tabs: 'Custom Monitors' and 'Recommended Monitors BETA'. Below the tabs, it says 'Browse preconfigured monitors for your installed integrations'. A list of integrations is shown on the left, with 'Federator.ai' selected. To the right, two monitor cards are displayed. The first card is titled '[Federator.ai] Node Memory Usage Prediction in Next 24 Hours is High on {{host.name}}' and includes a description, a 'Metric Monitor' tag, and a query. The second card is titled '[Federator.ai] Node CPU Load Prediction in Next 24 Hours is High on {{host.name}}' and also includes a description, a 'Metric Monitor' tag, and a query.

Monitors > Create New

Custom Monitors Recommended Monitors **BETA**

Browse preconfigured monitors for your installed integrations

Integrations Showing 1-2 of 2 results

- ☐ IO Connect MuleSoft A...
- ☐ Kubernetes
- ☐ Nomad
- ☐ Nginx
- ☐ Office 365
- ☒ Federator.ai
- ☐ Synthetic Email
- ☐ Trek10 AWS Coverage ...

[Federator.ai] Node Memory Usage Prediction in Next 24 Hours is High on {{host.name}}

Notify your team when the memory utilization prediction of the host in 24 hours is over the threshold

Metric Monitor integration: federatorai

QUERY [🔗](#)

```
avg(last_1h):({
  avg:federatorai.prediction.node.ma
  x{source_metric:kubernetes.memory.
  usage,prediction_window:24_hours}...
```

[Federator.ai] Node CPU Load Prediction in Next 24 Hours is High on {{host.name}}

Notify your team when the CPU utilization prediction of the host in the next 24 hours is over the threshold

Metric Monitor integration: federatorai

QUERY [🔗](#)

```
avg(last_1h):avg:federatorai.predi
  ction.node.max{source_metric:kuber
  netes.cpu.usage.total,prediction_w
  indow:24_hours} by {host} / (...
```

Federator.ai has two predefined monitors in Datadog to alert you when CPU load and memory usage is predicted to be high for a node over the next 24 hours. By default, these monitors are enabled and will trigger a warning when CPU/memory usage is above 80% and will trigger an alert when CPU/memory usage is above 90% is used.

You can modify these alert/warning thresholds and can set thresholds for warning and alert recovery. In addition, you need to set up email notification for your team members. When a warning/alert is triggered/recovered, an email will be sent to selected recipients.

In the Datadog navigation, select *Monitors* → *New Monitor*, click the *Recommended Monitors* tab, and select Federator.ai under *Integrations*. Click one of the monitors to set/change your threshold and specify email recipients. Click *Save* when done.

At the top of the monitor screen, you can see historical instances of when this monitor was triggered for nodes.


Federator.ai alerts and warnings that have occurred can be viewed on the *Monitors* → *Manage Monitors* page. Click on each for more details.

ProphetStor Federator.ai Cluster Overview Dashboard

The *Cluster Overview* dashboard has two sections. The tables at the top display future resource usage predictions and recommendations for clusters and nodes (hosts). The charts below display node resource utilization history.

By default, information in the tables is displayed for all clusters and all hosts for the upcoming 24 hours, but you can filter information for a specific cluster or node (host) and change the timeframe to 7 or 30 days. There are several cluster fields; use the appropriate field for your configuration based upon how your data is tagged.

Cluster Resource Usage Predications and Recommendations



ProphetStor

Cluster Resource Usage Predications and Recommendations

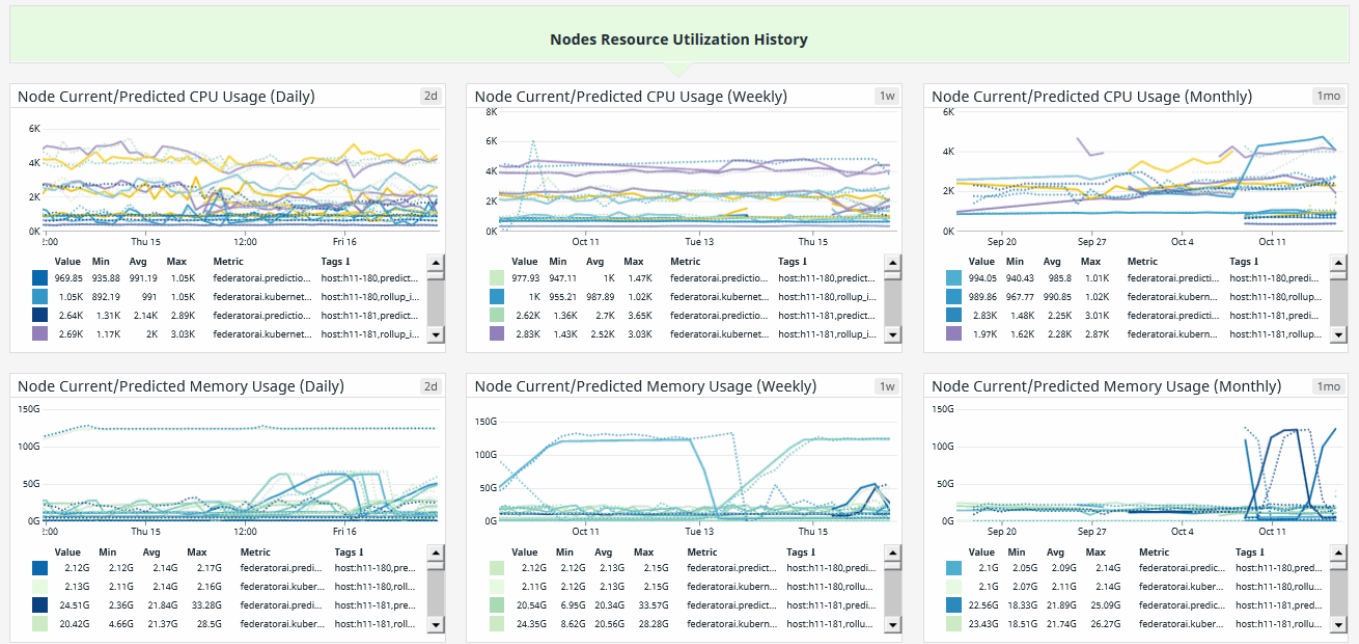
KUBE_CLUSTER...	KUBE_CLUSTER...	CLUSTER_NAME...	PREDICTION...	AVG CPU (MC)	MIN CPU (MC)	MAX CPU (MC)	REC CPU (MC)	AVG MEM (B)	MIN MEM (B)	MAX MEM (B)	REC MEM (B)
jason11-180	jason11-180	jason11-180	24_hours	7.88K	6.64K	9.15K	8.52K	65.54G	35.05G	88.44G	96.54G
jason-11-30	jason-11-30	jason-11-30	24_hours	6.88K	5.54K	8.23K	7.56K	136.6G	74.52G	185.34G	197.12G
jpn-k8s-117-1013	N/A	N/A	24_hours	4.31K	4.18K	4.53K	4.42K	13.05G	11.8G	13.48G	14.82G

Cluster Node Resource Usage Predications and Recommendations

KUBE_CLUSTER	KUBE_CLUSTER...	CLUSTER_NAME	HOST	PREDICTION...	AVG CPU (MCOR)	MIN CPU (MCOR)	MAX CPU (MCOR)	REC CPU (MCOR)	AVG MEM (BYTE)	MIN MEM (BYTE)	MAX MEM (BYTE)	REC MEM (BYTE)
jpn-k8s-117-1013	N/A	N/A	jpn-node-02	24_hours	4.31K	4.18K	4.53K	4.42K	13.05G	11.8G	13.48G	14.82G
jc-k8s-h4207	N/A	N/A	k8s-n3	24_hours	4.14K	3.79K	4.83K	4.49K	12.9G	11.64G	13.32G	14.65G
jason11-180	jason11-180	jason11-180	h11-183	24_hours	3.07K	2.58K	3.65K	3.36K	21.36G	11.34G	29.35G	32.28G
jason11-180	jason11-180	jason11-180	h11-182	24_hours	2.37K	1.93K	2.82K	2.6K	20.27G	4.59G	31.2G	33.57G
jason-11-30	jason-11-30	jason-11-30	ocp4-4phcd-work	24_hours	2.22K	1.71K	2.74K	2.48K	18.64G	6.44G	42.56G	46.82G
jason-11-30	jason-11-30	jason-11-30	ocp4-4phcd-mas	24_hours	1.99K	1.84K	2.13K	2.06K	24.7G	24.33G	25.39G	27.93G
jason-11-30	jason-11-30	jason-11-30	ocp4-4phcd-work	24_hours	1.68K	1.14K	2.23K	1.96K	50.56G	12G	67.55G	67.55G
jason-11-30	jason-11-30	jason-11-30	ocp4-4phcd-work	24_hours	1.68K	1.14K	2.23K	1.96K	50.56G	12G	67.55G	67.55G


In the *Cluster Resource Usage Predications and Recommendations* section, the tables display predicted average, minimum, and maximum CPU (in millicores) and memory usage, along with recommended allocations for each cluster and its nodes.

Node Resource Utilization History



In the *Node Resource Utilization History* section, current and previously predicted CPU and memory usage information is shown in charts. The solid line represents the observed actual usage, while the dotted line shows what was predicted. Click anywhere on a chart to see values for a specific point in time. By default, all nodes are listed, but you select a specific node from the *Host* drop-down box at the top of the dashboard or you can select/deselect nodes at the bottom of a chart to show/hide them from the display.

Daily charts include data from the past two days, weekly charts include data from the past week, and monthly charts include data from the past month.


To view a full screen chart, click the zoom icon  (top right of widget icon).

ProphetStor Federator.ai Application Overview Dashboard

The *Application Overview* dashboard has three sections. The tables at the top display future resource usage predictions and recommendations for all user-defined applications. Refer to your *Federator.ai Installation Guide* to configure applications that should be monitored by Federator.ai.

The charts below display application resource utilization history, and autoscaling/HPA recommendations. Information in the charts is combined for all selected applications, making them difficult to use. In order for the charts to be useful, you should specify a single application. To do this, select a cluster, a namespace, and a deployment (controller). You can also filter by deployment type. If you are deploying applications in a Kubernetes environment, filter with the fields *\$kubernetes_deployment* or *\$kubernetes_stateful_set*. If you are deploying applications in an OpenShift environment, filter with the fields *\$openshift_deployment_config* or *\$openshift_deployment*.

Application Workload Predictions/Resource Recommendations



Application Workload Prediction/Resource Recommendation

Workload Prediction for Next 24 Hours

KUBE_NAM...	KUBE_DEPL...	KUBE_STAT...	OSHIFT_DE...	AVG CPU (MC	MIN CPU (MC	MAX CPU (MC	REC CPU (MC)	AVG MEM (BY	MIN MEM (BY	MAX MEM (BY	REC MEM (BY
nginx1	my-nginx	N/A	N/A	238.64	194.75	304.1	272	31.1M	25.07M	34.82M	38.3M
myproject	consumer	N/A	N/A	82.96	65.64	116.89	100	1.42G	1.22G	1.8G	1.98G

Workload Prediction for Next 7 Days

KUBE_NAMESPA...	KUBE_DEPLOYM...	KUBE_STATEFUL...	OSHIFT_DEPLOY...	AVG CPU (MCORE)	MIN CPU (MCORE)	MAX CPU (MCORE)	REC CPU (MCORES)	AVG MEM (BYTES)	MIN MEM (BYTES)	MAX MEM (BYTES)	REC MEM (BYTES)
nginx1	my-nginx	N/A	N/A	263.98	239.62	288.33	277	35.77M	34.34M	37.16M	40.88M
myproject	consumer	N/A	N/A	94.41	90.72	98.1	97	1.56G	1.21G	2.05G	2.26G

Workload Prediction for Next 30 Days

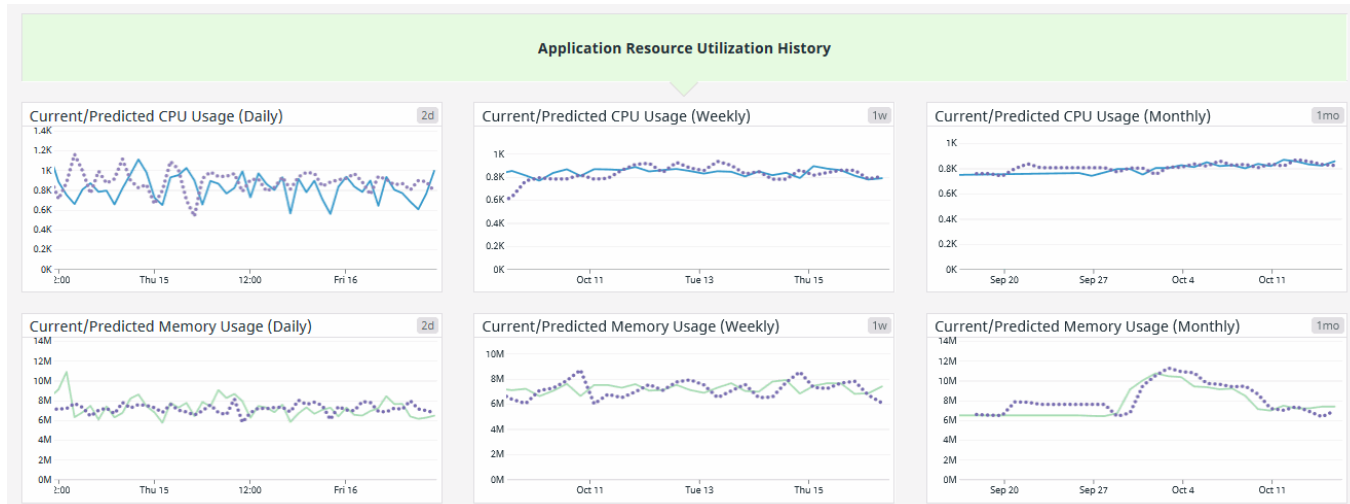
KUBE_NAMESPA...	KUBE_DEPLOYM...	KUBE_STATEFUL...	OSHIFT_DEPLOY...	AVG CPU (MCORE)	MIN CPU (MCORE)	MAX CPU (MCORE)	REC CPU (MCORES)	AVG MEM (BYTES)	MIN MEM (BYTES)	MAX MEM (BYTES)	REC MEM (BYTES)
nginx1	my-nginx	N/A	N/A	240.39	240.39	240.39	N/A	35.88M	35.88M	35.88M	N/A
myproject	consumer	N/A	N/A	80.43	80.43	80.43	N/A	1.46G	1.46G	1.46G	N/A

In the *Application Workload Prediction/Resource Recommendations* section, the tables display predicted average, minimum, and maximum CPU (in millicores) and memory usage, along with recommended allocations for each user-defined application in a cluster, namespace, deployment, and deployment type.

By default, information in the charts is displayed for all clusters, namespaces, and deployments, but you can filter information for a specific cluster, namespace, deployment, or deployment type. There are several cluster fields; use the appropriate field for your configuration based upon how your data is tagged.


Information is displayed for the next 24 hours, 7 days, and 30 days.

Application Resource Utilization History

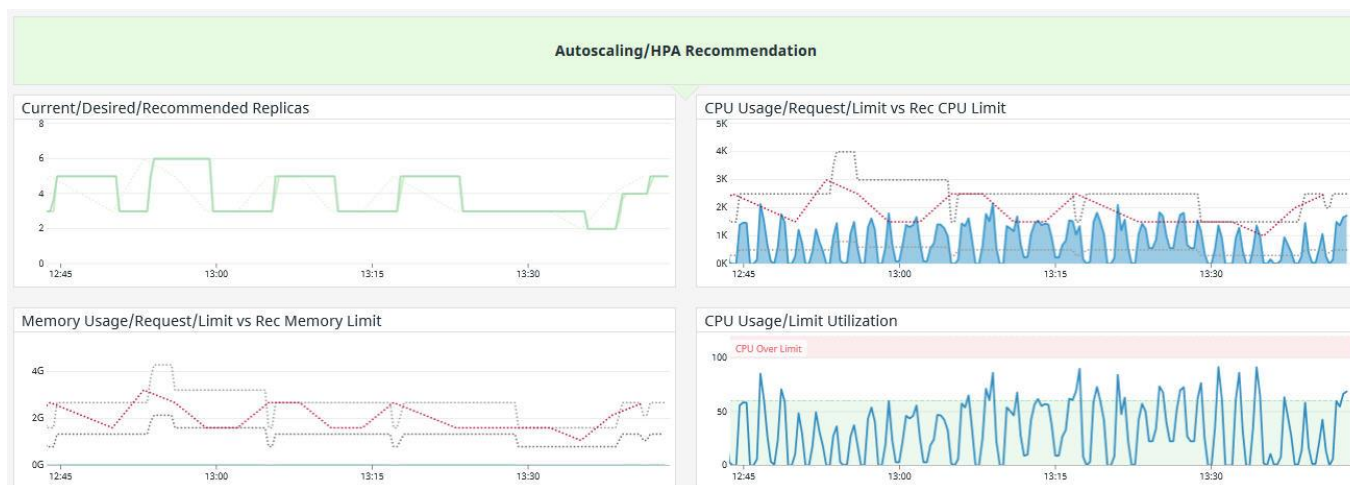


In the *Application Resource Utilization History* section, current and previously predicted CPU and memory usage information is shown in charts. The solid line represents the observed actual usage, while the dotted line shows what was predicted. Click anywhere on a chart to see values for a specific point in time.

Daily charts include data from the past two days, weekly charts include data from the past week, and monthly charts include data from the past month.


To view a full screen chart, click the zoom icon  (top right of widget icon).

Autoscaling/HPA Recommendations



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.

The charts display Horizontal Path Autoscaling (HPA) recommendations.

To view a full screen chart, click the zoom icon  (top right of widget icon).

Current/Desired/Recommended Replicas

This chart displays the number of current, desired, and recommended replicas.

CPU Usage/Request/Limit vs. Recommended CPU Limit

This chart displays the CPU usage by the current replicas, along with requested CPU usage, and the CPU limit. Because of the amount of detail, it is recommended to view this chart full screen. Click on the key at the bottom of the chart to show/hide individual metrics.

The solid line/area (blue, *kubernetes.cpu.usage.total*) represents the current CPU usage. The middle dotted line (light gray, *kubernetes.cpu.requests*) represents the requested CPU usage. The top dotted line (dark gray, *kubernetes.cpu.limits*) represents the CPU limit. The bottom dotted line (red, *federatorai.recommendation*) represents the recommended CPU usage.

If actual CPU usage is much lower than what is requested, it indicates that the amount of CPU is being over-provisioned.

Memory Usage/Request/Limit vs. Recommended Memory Limit

This chart displays the memory usage by the current replicas, along with requested memory usage, and the memory limit. Because of the amount of detail, it is recommended to view this chart full screen. Click on the key at the bottom of the chart to show/hide individual metrics.

The solid line/area (blue, *kubernetes.memory.usage*) represents the current memory usage. The middle dotted (dark gray, *kubernetes.memory.requests*) line represents the requested memory usage. The top dotted line (light gray, *kubernetes.memory.limits*) represents the memory limit. The bottom line (red, *federatorai.recommendation*) represents the recommended memory usage.

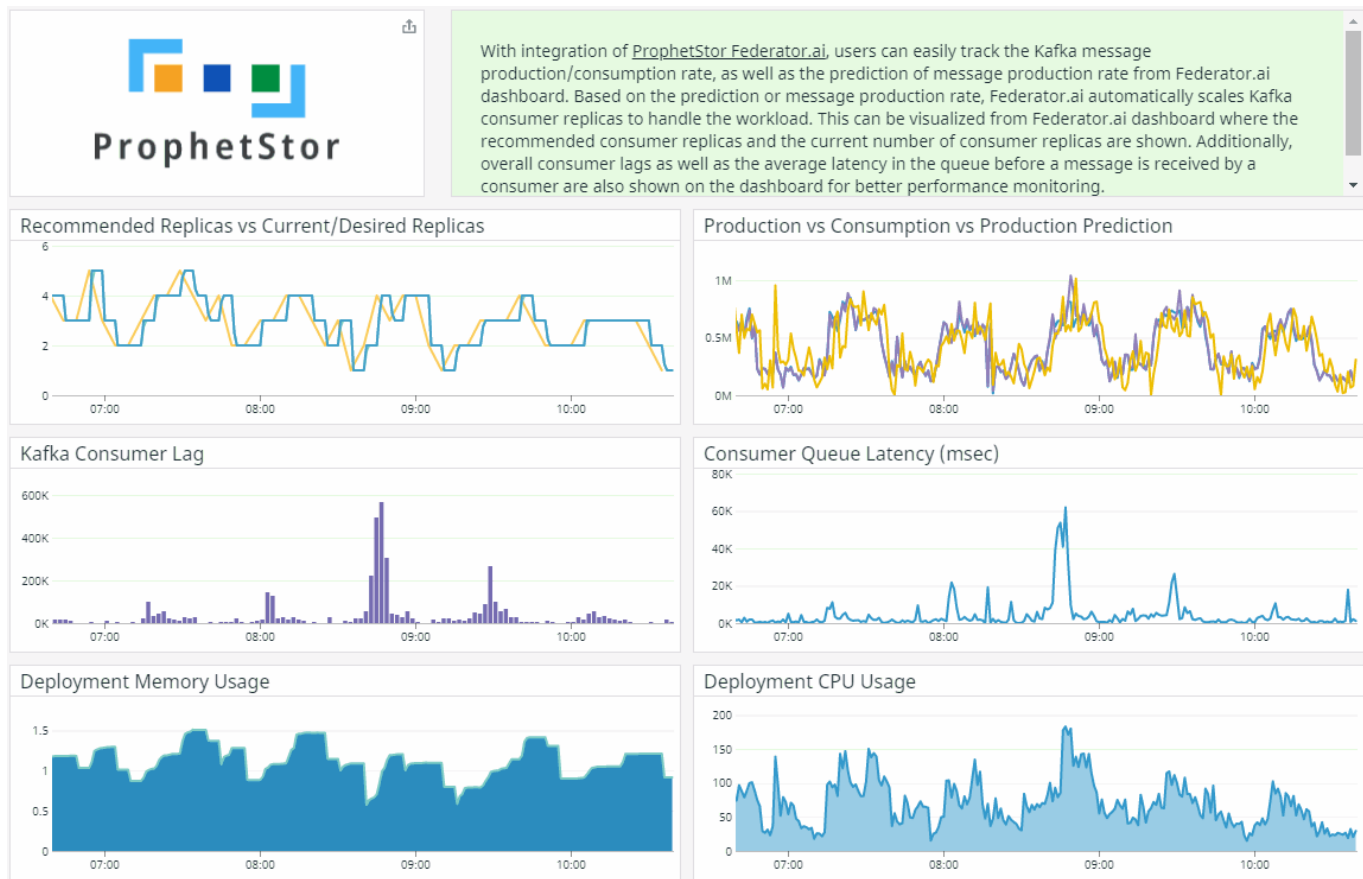
If actual memory usage is much lower than what is requested, it indicates that the amount of memory is being over-provisioned.

CPU Usage/Limit Utilization

This chart displays actual CPU utilization, which is determined by dividing actual CPU usage (represented by the blue line) by the CPU limitation.

Since autoscaling is based on usage, replicas are added/removed to keep your system within the CPU limits.

ProphetStor Federator.ai Kafka Overview Dashboard



If you are using Kafka software, the *Kafka Overview* dashboard displays usage information and recommendations about Kafka consumer replicas for clusters, namespaces, monitored topics, and monitored consumer groups.

With Kafka, the Datadog Agent monitors and collects many types of Kafka metrics, including resource (CPU and memory) utilization and Kafka workload and performance metrics (such as message queue latency and consumer lag). With the Federator.ai integration, the Kafka message production and consumption rates are continuously collected from Datadog and analyzed by Federator.ai to predict the Kafka message production rate. Based on these predictions, Federator.ai works with the Datadog Watermark Pod Autoscaler (WPA) to automatically scale Kafka consumer replicas to handle the increased or decreased workload.

By default, information in the charts is displayed for all clusters, namespaces, deployments, topics, and consumer groups for the past hour but you can filter information for a specific cluster, namespace, deployment, topic, or consumer group and select a different time frame. There are several cluster fields; use the appropriate field for your configuration based upon how your data is tagged.

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

Recommended Replicas vs Current Desired Replicas

This chart displays the recommended and current consumer replicas for the specified time range.

The yellow line represents recommended replicas every five minutes and the green line represents the actual number of replicas. The numbers align closely because once a recommendation is made to increase/decrease the number of replicas and autoscaling occurs, the lines match up.

Production vs Consumption vs Product Prediction

This chart displays the actual observed message production rate (blue line) and consumption/processed rate (purple line) for the specified time range and the predicted rate (yellow line).

With Kafka, autoscaling is based on message production (arrival) vs. consumption (processed); it is not based on CPU/memory usage.

Kafka Consumer Lag

This chart displays the number of messages in the Kafka broker that are yet to be processed by the Kafka consumer for the selected group and time range. This represents messages in the queue waiting to be processed.

Consumer Queue Latency

This chart displays how long it takes for each message to be processed (in milliseconds) for the selected time range.

Deployment Memory Usage

This chart displays observed actual memory usage for the selected time range for the specified consumer.

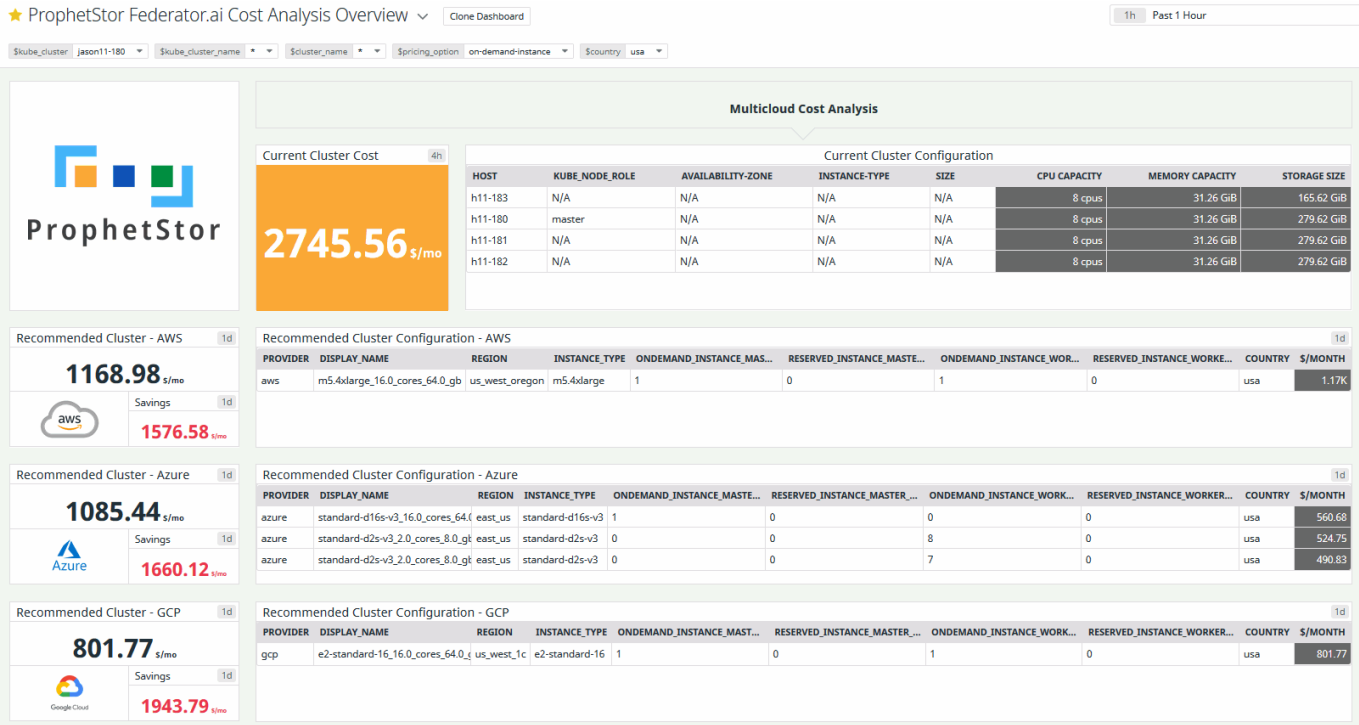
Deployment CPU Usage

This chart displays observed actual CPU usage for the selected time range for the specified consumer.

ProphetStor Federator.ai Cost Analysis Overview Dashboard

The *Cost Analysis Overview* dashboard has two sections. The Multi-cloud Cost Analysis displays your current cluster costs and outlines your cost for doing business with a variety of cloud providers. The Namespace charts below display the actual cost for each namespace.

Multi-cloud Cost Analysis



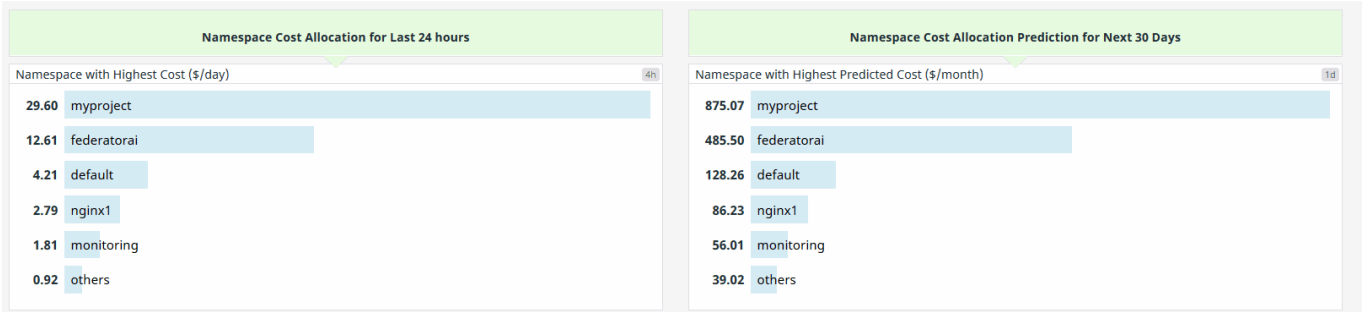
The *Multi-cloud Cost Analysis* displays the amount you are spending with your current cluster costs and outlines your cost for doing business with a variety of cloud providers based on fees charged by each provider and recommended changes to your current resource configuration. Select which cluster to display from the appropriate *Cluster* drop-down box. You can filter the display by pricing options and country.

The *Current Cluster Cost* box displays what you are currently paying to your cloud provider with your existing resource configuration along with your current cluster configuration, including hosts, roles, availability zone, instance type being used, CPU capacity, memory capacity, and storage size.

The *Recommended Cluster* boxes display potential savings with different providers. For each provider, the estimated cost/savings are displayed. The table on the right shows the recommended cluster configuration to realize these savings. This includes the recommended instances, regions, and instance types, as well as the number of master and worker servers with each pricing model (on-demand, reserved), and country. The savings typically come from lower provider fees as well as from reducing idle resources and using more cost-effective instances with a 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.

Namespace Cost Allocation Charts



Select which cluster to display from the appropriate *Cluster* drop-down box.

The *Namespace Cost Allocation for Last 24 hours* chart displays the namespaces with the highest cost for the last day.

The *Namespace Cost Allocation Prediction for Next 30 Days* chart displays a cost projection for the namespaces with the highest predicted costs for the next month.

Metrics

The following is a list of Federator.ai metrics that are used with Datadog. These queries can be used to build custom dashboards in Datadog. The syntax for metrics is:

Metric (Metric name), Value (Metric data), Tags

Common Metrics

Recommendation metrics for generic applications and Kafka

Resource recommendation for an application for autoscaling purposes

Metric

federatorai.recommendation

Used for WPA/HPA to auto-scale Kubernetes controllers for:

- Generic applications - The metric is updated to Datadog every three minutes and provides recommended CPU, memory, number of replicas.
- Kafka - The metric is updated to Datadog every five minutes and provides the recommended number of replicas.

Value

Recommended resource usage.

Tags

Name	Description
resource	Type of resource being recommended for auto-scaling the application. Options: <i>replicas</i> , <i>cpu</i> , <i>memory</i> . For Kafka, the resource is <i>replicas</i> . For generic applications, the resource is <i>cpu</i> , <i>memory</i> , or <i>replicas</i> .
kube_cluster	Kubernetes cluster name from Datadog, identifying the source cluster of the application being monitored.
kube_namespace	Kubernetes namespace from Datadog, identifying the source namespace of the application being monitored.
kube_deployment	Kubernetes deployment from Datadog, identifying the source deployment of the application being monitored.

Rolled-up CPU usage metrics by Federator.ai for a Kubernetes node

Metric

federatorai.kubernetes.cpu.usage.total.node

The rolled-up (3600sec, 21600sec, 86400sec) metrics of *kubernetes.cpu.usage.total*, which is the number of cores (in millicores) used by the Kubernetes node.

Value

CPU usage in millicores.

Tags

Name	Description
kube_cluster	Kubernetes cluster name from which <i>kubernetes.cpu.usage.total</i> is collected. Note: This tag is copied from <i>kubernetes.cpu.usage.total</i> , if available.
host	Hostname from which <i>kubernetes.cpu.usage.total</i> is collected. Note: This tag is copied from <i>kubernetes.cpu.usage.total</i> , if available.
rollup_interval	Rolled-up interval (in second) of the metric. Possible values: 3600, 21600, 86400.

Rolled-up memory metrics by Federator.ai for a Kubernetes node

Metric

federatorai.kubernetes.memory.usage.node

The rolled-up (3600sec, 21600sec, 86400sec) metrics of *kubernetes.memory.usage*, which is the memory usage (in bytes) of the Kubernetes node.

Value

Memory usage in bytes.

Tags

Name	Description
kube_cluster	Kubernetes cluster name from which <i>kubernetes.memory.usage</i> is collected. Note: This tag is copied from <i>kubernetes.memory.usage</i> , if available.
host	Hostname from which <i>kubernetes.memory.usage</i> is collected. Note: This tag is copied from <i>kubernetes.memory.usage</i> , if available.
rollup_interval	Rolled-up interval (in second) of the metric. Possible values: 3600, 21600, 86400.

Rolled-up CPU usage by Federator.ai for a Kubernetes controller

Metric

federatorai.kubernetes.cpu.usage.total.controller

The rolled-up (3600sec, 21600sec, 86400sec) metrics of *kubernetes.cpu.usage.total*, which is the number of cores (in millicores) used by the Kubernetes controller.

Value

CPU usage in millicores.

Tags

Name	Description
kube_cluster	Kubernetes cluster name from which <i>kubernetes.cpu.usage.total</i> is collected. Note: This tag is copied from <i>kubernetes.cpu.usage.total</i> , if available.
host	Hostname from which <i>kubernetes.cpu.usage.total</i> is collected. Note: This tag is copied from <i>kubernetes.cpu.usage.total</i> , if available.
rollup_interval	Rolled-up interval (in second) of the metric. Possible values: 3600, 21600, 86400.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Rolled-up memory metrics by Federator.ai for a Kubernetes controller

Metric

federatorai.kubernetes.memory.usage.controller

The rolled-up (3600sec, 21600sec, 86400sec) metrics of *kubernetes.memory.usage*, which is the memory usage (in bytes) of the Kubernetes node.

Value

Memory usage in bytes.

Tags

Name	Description
kube_cluster	Kubernetes cluster name from which <i>kubernetes.memory.usage</i> is collected. Note: This tag is copied from <i>kubernetes.memory.usage</i> , if available.
host	Hostname from which <i>kubernetes.memory.usage</i> is collected. Note: This tag is copied from <i>kubernetes.memory.usage</i> , if available.
rollup_interval	Rolled-up interval (in second) of the metric. Possible values: 3600, 21600, 86400.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from the <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Health status of Federator.ai integration

Metric

federatorai.integration.status

Used for sending Federator.ai health status to Datadog. This metric is always emitted and is used to show on the Datadog integration tile that Federator.ai integration is functioning correctly.

Value

0 – Status ok.

1 – Status error.

Generic Application Metrics for Predictions – Nodes

Workload predictions for a node

Metric

federatorai.prediction.node

Workload prediction for a Kubernetes node. The type of prediction is defined by the *source_metric* tag.

Value

Predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
prediction_granularity	Time interval between each prediction point. Possible values are <i>1_hour</i> , <i>6_hours</i> , or <i>24_hours</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
host	Hostname from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Average value of predictions for a node

Metric

federatorai.prediction.node.avg

The average value of workload predictions for a Kubernetes node for a prediction window. The type of workload predicted is defined by the *source_metric* tag.

Value

Average predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
host	Hostname from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Minimum value of predictions for a node

Metric

federatorai.prediction.node.min

The minimum value of workload predictions for a Kubernetes node for a prediction window. The type of prediction is defined by the *source_metric* tag.

Value

Minimum predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
host	Hostname from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Maximum value of predictions for a node

Metric

federatorai.prediction.node.max

The maximum value of workload predictions for a Kubernetes node for a prediction window. The type of prediction is defined by the *source_metric* tag.

Value

Maximum predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
host	Hostname from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Generic Application Metrics for Predictions – Controllers

Workload predictions for a specific controller

Metric

federatorai.prediction.controller

Workload prediction for a specific controller. The type of prediction is defined by the *source_metric* tag.

Value

Predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
prediction_granularity	Time interval between each prediction point. Possible values are <i>1_hour</i> , <i>6_hours</i> , or <i>24_hours</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Average value of predictions for a specific controller

Metric

federatorai.prediction.controller.avg

The average value of workload predictions for a specific controller for a prediction window. The type of prediction is defined by the *source_metric* tag.

Value

Average predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	The time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Minimum value of predictions for a specific controller

Metric

federatorai.prediction.controller.min

The minimum value of workload predictions for a specific controller for a prediction window. The type of prediction is defined by the *source_metric* tag.

Value

Minimum predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Maximum value of predictions for a specific controller

Metric

federatorai.prediction.controller.max

The maximum value of workload predictions for a specific controller for a prediction window. The type of prediction is defined by the *source_metric* tag.

Value

Maximum predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Generic Application Metrics for Resource Planning – Nodes

Resource planning for a node

Metric

federatorai.resource_planning.node

Resource planning for a specific node.

Value

Planned value.

Tags

Name	Description
resource	Type of resource being recommended for a node. Possible values are <i>cpu</i> or <i>memory</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
prediction_granularity	Time interval between each prediction point. Possible values are <i>1_hour</i> , <i>6_hours</i> , or <i>24_hours</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
host	Hostname from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Resource planning for a controller

Metric

federatorai.resource_planning.controller

Resource planning for a specific Kubernetes controller.

Value

Planned value.

Tags

Name	Description
resource	Type of resource being recommended for a controller. Possible values are <i>cpu</i> or <i>memory</i> .
prediction_window	Time interval for which the prediction value is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
prediction_granularity	Time interval between each prediction point. Possible values are <i>1_hour</i> , <i>6_hours</i> , or <i>24_hours</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_stateful_set	Kubernetes <i>StatefulSet</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
oshift_deployment_config	OpenShift <i>DeploymentConfig</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.

Kafka Metrics

Workload predictions for Kafka

Metric

federatorai.prediction.kafka

Workload prediction for a Kafka.

Value

Predicted workload value.

Tags

Name	Description
source_metric	Source metric from Datadog used for workload prediction. For example, <i>kubernetes.cpu.usage.toal</i> or <i>kubernetes.memory.usage</i> .
prediction_granularity	Time interval between each prediction point. The value is always <i>1_minute</i> .
kube_cluster	Kubernetes cluster name from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_namespace	Kubernetes namespace from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
kube_deployment	Kubernetes <i>Deployment</i> from which the source metric is collected. Note: This tag is copied from <i>source_metric</i> , if available.
topic	Kafka topic name of the Kafka workload being monitored.
consumer_group	Kafka consumer group of the Kafka workload being monitored.

Kafka broker offset rate

Metric

federatorai.kafka.broker_offset_rate

The number of broker offsets increased in the last 60 seconds.

Value

Kafka broker offset rate.

Tags

Name	Description
kube_cluster	Kubernetes cluster name from Datadog, which identifies the source cluster of the Kafka broker being monitored.
topic	Kafka topic name of the Kafka workload being monitored.

Kafka consumer offset rate

Metric

federatorai.kafka.consumer_offset_rate

The number of consumer offsets increased in the last 60 seconds.

Value

Kafka consumer offset rate.

Tags

Name	Description
kube_cluster	Kubernetes cluster name from Datadog, which identifies the source cluster of the Kafka consumer being monitored.
topic	Kafka topic name of the Kafka workload being monitored.
consumer_group	Kafka consumer group of the Kafka workload being monitored.

Cost Analysis Metrics

Recommendation for cloud providers and instances.

Cost of a recommended cloud instance

Metric

`federatorai.recommendation.instance`

Cost of a recommended cloud instance.

Value

Cost value.

Tags

Name	Description
<code>kube_cluster</code>	Kubernetes cluster name.
<code>time_interval</code>	Time interval over which the cost of the instance is calculated. The value is <i>30_days</i> .
<code>country</code>	Possible values are <i>All, Australia, Bahrain, Belgium, Brazil, Canada, China, Finland, France, Germany, India, Ireland, Japan, Milan, Netherlands, Norway, Republic of South Africa, Singapore, South Korea, Sweden, Switzerland, Taiwan, UK, USA, United Arab Emirates</i>
<code>display_name</code>	Instance display name, instance name, and brief information.
<code>instance_type</code>	Instance type from a cloud provider.
<code>ondemand_instance_master_num</code>	Number of on-demand instances for the master node.
<code>reserved_instance_master_num</code>	Number of reserved instances for the master node.
<code>provider</code>	Cloud provider.
<code>region</code>	Region where the instance is hosted.
<code>ondemand_instance_worker_num</code>	Number of on-demand instances for worker nodes.
<code>reserved_instance_worker_num</code>	Number of reserved instances for worker nodes.
<code>spot_instance_master_num</code>	Number of AWS spot instances for the master node.
<code>spot_instance_worker_num</code>	Number of AWS spot instance for the worker node.
<code>pricing_option</code>	Calculated price of an instance based on instance type. Possible values are <i>on-demand-instance</i> , <i>reserved-instance</i> , and <i>spot-instance</i> .

Cost analysis for a cloud instance

Metric

federatorai.cost_analysis.instance.cost

Cost analysis for a cloud instance.

Value

Cost value.

Tags

Name	Description
kube_cluster	Kubernetes cluster name.
host	Kubernetes node name.
kube_node_role	Kubernetes node role. Possible values are <i>master</i> or <i>worker</i> .
time_interval	Time interval over which the cost of the instance is calculated. The value is <i>30_days</i> .
instance_type	Node instance type provided by the cloud provider.
provider	Cloud provider. Possible values are <i>aws</i> , <i>gcp</i> , or <i>azure</i> .
region	Region where the instance is located.

Cost allocation for a namespace

Metric

federatorai.cost_analysis.namespace.cost

Cost of a namespace in a Kubernetes cluster.

Value

Cost value.

Tags

Name	Description
time_interval	Time interval over which the cost of the namespace is calculated. The value is <i>24_hours</i> .
kube_cluster	Kubernetes cluster name from Datadog, identifying the source cluster of the namespace being monitored.
kube_namespace	Kubernetes namespace from Datadog, identifying the source namespace of the application being monitored.
provider	Cloud provider. Possible values are <i>aws</i> , <i>gcp</i> , or <i>azure</i> .

Cost prediction for a namespace

Metric

federatorai.prediction.namespace.cost

Cost prediction for a namespace in a Kubernetes cluster.

Value

Cost value.

Tags

Name	Description
prediction_window	Time interval for which the cost prediction of a namespace is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from Datadog, identifying the source cluster of the namespace being monitored.
kube_namespace	Kubernetes namespace from Datadog, identifying the source namespace being monitored.
provider	Cloud provider. Possible values are <i>aws</i> , <i>gcp</i> , or <i>azure</i> .

Cost percentage prediction for a namespace

Metric

federatorai.prediction.namespace.cost_pct

Cost percentage prediction of a namespace in a Kubernetes cluster.

Value

Cost percentage.

Tags

Name	Description
prediction_window	Time interval for which the cost prediction of a namespace is calculated. Possible values are <i>24_hours</i> , <i>7_days</i> , or <i>30_days</i> .
kube_cluster	Kubernetes cluster name from Datadog, identifying the source cluster of the namespace being monitored.
kube_namespace	Kubernetes namespace from Datadog, identifying the source namespace being monitored.
provider	Cloud provider. Possible values are <i>aws</i> , <i>gcp</i> , or <i>azure</i> .

Datadog Federator.ai Metrics List and Updating Interval

Metric_name	Metric_type	Interval	Description
federatorai.integration.status	gauge	1 min	Integration status for showing Federator.ai health status.
federatorai.recommendation	gauge	For generic applications: 3 mins. For Kafka, 5 mins.	Recommended deployment, statefulset replicas.
federatorai.prediction.kafka	gauge	1 min	Workload prediction for Kafka metrics.
federatorai.kafka.broker_offset_rate	gauge	1 min	The delta of kafka.broker_offset timeseries per minute.
federatorai.kafka.consumer_offset_rate	gauge	1 min	The delta of kafka.consumer_offset timeseries per minute.
federatorai.prediction.node	gauge	1 hour, 6 hours, and 24 hours based on the prediction window	Workload prediction for a Kubernetes node.
federatorai.prediction.node.avg	gauge	1 min	The average value of workload predictions for a Kubernetes node for a prediction window.
federatorai.prediction.node.min	gauge	1 min	The minimum value of workload predictions for a Kubernetes node for a prediction window.
federatorai.prediction.node.max	gauge	1 min	The maximum value of workload predictions for a Kubernetes node for a prediction window.
federatorai.prediction.controller	gauge	1 hour, 6 hours, and 24 hours based on the prediction window	Workload prediction for a specific controller.

Metric_name	Metric_type	Interval	Description
federatorai.prediction.controller.avg	gauge	1 min	The average value of workload predictions for a specific controller for a prediction window.
federatorai.prediction.controller.min	gauge	1 min	The minimum value of workload predictions for a specific controller for a prediction window.
federatorai.prediction.controller.max	gauge	1 min	The maximum value of workload predictions for a specific controller for a prediction window.
federatorai.resource_planning.node	gauge	1 hour, 6 hours, and 24 hours based on the prediction window	Workload predictions for resource planning of a Kubernetes node.
federatorai.resource_planning.controller	gauge	1 hour, 6 hours, and 24 hours based on the prediction window	Workload predictions for resource planning of a Kubernetes controller.
federatorai.recommendation.instance	gauge	1 min	Cost of a recommended cloud instance.
federatorai.cost_analysis.instance.cost	gauge	1 min	Cost analysis for a cloud instance.
federatorai.cost_analysis.namespace.cost	gauge	1 min	Cost analysis for a namespace in a Kubernetes cluster.
federatorai.prediction.namespace.cost	gauge	1 min	Cost prediction for a namespace in a Kubernetes cluster.
federatorai.kubernetes.cpu.usage.total.controller	gauge	1 hour, 6 hours, or 24 hours	The number of cores (in millicores) used by the Kubernetes controller.
federatorai.kubernetes.memory.usage.controller	gauge	1 hour, 6 hours, or 24 hours	The memory usage (in bytes) of the Kubernetes controller.

Metric_name	Metric_type	Interval	Description
federatorai.kubernetes.cpu.usage.total.node	gauge	1 hour, 6 hours, or 24 hours	The number of cores (in millicores) used by the Kubernetes node.
federatorai.kubernetes.memory.usage.node	gauge	1 hour, 6 hours, or 24 hours	The memory usage (in bytes) of the Kubernetes node.