

Federator.ai 4.3 for OpenShift Installation Guide

(for Red Hat Marketplace)

Content

Overview	2
Federator.ai.....	2
Datadog Integration Workflows	2
Requirements and Recommended Resource Configuration	3
Platform	3
Federator.ai Resource Requirements	3
Federator.ai Version	3
Datadog Agent Version(reference).....	3
Persistent Volumes	3
Kafka.....	3
Federator.ai Installation and Configuration.....	4
Summary of Installation Steps	4
Pre-installation Check List.....	4
Before You Start.....	5
Installation	7
Configuration	13
Manage Federator.ai License Keycode	20
Apply Keycode.....	20
Delete Keycode	20
Activate Keycode	21
Appendix	23
Datadog Dashboards.....	23

Overview

Federator.ai

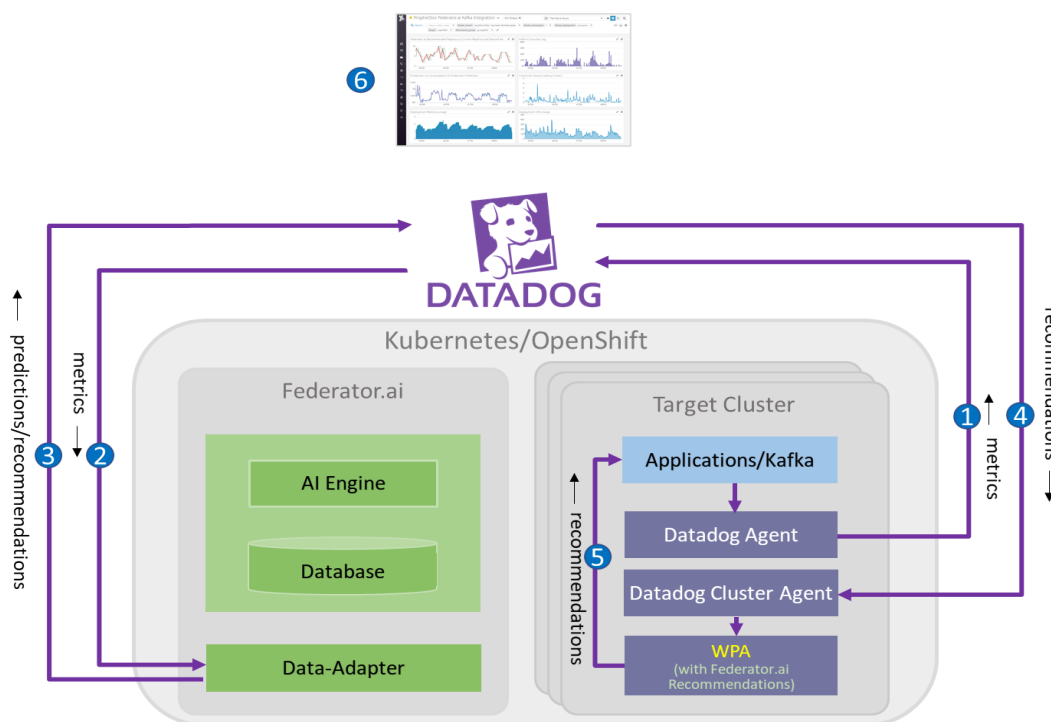
ProphetStor Federator.ai is an AI-based solution that helps enterprise manage, optimize, auto-scale resources for any applications on OpenShift. Using advanced machine learning algorithms to predict application workload, Federator.ai scales the right amount of resources at the right time for optimized application performance.

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

Datadog Integration Workflows

The following diagram shows how applications metrics are used by Federator.ai to predict workload and to automatically scale applications for better performance. Specifically,

- Datadog Agent sends cluster/applications metrics to Datadog Services
- Federator.ai's Data-adapter queries cluster/applications metrics from Datadog Services and forwards them to Federator.ai AI engine
- Data-adapter posts the prediction/recommendation/plan created by Federator.ai to Datadog Services
- Datadog Cluster Agent gets prediction/recommendation/plan from Datadog Services
- WPA applies plans and auto-scales applications
- Datadog Dashboard displays cluster/applications metrics and prediction/recommendation/plan by Federator.ai



Requirements and Recommended Resource Configuration

Platform

- OpenShift 4.4 and above

Federator.ai Resource Requirements

- Total Resource Requirements
 - 4 CPU cores
 - 4 GB Memory
 - StorageClass: 420GB (require ReadWriteMany access mode)
- Resource requirements for AI Engine
 - There must be at least one worker node with at least 2 CPU cores and 1 GB memory available
 - The 2 CPU cores and 1 GB memory are included in the total 4 CPU cores and 4 GB memory requirements

Federator.ai Version

- Version: Release 4.3
- 30 days trial license

Datadog Agent Version(reference)

- Datadog Agent version : v7.21.1
- Datadog Cluster Agent version : v1.7.0
- Datadog Watermark Pod Autoscaler version : v0.1.0
- kube-state-metrics : v1.5.0 (for OpenShift 3.11, Kubernetes 1.11 ~ 1.12)
v1.9.6 (for OpenShift 4.3/4.4/4.5, Kubernetes 1.13 ~ 1.18.x)

Persistent Volumes

- The StorageClass that provides the persistent volumes must support RWX (read-write many) access mode.
- It is recommended to use persistent volumes instead of using ephemeral storage to store the data in the production environment.

Kafka

- For Federator.ai's application-aware Kafka consumer resource/performance optimization feature, the following version of Kafka is supported :

Kafka operator version : Strimzi/kafka:0.17.0-kafka-2.4.0

Federator.ai Installation and Configuration

Summary of Installation Steps

- Step 0: Review pre-installation checklist items, make sure the environment and required information are ready.
- Step 1: Collect information on Datadog Cloud Service account, API Key, Application Key. Instructions are provided below.
- Step 2: Install and configure Datadog Agent/Cluster Agent if they have not been installed. Please follow Datadog documentation on how to install Datadog Agent and Cluster Agent.
- Step 3: Install Federator.ai.
- Step 4: Configure Federator.ai Data Adapter for Datadog.
- Step 5: Optionally install Datadog WPA and apply WPA autoscaling CR if using Datadog WPA for autoscaling.
- Step 6: Review installation result on Datadog Cloud Dashboard.

Pre-installation Check List

OpenShift:

#	Check list Item	Requirement	Details
1	What is the OpenShift version?	4.4 and above	Use the command below to get OpenShift version: ~# oc version Client Version: 4.4.5 Server Version: 4.4.5 Kubernetes Version: v1.17.1
2	StorageClass and Persistent Volumes requirement	StorageClass supports ReadWriteMany access mode. Available storage size is larger than 430GB.	Minimum storage size for Federator.ai Release 4.3 is 430GB, including database, data, and logs.
3	OpenShift cluster CPU/memory requirement	Minimum CPU/mem/storage: <ul style="list-style-type: none">- CPU: 4 Cores- Memory: 4 GB- Storage Class Capacity: 430GB At least one worker node with <ul style="list-style-type: none">- CPU: 2 Cores- Memory: 1GB	To be able to run AI Engine pod, there must be at least one worker node that has more than 2 CPU cores and 1 GB memory available. 2 CPU Cores and 1GB for AI Engine are included in the total 4 CPU Cores and 4GB memory requirements.
4	Will there be a resource quota imposed for the namespace where Federator.ai is installed?	CPU/mem request quota should be more than minimum resource requirement <ul style="list-style-type: none">- CPU: 4 Cores- Memory: 4 GB	The CPU/memory required for Federator.ai depends on the number of clusters and applications being monitored/managed. Suggestion for initial namespace quota is <ul style="list-style-type: none">- CPU 8 cores- Memory 12G The quota could be adjusted if number of managed clusters/applications increases. Use the command to get namespace resource quota ~# oc get resourcequota --all-namespaces

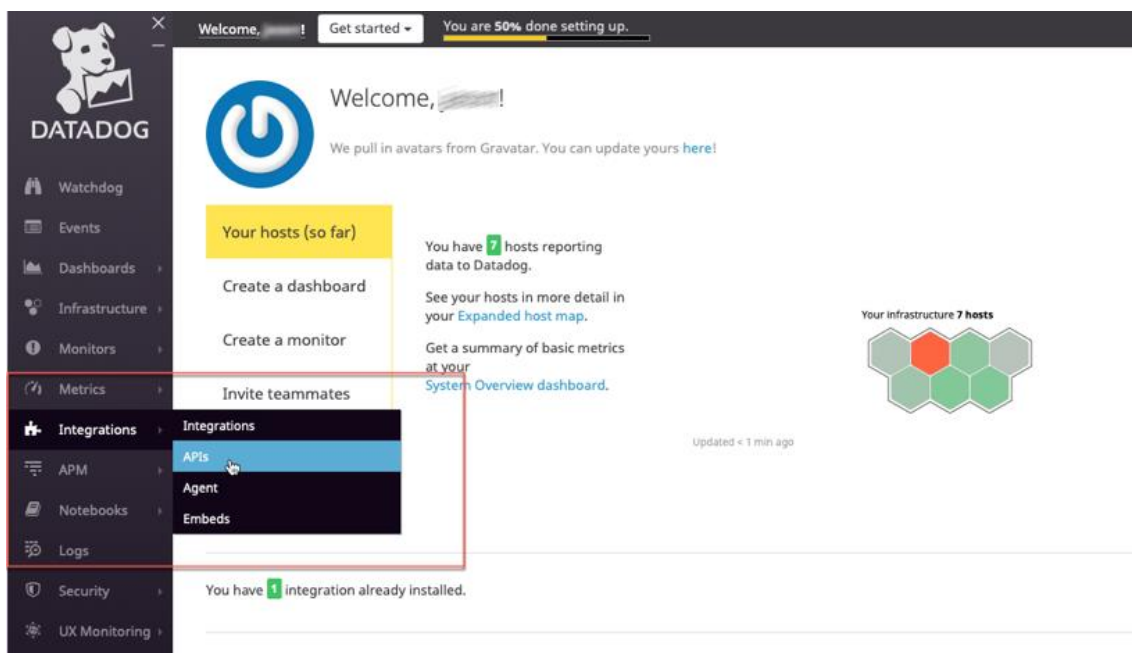
Datadog Agent:

#	Check list Item	Requirement	Details
5	Is Datadog Agent installed?	Datadog Agent is mandatory	OpenShift resources and workload metrics are collected by Datadog Agent.
6	Is Datadog Cluster Agent installed?	Cluster Agent is mandatory for HPA autoscaling feature	Cluster Agent provides metrics to HPA Autoscaler for autoscaling.
7	Is Datadog WPA controller installed?	Datadog WPA is required if auto-scaling is done by WPA	Datadog WPA is the HPA Autoscaler developed by Datadog. Users can use Datadog WPA or Kubernetes native HPA to do autoscaling.
8	Datadog Kafka Consumer integration is enabled?	Datadog Kafka Consumer integration is mandatory if user wants to use Kafka optimization feature	Use the command to confirm Kafka integration is enabled <pre>~# oc exec <datadog-agent-pod> -n <datadog-agent-namespace> -- agent integration show datadog-kafka-consumer</pre> <p>Refer to https://www.datadoghq.com/blog/monitor-kafka-with-datadog/ for Kafka Consumer integration installation</p>
9	Datadog account API key	API key is mandatory for connecting Datadog Service	Follow the steps described in the “Before You Start” session to obtain the API key.
10	Datadog account Application key	Application key is mandatory for connecting Datadog Service	Follow the steps described in the “Before You Start” session to obtain the Application key.
11	DD_TAGS with value="kube_cluster:<cluster_name>" is configured for Datadog Agent?	DD_TAGS with value="kube_cluster:<cluster_name>" is required for Federator.ai to identify OpenShift cluster	Use the command below to confirm DD_TAGS with value="kube_cluster:<cluster_name>" is configured <pre>~# oc get daemonset <datadog_agent_daemonset_name> -n <datadog_agent_namespace> -o yaml</pre> <p>...</p> <pre>- name: DD_TAGS value: kube_cluster:my-cluster</pre>

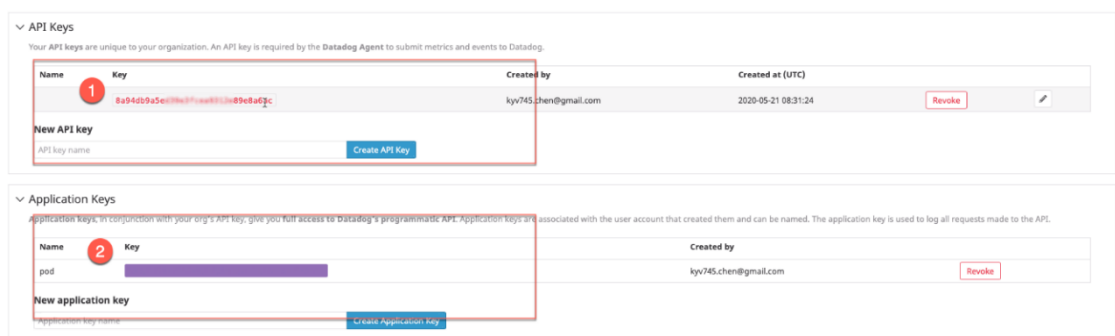
Before You Start

- The admin role for installing Federator.ai is "Cluster Admin."
- Datadog agent must be ready if Federator.ai runs in the same OpenShift cluster that is being monitored.
- Obtain Datadog account APIKey, APPKey.
 1. A Datadog account is required for connecting and using Datadog Cloud Service. If you don't have an account, visit Datadog website and sign up for a free trial account.
<https://www.datadoghq.com/>
 2. Log in Datadog Cloud Service with your account and get an API key and Application key for using Datadog API

https://docs.datadoghq.com/account_management/api-app-keys/



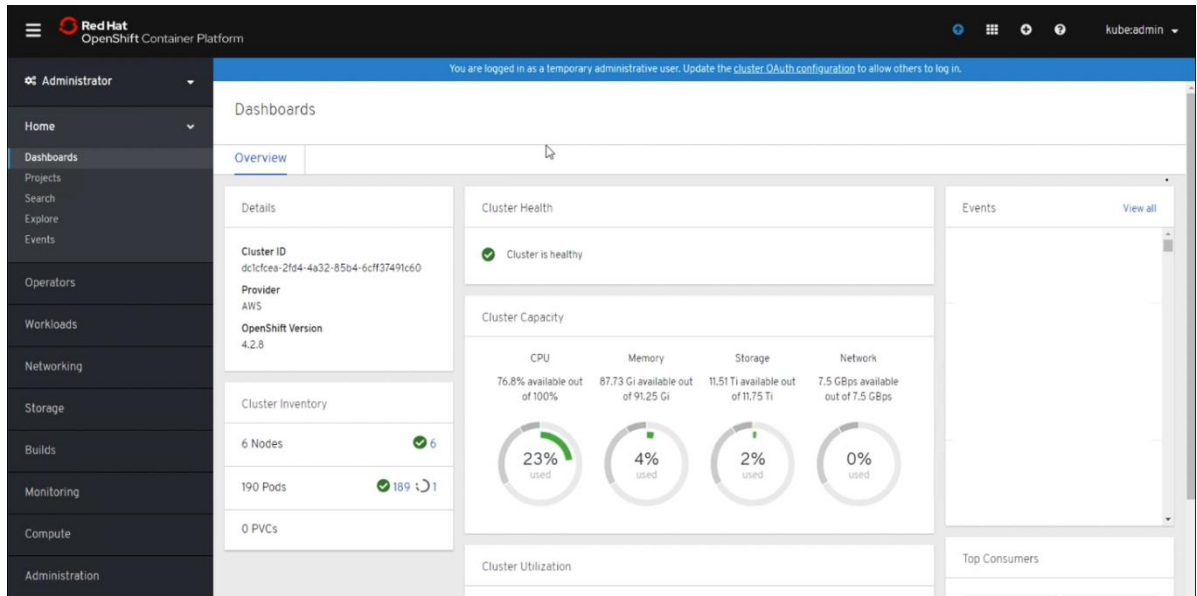
Copy the API Key and Application Key for Federator.ai Data-Adapter configuration



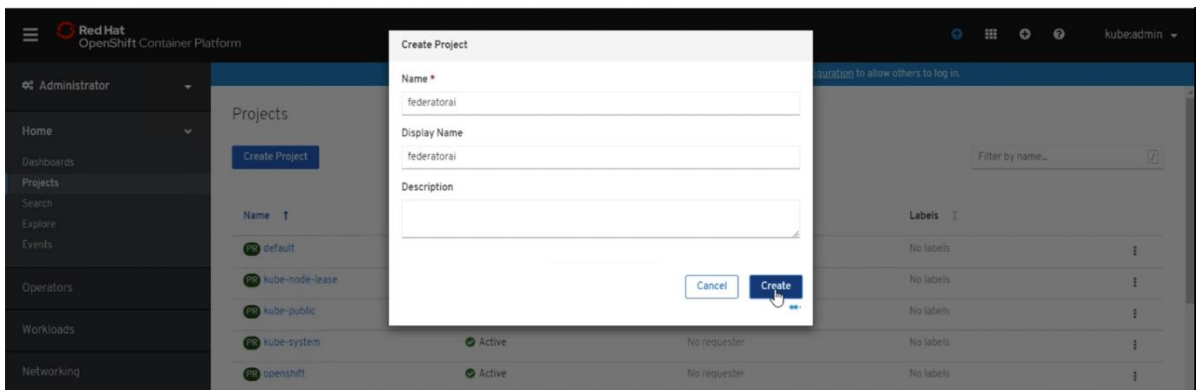
- To install Federator.ai from Red Hat Marketplace, you need to have a Red Hat account and log in Red Hat Marketplace at <https://marketplace.redhat.com/api-security/en-us/login/landing>. If you don't have a Red Hat account, register an account at <https://marketplace.redhat.com/en-us/registration/redhat-marketplace>.
- Red Hat Marketplace integration with OpenShift provides easy install of purchases and trials on Red Hat Marketplace. This requires registering your OpenShift clusters to your Red Hat account. The procedure of adding clusters is at <https://marketplace.redhat.com/en-us/workspace/clusters/add/register>. OpenShift 4.2 and above clusters are supported.

Installation

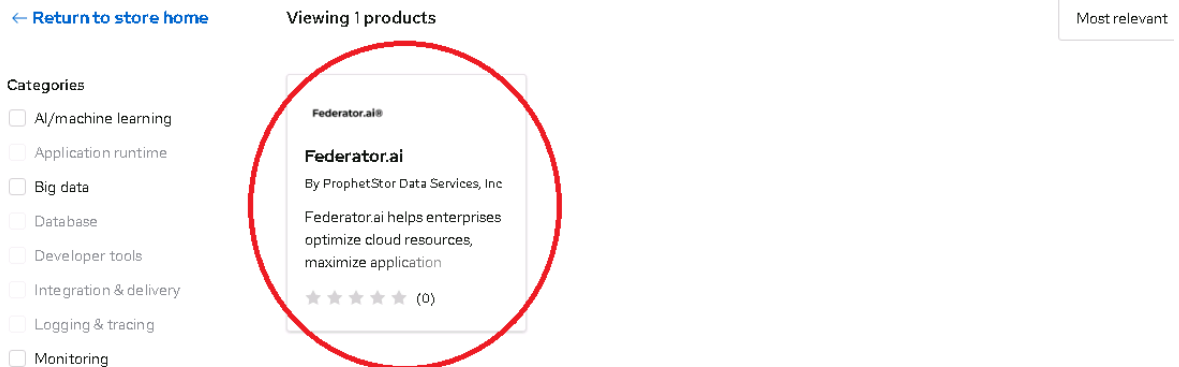
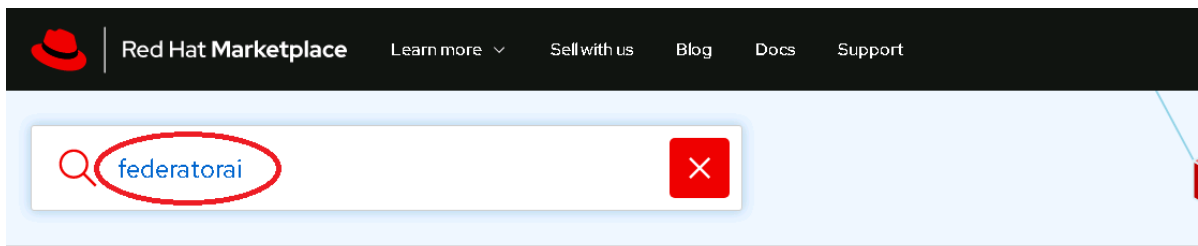
1. Log in OpenShift administration console as cluster admin



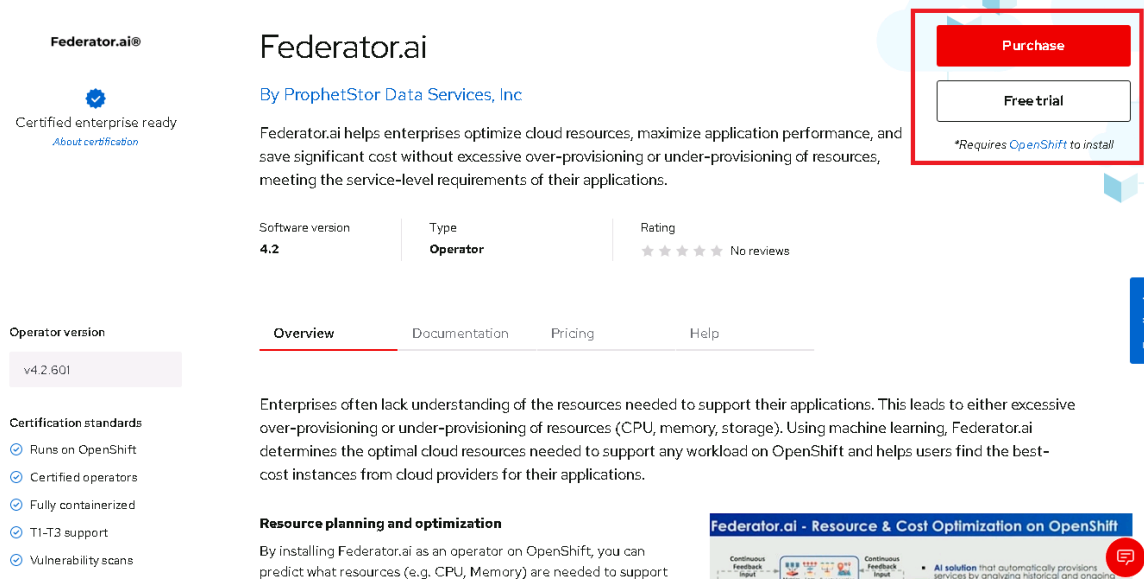
2. Go to “Projects” page and create a new project. For example, “federatorai” for Federator.ai



- Go to Red Hat Marketplace and use keyword “federator” to search for products. Federator.ai Operator will be listed for installation.



- Click “Purchase” or “Free Trial” to initiate the installation.



- Click the “Install Operator” and select the namespace “federatorai” to start install the software. It’s recommended to keep all the default options.

Federator.ai®

Federator.ai


Free 30-day Trial | Version 4.2

Overview

Operators

Documentation

Support



You haven't installed any Operators

You're all ready to go, just click "Install Operator" to get started.

Install Operator

Update channel

Operators are organized into packages and streams of updates called "channels". If an operator is available through multiple channels, you can choose which one you want to subscribe to. [Learn more](#)

☒ stable

Approval strategy

Automatic updates keep the operator and any instances on the cluster up to date. Manual updates require approval and are done via OpenShift console or CLI. [Learn more](#)

☒ Automatic
☐ Manual

Target clusters

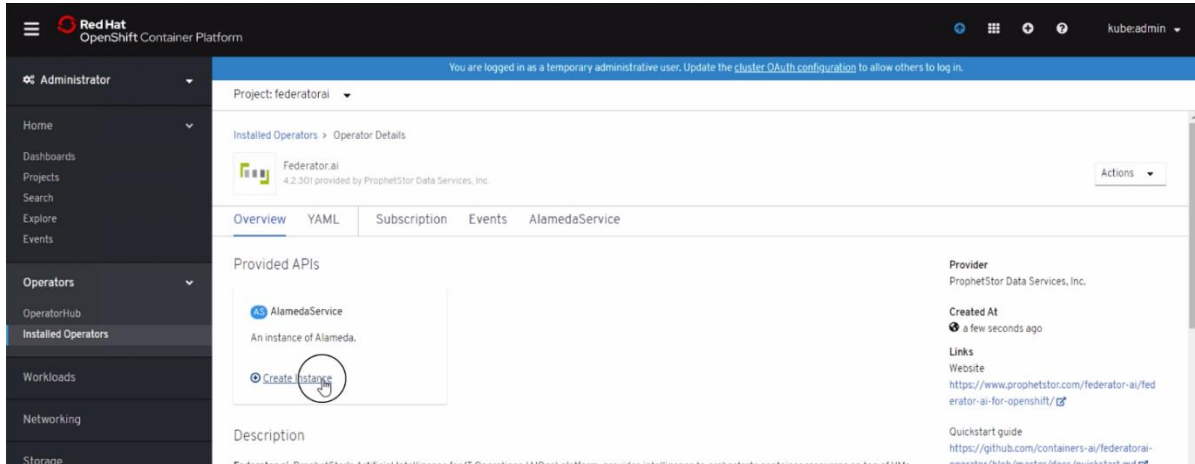
Choose clusters where you want to install and manage this operator. Then select the Namespace scope for each cluster you are installing into. [Learn more](#)

<input checked="" type="checkbox"/>	Name	Platform	Namespace Scope
<input checked="" type="checkbox"/>	jason-test	Libvirt	<div>federatorai</div>

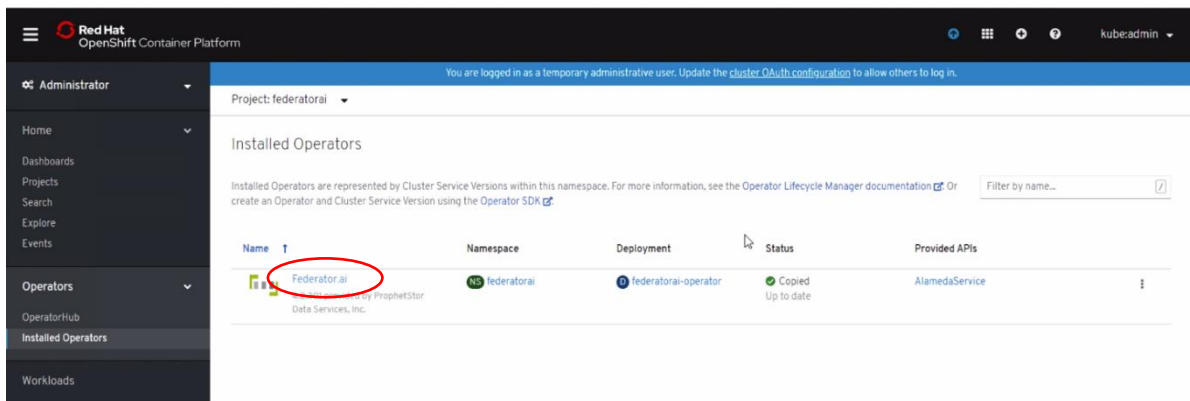
Cancel

Install

- It will take a few minutes to pull the software images and install Federator.ai in the “federatorai” project. Once the installation completes, Federator.ai will show up on “Installed Operators” page

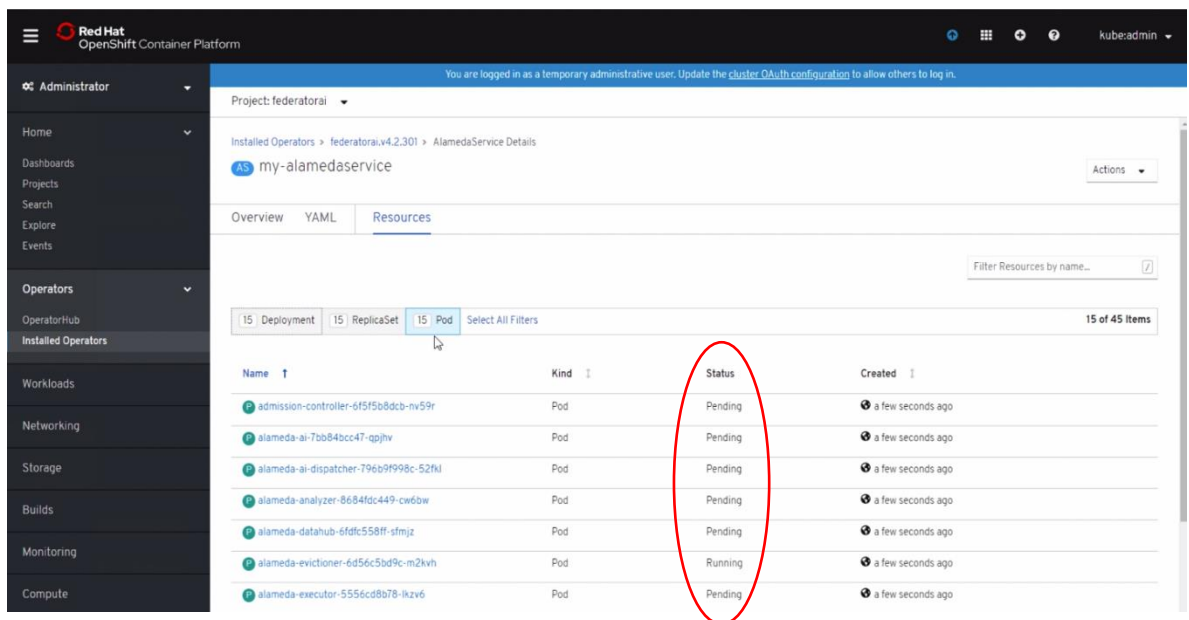
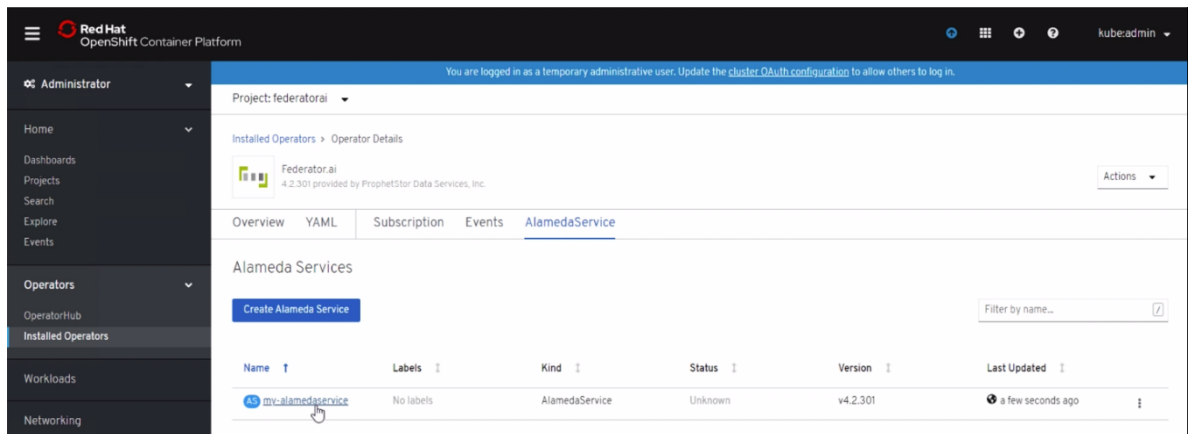


- Click “Federator.ai” to see the details of Federator.ai Operator. After Federator.ai Operator is installed, configure Federator.ai Operator to install the rest of Federator.ai components by creating an “AlamedaService” resource. You can configure with your persistent volumes (it is recommended) or leave the default ephemeral storage.

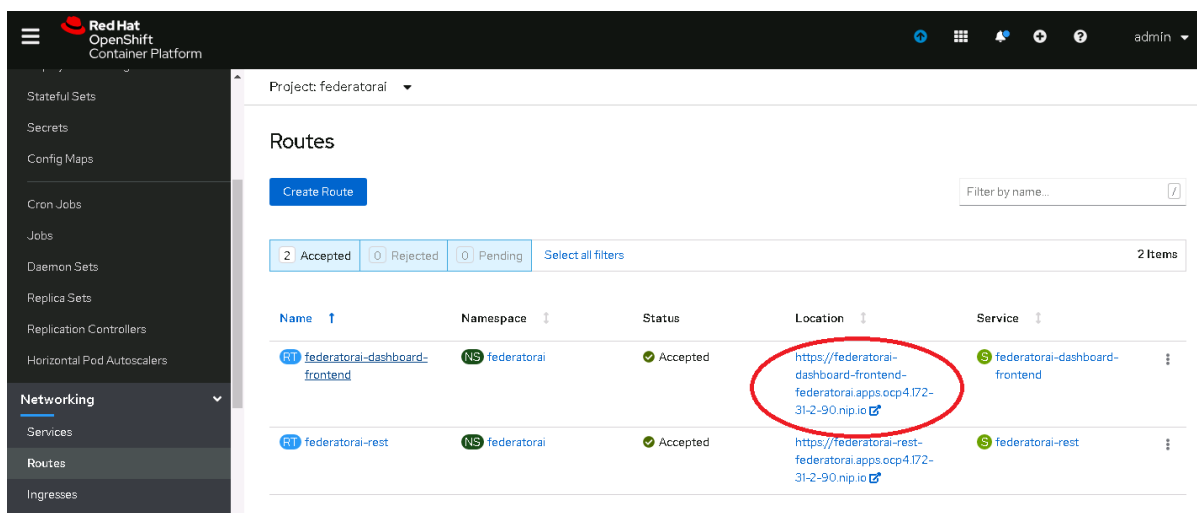


```
1 apiVersion: federatorai.containers.ai/v1alpha1
2 kind: AlamedaService
3 metadata:
4   name: my-alamedaservice
5   namespace: federatorai
6 spec:
7   keycode:
8     codeNumber: D3JXN-LIFTQ-KQEZ3-WZBNI-DA3WZ-A7HKQ
9   enableExecution: false
10  version: v4.3.956
11  storages:
12    - usage: log
13      type: ephemeral
14    - usage: data
15      type: ephemeral
16
```

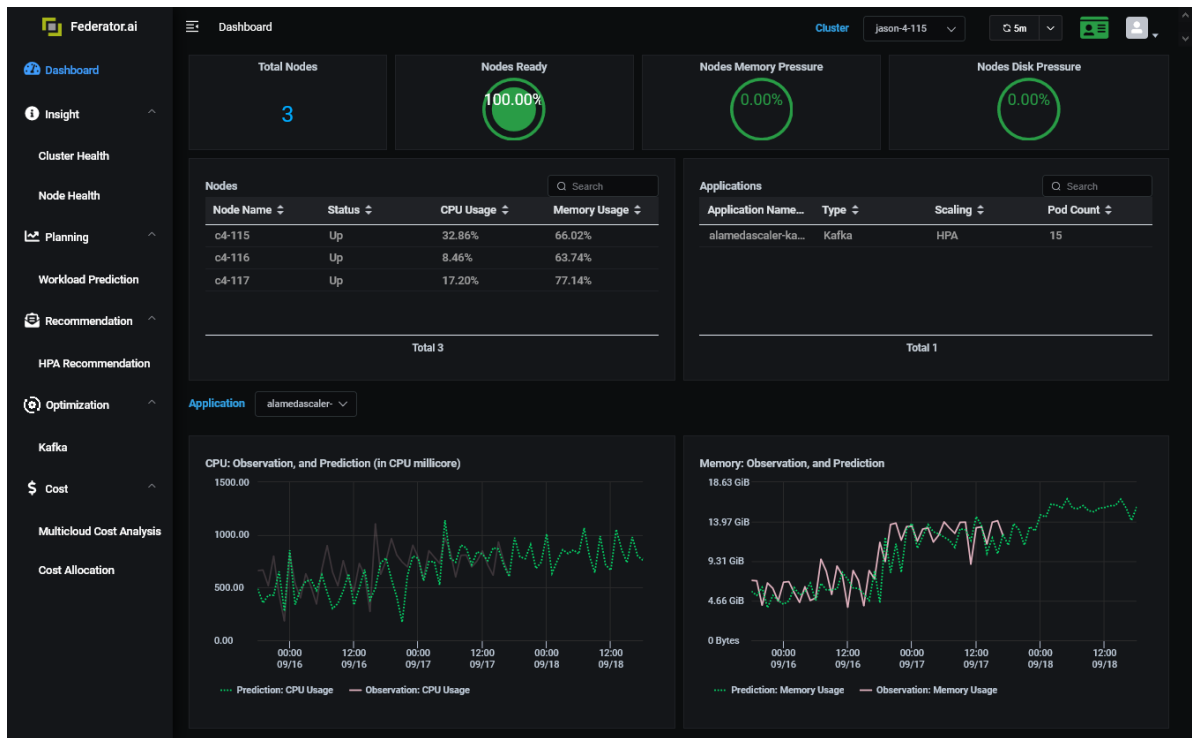
8. Click “my-almatedaservice” resource to see the details. “Resources” view shows the status of the rest of Federator.ai components. When the status of all components are “Running,” Federator.ai installation is complete successfully.



9. The URL of Federator.ai GUI can be found at “Projects -> federatorai -> Route -> federatorai-dashboard-frontend”



10. Log in Federator.ai GUI with the default account id/password, “admin/admin”. The portal page of Federator.ai GUI shows the summary of Federator.ai and OpenShift cluster information.



Configuration

Federator.ai supports two types of applications, Kafka consumers and generic applications. The configuration procedure illustrated below uses one Kafka and one generic application (NGINX) as examples.

1. Prepare your Kafka configuration information if you will configure Federator.ai to manage Kafka consumers. This step is optional.

- Get Kafka connection string (e.g., “my-cluster-kafka-brokers.myproject:9092”)

```
~# oc get svc -n myproject
my-cluster-kafka-bootstrap      ClusterIP    10.107.237.39    <none>
9091/TCP,9092/TCP,9093/TCP,9404/TCP    15d
my-cluster-kafka-brokers        ClusterIP    None             <none>
9091/TCP,9092/TCP,9093/TCP          15d
my-cluster-kafka-exporter        ClusterIP    10.98.96.53      <none>
15d                                9404/TCP
my-cluster-zookeeper-client      ClusterIP    10.110.115.16    <none>
9404/TCP,2181/TCP                  15d
my-cluster-zookeeper-nodes       ClusterIP    None             <none>
2181/TCP,2888/TCP,3888/TCP         15d
```

- Find topic ID of interest (e.g., “topic001”)

```
~# oc get pod -n myproject
my-cluster-entity-operator-995df8959-vkwrn    3/3    Running    0    6d
my-cluster-kafka-0                            2/2    Running    0    3d5h
my-cluster-kafka-1                            2/2    Running    0    12h
my-cluster-kafka-2                            2/2    Running    0    4d3h
my-cluster-kafka-exporter-6b84688dbd-4dgv2    1/1    Running    57   15d
my-cluster-zookeeper-0                        2/2    Running    0    6d
my-cluster-zookeeper-1                        2/2    Running    0    15d
my-cluster-zookeeper-2                        2/2    Running    0    15d
producer-topic0001-8c8c4f5-xfdz7             1/1    Running    0    43h
strimzi-cluster-operator-77555d4b69-j4975     1/1    Running    1    6d

~# oc -n myproject exec my-cluster-kafka-0 -c kafka -- bin/kafka-topics.sh --bootstrap-server my-cluster-kafka-bootstrap:9092 --list
OpenJDK 64-Bit Server VM warning: If the number of processors is expected to increase from one, then you should configure the number of parallel GC threads appropriately using -XX:ParallelGCThreads=N
__consumer_offsets
topic0001
topic0002
```

- Find Consumer Group ID (e.g., “group0001”)

```
~# oc get pod -n myproject
my-cluster-entity-operator-995df8959-vkwrn    3/3    Running    0    6d
my-cluster-kafka-0                            2/2    Running    0    3d5h
my-cluster-kafka-1                            2/2    Running    0    12h
my-cluster-kafka-2                            2/2    Running    0    4d3h
...

~# oc -n myproject exec my-cluster-kafka-0 -c kafka -- bin/kafka-consumer-groups.sh --bootstrap-server my-cluster-kafka-bootstrap:9092 --list
OpenJDK 64-Bit Server VM warning: If the number of processors is expected to increase from one, then you should configure the number of parallel GC threads appropriately using -XX:ParallelGCThreads=N
group0001
group0002
```

2. Configure Federator.ai Data Adapter to connect to Datadog Service

- Use the command to download configuration helper scripts from Github

```
~# curl https://raw.githubusercontent.com/containers-ai/federatorai-operator/master/deploy/federatorai-launcher.sh | bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 13270  100 13270    0     0  21667      0 --:--:-- --:--:-- --:--:-- 21683
Please input Federator.ai version tag (e.g., v4.2.755): v4.3.datadog

Downloading scripts ...

Downloading Federator.ai CR yamls ...

Downloading Federator.ai operator yamls ...
Done
Do you want to use a private repository URL? [default: n]: n
Do you want to launch the Federator.ai installation script? [default: y]: n

Downloaded files are located under /tmp/federatorai-scripts/v4.3.datadog
```

- Use Data Adapter configuration helper script in “/tmp/federatorai-scripts/v4.3.datadog/scripts”

```
~# ls -l /tmp/federatorai-scripts/v4.3.datadog
-rw-r--r-- 1 root root 11968 Aug 19 17:49 email-notifier-setup.sh
-rw-r--r-- 1 root root 31708 Aug 19 17:49 federatorai-setup-for-datadog.sh
-rw-r--r-- 1 root root 42722 Aug 19 17:49 install.sh
-rw-r--r-- 1 root root 4949 Aug 19 17:49 node-label-assignor.sh
-rw-r--r-- 1 root root 34819 Aug 19 17:49 planning-util.sh
-rw-r--r-- 1 root root 49215 Aug 19 17:49 preloader-util.sh
-rw-r--r-- 1 root root 1721 Aug 19 17:49 prepare-private-repository.sh
-rw-r--r-- 1 root root 4433 Aug 19 17:49 uninstall.sh
```

- Change file permission to be executable

```
~# chmod +x federatorai-setup-for-datadog.sh
```

- Run the configuration helper script and follow the instructions to input configuration parameters

```
~# ./federatorai-setup-for-datadog.sh
```

```
~# cd /tmp/federatorai-scripts/v4.3.datadog/scripts
~# ./federatorai-setup-for-datadog.sh
Checking environment version...
...Passed
You are connecting to cluster: https://172.31.3.34:8443
Do you want to reconfigure Datadog API & Application keys? [default: n]: n

Do you want to configure alamedascaler for generic application? [default: y]: y

Getting generic application info... No.1
Input alamedascaler name []: nginx-sample
Input cluster name []: k8s-4-205

Getting controller info for nginx-sample alamedascaler... No.1
Input target app kind (Deployment/DeploymentConfig/StatefulSet[]): Deployment
Input target app namespace []: nginx-sample
Input Deployment name []: nginx-sample
Do you want to enable HPA recommendation? [default: y]:
Input minimum replicas number []: 1
```

```

Input maximum replicas number []: 5

Do you want to add another controller in nginx-sample alamedascaler? [default: n]: y

Getting controller info for nginx-sample alamedascaler... No.2
Input target app kind (Deployment/DeploymentConfig/StatefulSet)[]: Deployment
Input target app namespace []: nginx-sample
Input Deployment name []: nginx-sample-1
Do you want to enable HPA recommendation? [default: y]:
Input minimum replicas number []: 1
Input maximum replicas number []: 3

Do you want to add another controller in nginx-sample alamedascaler? [default: n]:

Do you want to add another generic application? [default: n]:

Do you want to configure alamedascaler for kafka? [default: y]: : y

Getting Kafka info... No.1
Input alamedascaler name []: kafka-consumer
Input cluster name []: k8s-4-205

Getting controller info for kafka-consumer alamedascaler... No.1
Input Kafka exporter namespace []: myproject
Input Kafka consumer group kind (Deployment/DeploymentConfig/StatefulSet) []: Deployment
Input Kafka consumer group kind name []: consumer1-topic0001-group-0001
Input Kafka consumer group namespace []: myproject
Input Kafka consumer topic name []: topic0001

You can use Kafka command-line tool 'kafka-consumer-group.sh' (download separately or
enter into a broker pod, in /bin directory) to list consumer groups.
e.g.: "/bin/kafka-consumer-groups.sh --bootstrap-server <kafka-bootstrap-service>:9092 --
describe --all-groups --members"
The first column of output is the 'kafkaConsumerGroupId'.
Input Kafka consumer group id []: group0001
Input Kafka consumer minimum replica number []: 1
Input Kafka consumer maximum replica number []: 3

Do you want to add another controller in kafka-consumer alamedascaler? [default: n]: y

Getting controller info for kafka-consumer alamedascaler... No.2
Input Kafka exporter namespace []: myproject
Input Kafka consumer group kind (Deployment/DeploymentConfig/StatefulSet) []: Deployment
Input Kafka consumer group kind name []: consumer2-topic0002-group-0002
Input Kafka consumer group namespace []: myproject
Input Kafka consumer topic name []: topic0002

You can use Kafka command-line tool 'kafka-consumer-group.sh' (download separately or
enter into a broker pod, in /bin directory) to list consumer groups.
e.g.: "/bin/kafka-consumer-groups.sh --bootstrap-server <kafka-bootstrap-service>:9092 --
describe --all-groups --members"
The first column of output is the 'kafkaConsumerGroupId'.
Input Kafka consumer group id []: group0002
Input Kafka consumer minimum replica number []: 1
Input Kafka consumer maximum replica number []: 5

Do you want to add another controller in kafka-consumer alamedascaler? [default: n]:

Do you want to add another Kafka set? [default: n]:

Updating Federator.ai data adapter configmap...
Warning: oc apply should be used on resource created by either oc create --save-config or
oc apply
configmap/federatorai-data-adapter-config configured

...Done.

Adding alamedascaler for generic applications...

```



```

alamedascaler.autoscaling.containers.ai/nginx-sample created

...Done.

Adding alamedascaler for Kafka...
alamedascaler.autoscaling.containers.ai/nginx-sample unchanged
alamedascaler.autoscaling.containers.ai/kafka-consumer created

...Done.

Restarting Federator.ai data adapter...
pod "federatorai-data-adapter-b7d9db494-s9g6v" deleted

Checking pods...

All federatorai pods are ready.

...Done.

Setup Federator.ai for Datadog successfully
Yaml files generated are under ./config_result

```

#notes-1: input cluster name must match with the <cluster_name> configured in Datadog Agent DD_TAGS (value="kube_cluster:<cluster_name>")

- Verify configuration result

```

~# ls -l config-result/
-rw-r--r-- 1 root root 35666  9月 16 12:06 adapter-configmap.yaml
-rw-r--r-- 1 root root   912  9月 16 12:06 kafka-consumer.yaml
-rw-r--r-- 1 root root   690  9月 16 12:06 nginx-sample.yaml

```

kafka-consumer.yaml

```

~# cat config-result/kafka-consumer.yaml
apiVersion: autoscaling.containers.ai/v1alpha2
kind: AlamedaScaler
metadata:
  name: kafka-consumer
  namespace: federatorai
spec:
  clusterName: k8s-4-205
  controllers:
    - type: kafka
      enableExecution: false
      scaling: hpa
      kafka:
        exporterNamespace: myproject
        consumerGroup:
          namespace: myproject
          name: consumer1-topic0001-group-0001
          kind: Deployment
          topic: topic0001
          groupId: group0001
        hpaParameters:
          maxReplicas: 3
          minReplicas: 1
    - type: kafka
      enableExecution: false
      scaling: hpa
      kafka:
        exporterNamespace: myproject
        consumerGroup:
          namespace: myproject
          name: consumer2-topic0002-group-0002

```

```
kind: Deployment
topic: topic0002
groupId: group0002
hpaParameters:
  maxReplicas: 5
  minReplicas: 1
```

nginx-sample.yaml

```
~# cat config-result/nginx-sample.yaml
apiVersion: autoscaling.containers.ai/v1alpha2
kind: AlamedaScaler
metadata:
  name: nginx-sample
  namespace: federatorai
spec:
  clusterName: k8s-4-205
  controllers:
    - type: generic
      enableExecution: false
      scaling: hpa
      generic:
        target:
          namespace: nginx-sample
          name: nginx-sample
          kind: Deployment
          hpaParameters:
            maxReplicas: 5
            minReplicas: 1
    - type: generic
      enableExecution: false
      scaling: hpa
      generic:
        target:
          namespace: nginx-sample
          name: nginx-sample-1
          kind: Deployment
          hpaParameters:
            maxReplicas: 3
            minReplicas: 1
```

3. (Optional) Install Datadog Watermark Pod Autoscaler Controller if you enable HPA autoscaling and would like to use WPA to do autoscaling

- Download Datadog WPA package

```
~# wget https://github.com/DataDog/watermarkpodautoscaler/archive/master.zip
~# unzip master.zip
```

- Install Watermark Pod Autoscaler controller
WPA Helm Chart package requires using 'helm' to install. If you don't have 'helm' installed, use the following command to install.

```
~# curl -L https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
```

- Set up environment variables and then use 'helm' command to install WPA

```
~# DD_NAMESPACE="default"
~# DD_NAMEWPA="wpacontroller"
~# helm install $DD_NAMEWPA -n $DD_NAMESPACE ./chart/watermarkpodautoscaler
```

```

~# pwd
/root/datadog_wpa/watermarkpodautoscaler
~# DD_NAMESPACE="default"
~# DD_NAMEWPA="wpacontroller"
~# helm install $DD_NAMEWPA -n $DD_NAMESPACE ./chart/watermarkpodautoscaler
~# oc get pods -n default

```

NAME	READY	STATUS	RESTARTS	AGE
datadog-monitoring-6lckr	2/2	Running	0	2d19h
datadog-monitoring-cluster-agent-7d79559979-cnjhj	1/1	Running	0	2d19h
datadog-monitoring-dwq7f	2/2	Running	0	2d19h
datadog-monitoring-hlm8x	2/2	Running	0	2d19h
datadog-monitoring-kube-state-metrics-765978777d-b5dnq	1/1	Running	0	6d3h
nfs-client-provisioner-7cd5f68cf7-cfqqb	1/1	Running	0	6d3h
wpacontroller-watermarkpodautoscaler-68484f8dd4-zxm22	1/1	Running	18	6d3h

- Download WPA pod autoscaler CR yaml file

```

~# wget
https://github.com/DataDog/watermarkpodautoscaler/blob/master/deploy/crds/datadoghq.com_watermarkpodautoscalers_cr.yaml

```

- Edit datadoghq.com_watermarkpodautoscalers_cr.yaml
Configure WPA to auto-scale Kafka consumer group and generic application (NGINX)

```

~# mv datadoghq.com_watermarkpodautoscalers_cr.yaml wpa.yaml
~# vi wpa.yaml
apiVersion: datadoghq.com/v1alpha1
kind: WatermarkPodAutoscaler
metadata:
  name: consumer
  namespace: myproject
spec:
  # Add fields here
  # algorithm must be average
  algorithm: average
  maxReplicas: 10
  minReplicas: 1
  tolerance: 0.01
  downscaleForbiddenWindowSeconds: 300
  upscaleForbiddenWindowSeconds: 15
  scaleUpLimitFactor: 90
  scaleDownLimitFactor: 90
  scaleTargetRef:
    kind: Deployment
    apiVersion: apps/v1
    name: consumer
  readinessDelay: 10
  metrics:
    # Resource or External type supported
    # Example usage of External type
    - type: External
      external:
        # do not edit highWatermark, and lowWatermark
        # highWatermark and lowWatermark must be 1
        highWatermark: "1"
        lowWatermark: "1"
        metricName: federatorai.recommendation
        metricSelector:
          matchLabels:
            resource: replicas
            kube_cluster: k8s-4-205 ← see below #notes-1 for more details
            kube_deployment: consumer
            kube_namespace: myproject

```

```

# Example usage of Resource type
# - type: Resource
#   resource:
#     highWatermark: "50"
#     lowWatermark: "10"
#     name: cpu
#     metricSelector:
#       matchLabels:
#         foo: bar
---
apiVersion: datadoghq.com/v1alpha1
kind: WatermarkPodAutoscaler
metadata:
  name: nginx-sample
  namespace: nginx-sample
spec:
  # Add fields here
  # algorithm must be average
  algorithm: average
  maxReplicas: 5
  minReplicas: 1
  tolerance: 0.01
  downscaleForbiddenWindowSeconds: 300
  upscaleForbiddenWindowSeconds: 15
  scaleUpLimitFactor: 90
  scaleDownLimitFactor: 90
  scaleTargetRef:
    kind: Deployment
    apiVersion: apps/v1
    name: nginx-sample
  readinessDelay: 10
  metrics:
    # Resource or External type supported
    # Example usage of External type
    - type: External
      external:
        # do not edit highWatermakr, and lowWatermark
        # highWatermark and lowWatermark must be 1
        highWatermark: "1"
        lowWatermark: "1"
        metricName: federatorai.recommendation
        metricSelector:
          matchLabels:
            resource: replicas
            kube_cluster: k8s-4-205 ← see below #notes-1 for more details
            kube_deployment: nginx-sample
            kube_namespace: nginx-sample

```

#notes-1: "kube_cluster" value must match with DD_TAGS (value="kube_cluster:<cluster_name>") configured in Datadog Agent (datadog-values.yaml)

- Deploy WPA and confirm the status

```
~# oc apply -f wpa.yaml
```

Managing Federator.ai License Keycode

Federator.ai uses a keycode for license control. A 30-day trial keycode is installed by default. It is required to replace it with a valid keycode from ProphetStor in order to continue using Federator.ai after the 30-day trial.

The keycode operations are done by editing the “AlamedaService” CR which is created during Federator.ai installation.

Applying A New Keycode

1. Get the “AlamedaService” CR name

```
~# oc get alamedaservice --all-namespaces
NAMESPACE      NAME                               EXECUTION  VERSION  PROMETHEUS
AGE
federatorai    my-alamedaservice                false      v4.3.958  https://prometheus-
k8s.openshift-monitoring:9091  45d
```

2. Edit the “AlamedaService” CR

```
~# oc edit alamedaservice my-alamedaservice -n <namespace>
```

3. Go to “keycode:” section, replace the value of “codeNumber” with the new keycode and then save the change

```
apiVersion: federatorai.containers.ai/v1alpha1
kind: AlamedaService
metadata:
  name: my-alamedaservice
.....
spec:
  .....
  keycode:
    codeNumber: K4AM0C4TSDXXXXXXXXXXXXXXXXXXXXX
```

Deleting An Existing Keycode

1. Get the “AlamedaService” CR name

```
~# oc get alamedaservice --all-namespaces
NAMESPACE      NAME                               EXECUTION  VERSION  PROMETHEUS
AGE
federatorai    my-alamedaservice                false      v4.3.958  https://prometheus-
k8s.openshift-monitoring:9091  45d
```

2. Edit the “AlamedaService” CR

```
~# oc edit alamedaservice my-alamedaservice
```

3. Go to “keycode:” section, delete the keycode from “codeNumber” and then save the change

```
apiVersion: federatorai.containers.ai/v1alpha1
kind: AlamedaService
metadata:
  name: my-almadaservice
.....
spec:
  .....
  keycode:
    codeNumber:
```

Activating A Keycode

1. Get the “AlamedaService” CR name

```
~# oc get alamedaservice --all-namespaces
NAMESPACE          NAME                               EXECUTION   VERSION   PROMETHEUS
AGE
federatorai        my-almadaservice                 false       v4.3.958  https://prometheus-
k8s.openshift-monitoring:9091  45d
```

2. Edit the “AlamedaService” CR

```
~# oc edit alamedaservice my-almadaservice
```

3. Go to “status.keycodeStatus:” section, copy the value of “registrationData” and email it to register@prophetstor.com.

```
apiVersion: federatorai.containers.ai/v1alpha1
kind: AlamedaService
metadata:
  name: my-almadaservice
.....
status:
  .....
  keycodeStatus:
    codeNumber: K4AM0C4TSDXXXXXXXXXXXXXXXXXXXXXQ
    lastErrorMessage: ""
    registrationData: H4sICAavJl8C/2ZlZGFpLXJlZ2RhGEudGd6A03ad1DTZxjA8R9IoMoe
KkMhZSiKSAYgG0IwjBBkb4kRagZJwwa3oAiKUPYoSB0E0YooKquADCGmgkiQCigqVoYgURCFiEJtr3
wegjWPwWLNyAt0UfY4hgsAYYrCGCxiBfQExUNC0SjUzoYTQmPYj2r+ciWazo/1vy76W+yNM/B2c3R3
uiEIKesagUrZnnh3s6lyZ/YfrFk5cVpi86XqYU4E4XqnoFLDYpsOp1xeTw5iR365holofk8dRD7VQP
2prLF+uEPGkmGsIS7AxNoeT1cR2W6u7Gek03Lp/TEBGxrKoUXEP5TmlvF3RNqd6N2UoyPbrr+8Z8Zi
e9613bfzzvvHs+/zz3vXvfu/f5/6UxdKPokbGMQDo1kh6yOYgWjXx2mE8M9fX/mNgtBtg/+50/Jg6P
...
JJhu2gMPKh7XE116h40jfv5pHrafOCxvxB0zbTXkyjk1VgoLsdVXGd1HDARd6sVWbcrBqLP3M2bDP9
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

4. Once ProphetStor received the activation request email and validated the “registrationData”, you will receive the license activation code, “signatureData”, via an email. Copy the “signatureData” from the email to the “keycode.signatureData” field and save the change. Your license keycode is now activated.

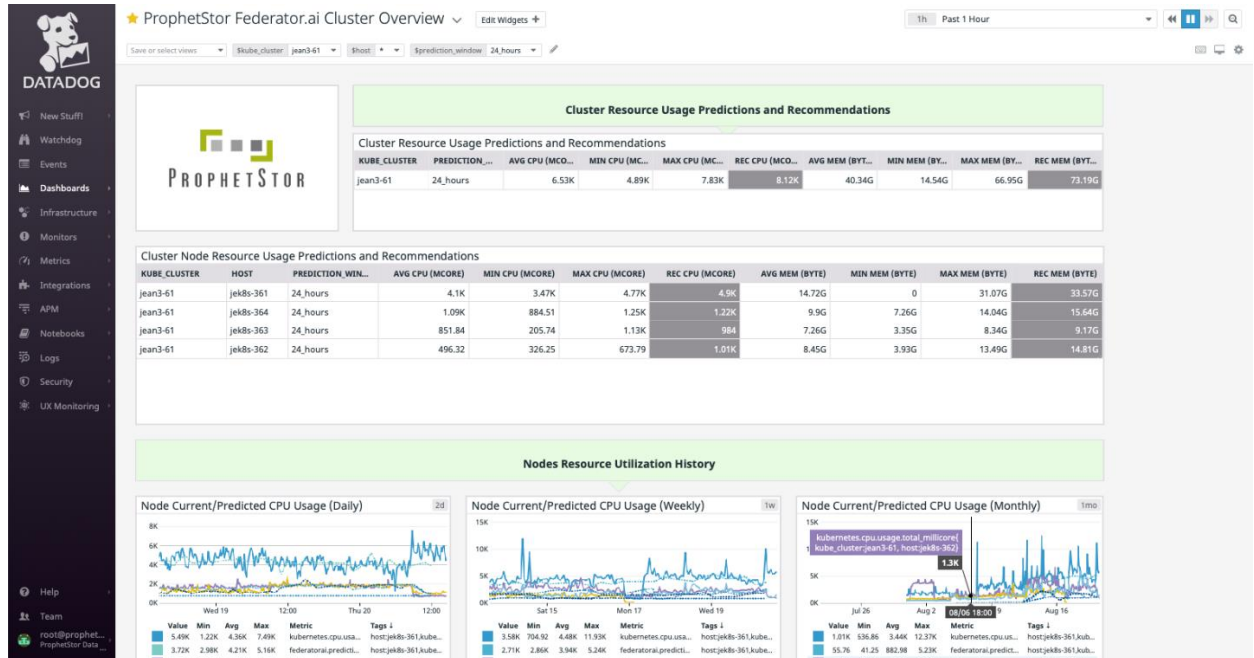
```
apiVersion: federatorai.containers.ai/v1alpha1
kind: AlamedaService
metadata:
  name: my-alamedaservice
.....
spec:
  .....
  keycode:
    codeNumber: K4AM0C4TSDXXXXXXXXXXXXXXXXXXXXXQ
    signatureData: F5nmus478ertgnldd430gvsef90gNYAt0UfY4hgsAYYrCGCxibfQExUNC0S
KkMhZSiKSAYgG pi86XqYU4E4a3oAiKUPYoSB0E0YooKquADCGmgkiQCigYrCGCxergHwernREBo4E
wegjWPwWLNyAt0UfY4hgsAYYrCGCx6UxdKpOkG0SjUZoYTQmPYj2r+ciWazo/1vy76W+yNM/B2c3R3
OYgWjXx2mE8M9fh3s6lyZ/YfrFk5cVpixdKpOkbGMQDo1khRTNB0p1xeTw5iR365holofk8dRD7VQP
2prLF+uEPGkmGsIS7AxNoeT1cR2W6u7Gek03Lp/TEBGxrKoUXEP5TmlvF3RNqd6N2UoyPbrr+8Z8Zi
e9613bfzzvvHs+/2Z1ZGFpLXJ1ZA03ad1DTZxjA8R9IoxdKpOkbGMQDo1kh6yOYg8M9fX/RwtBVerh
...
YBoerBTR445h4536g456UJdfsheryhryu6JwerJwerYjJker5zQ6kZrFFhkr6sVwbcRbQLPregUh9
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Appendix

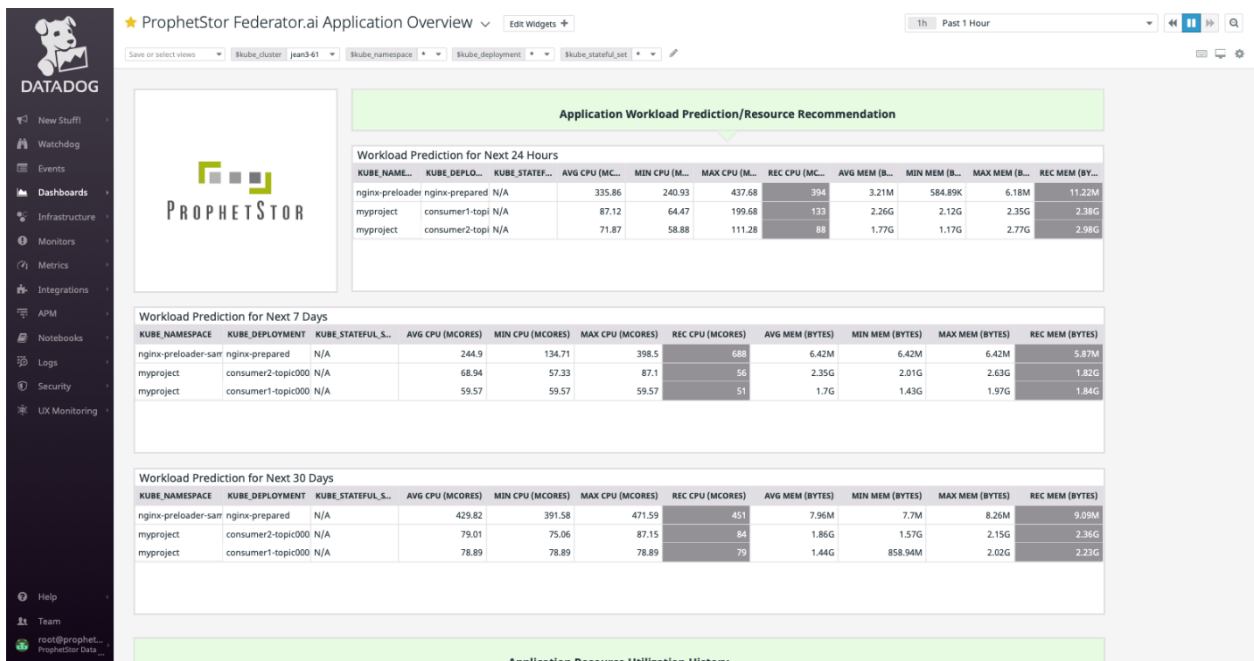
Datadog Dashboards

The following Custom Datadog Dashboards are available after Federator.ai is installed.

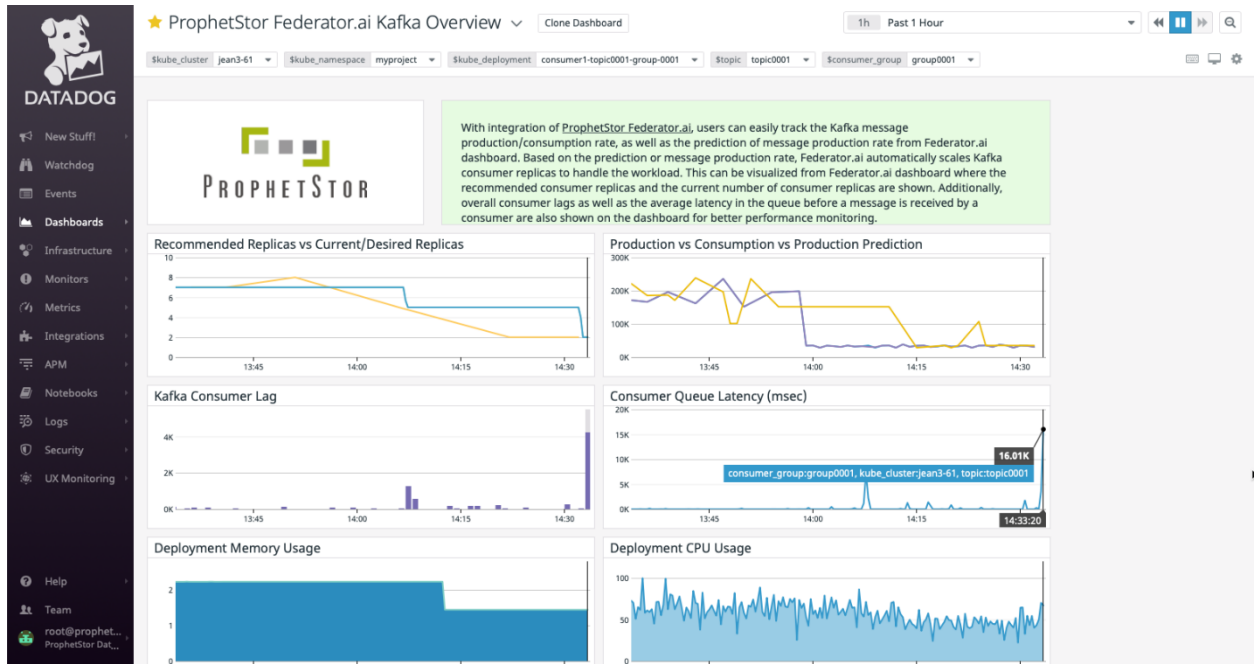
- ProphetStor Federator.ai Cluster Overview



- ProphetStor Federator.ai Application Overview



- ProphetStor Federator.ai Kafka Overview



- ProphetStor Federator.ai Cost Analysis Overview

