Federator.ai Release 4.3.1 for Datadog Installation Guide

Content

Overview

Federator.ai

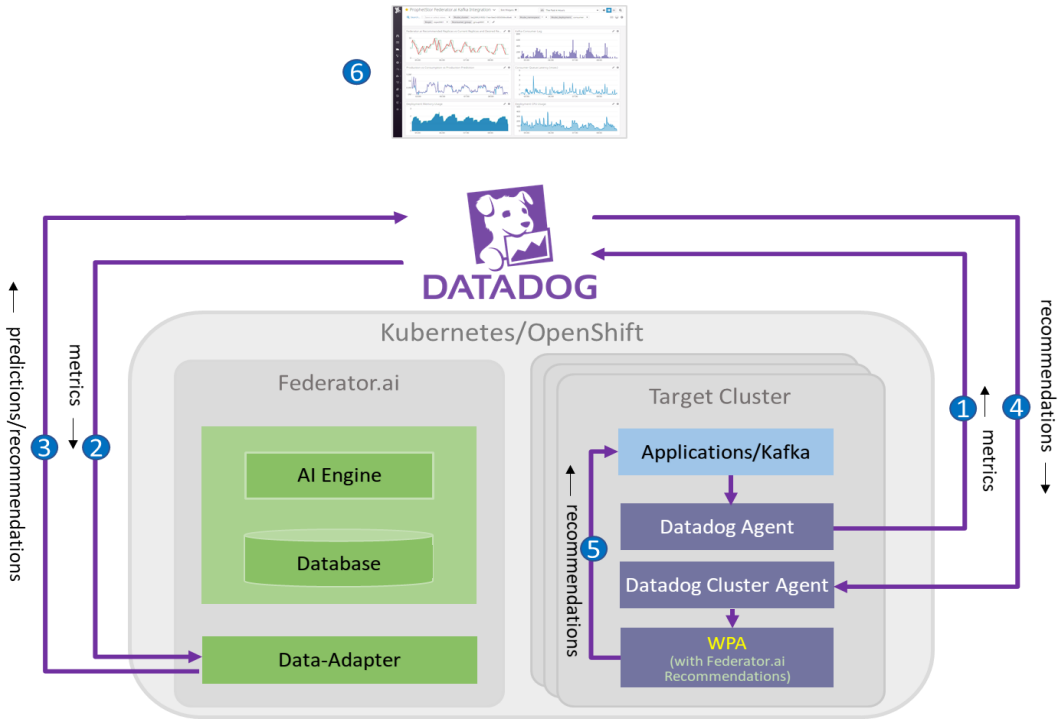
ProphetStor Federator.ai is an AI-based solution that helps enterprise manage, optimize, auto-scale resources for any applications on Kubernetes. Using advanced machine learning algorithms to predict application workload, Federator.ai scales the right amount of resources at the right time for optimized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

- AI-based workload prediction for Kafka or other applications
- Resource recommendation based on workload prediction, application, Kubernetes 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 fowards 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 : 3.11/4.3/4.4/4.5
- Kubernetes : 1.11 ~ 1.18.x

Federator.ai Resource Requirements

- Total Resource Requirments
  - 4 CPU cores
  - 4 GB Memory
  - StorageClass: 430GB (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 *4.3.1*
- 14 days trial license

Datadog Agent Version(reference)

- Datadog Agent helm chart version : v2.4.24
- Datadog Agent version : v7.22.0
- Datadog Cluster Agent version : v1.8.0
- 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 optimaztion 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

Kubernetes:

#	Checklist Item	Requirement	Details

1	What is the Kubernetes version?	1.11~1.18.x	Use the command below to get Kubernetes version:
			
			`$ kubectl version`
			`...`
			`Server Version: version.Info{Major:"1", Minor:"17", GitVersion:"v1.17.2", GitCommit:"59603c6e503c87169aea6106f57b9f242f64df89", GitTreeState:"clean", BuildDate:"2020-01-18T23:22:30Z", GoVersion:"go1.13.5", Compiler:"gc", Platform:"linux/amd64"}`

2	Does installation on Kubernetes cluster require private image repository?	If private image repository is required, the following information is needed during installation - Private image repository URL - Credential of the private image repository	Input the URL and credential when Federator.ai installation script asks for the information.
3	StorageClass and Persistent Volumes requirement	StorageClass supports ReadWriteMany access mode. Available storage size is larger than 430GB.	Minimum storage size for Federator.ai Release 4.3.1 is 430GB, including database, data, and logs.
4	Kubernetes cluster CPU/memory requirement	Minimum CPU/mem/storage: - CPU: 4 Cores - Memory: 4 GB - Storage Class Capacity: 430GB At least one worker node with - 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.

5	Is Kubernetes cluster allowed for NodePort configuration?	Federator.ai creates two NodePorts for GUI and REST API by default - REST API - https://<server>:31011 - GUI - https://<server>:31012	If NodePort is not allowed, answer 'N' when installation script prompts for creating NodePorts. Users need to expose Federator.ai GUI and REST API service manually.
6	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 - 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 - 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 $ **kubectl get resourcequota --all-namespaces**
7	Does deployment must have resource request/limit specified?	By default, Federator.ai deployments do not specify resource requests/limits. It can be done by setting up an environment variable before installation starts.	To turn on resource request/limit settings for all Federator.ai deployments, manually export environment variable before running 'federatorai-launcher.sh' $ **export ENABLE_RESOURCE_REQUIREMENT=y** $ ./federatorai-launcher.sh

Datadog Agent:

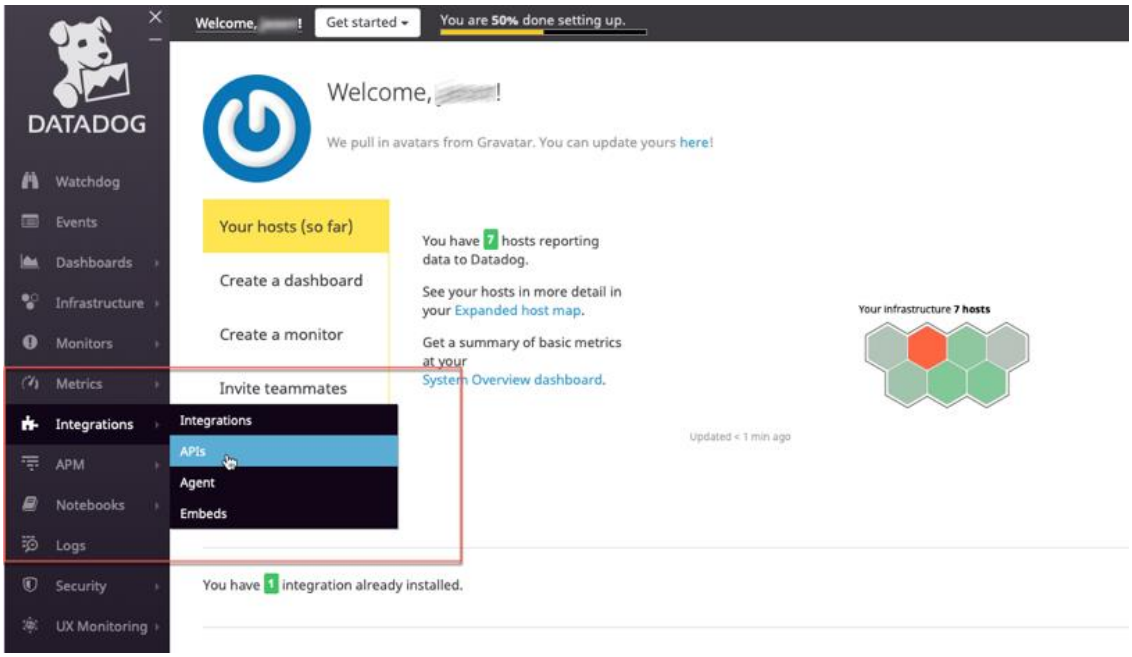
#	Checklist Item	Requirement	Details
1	Is Datadog Agent installed?	Datadog Agent is mandatory	Kubernetes resources and workload metrics are collected by Datadog Agent.
2	Is Datadog Cluster Agent installed?	Cluster Agent is mandatory	Cluster Agent provides metrics to HPA Autoscaler for autoscaling.
3	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.
4	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 enables $ **kubectl exec *<datadog-agent-pod>* -n *<datadog-agent-namespace>* -- agent integration show datadog-kafka-consumer** Refer to https://www.datadoghq.com/blog/monitor-kafka-with-datadog/ for Kafka Consumer integration installation
5	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.
6	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.
7	"DD_CLUSTER_NAME" is configured for Datadog Agent?	Cluster Agent uploads 'kube_cluster_name' as the tag and the value of DD_CLUSTER_NAME as the value to Datadog Service	Use the command to confirm DD_CLUSTER_NAME is configured *$* **kubectl exec -it *<datadog-cluster-agent-pod>* -n *<datadog-agent-namespace>* -- env | grep DD_CLUSTER_NAME**

Before You Start

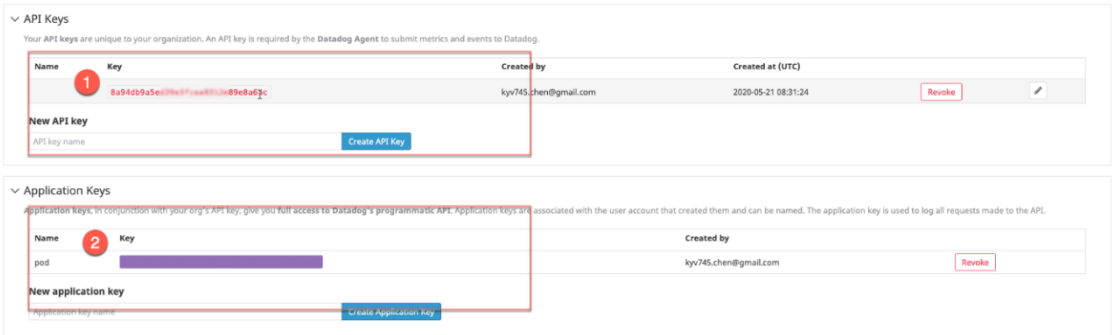
- The admin role for installing Fedeator.ai is “Cluster Admin”.
- Datadog agent must be ready if Federator.ai runs in the same Kubernetes cluster that is being monitored.
- Obtain Datadog account APIKey, APPKey.
  1. A Datadog account is required for connecting and using Datadog Cloud Service. If you 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



Installation

1. Log into Kubernetes cluster

2. Install the Federator.ai for Kubernetes by using the following command

```
$ curl https://raw.githubusercontent.com/containers-ai/federatorai-
operator/master/deploy/federatorai-launcher.sh |bash
```

```
~# 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 13505  100 13505    0     0  19617      0 --:--:-- --:--:-- --:--:-- 19600
Please input Federator.ai version tag (e.g., v4.2.755):  v4.3.datadog-patch1

Downloading scripts ...

Downloading Federator.ai CR yamls ...

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

Executing install.sh ...
Checking environment version...
...Passed
Enter the namespace you want to install Federator.ai [default: federatorai]:

----------------------------------------
tag_number = v4.3.datadog-patch1
install_namespace = federatorai
----------------------------------------
Is the above information correct? [default: y]:
Downloading file 00-namespace.yaml ...
Done
Downloading file 01-serviceaccount.yaml ...
Done
Downloading file 02-alamedaservice.crd.yaml ...
Done
Downloading file 03-federatorai-operator.deployment.yaml ...
Done
Downloading file 04-clusterrole.yaml ...
Done
Downloading file 05-clusterrolebinding.yaml ...
Done
Downloading file 06-role.yaml ...
Done
Downloading file 07-rolebinding.yaml ...
Done

Applying Federator.ai operator yaml files...
Applying 00-namespace.yaml...
namespace/federatorai created
Applying 01-serviceaccount.yaml...
serviceaccount/federatorai-operator created
Applying 02-alamedaservice.crd.yaml...
customresourcedefinition.apiextensions.k8s.io/alamedaservices.federatorai.containers.ai
created
Applying 03-federatorai-operator.deployment.yaml...
deployment.apps/federatorai-operator created
Applying 04-clusterrole.yaml...
clusterrole.rbac.authorization.k8s.io/federatorai-operator created
clusterrole.rbac.authorization.k8s.io/alameda-gc created
```

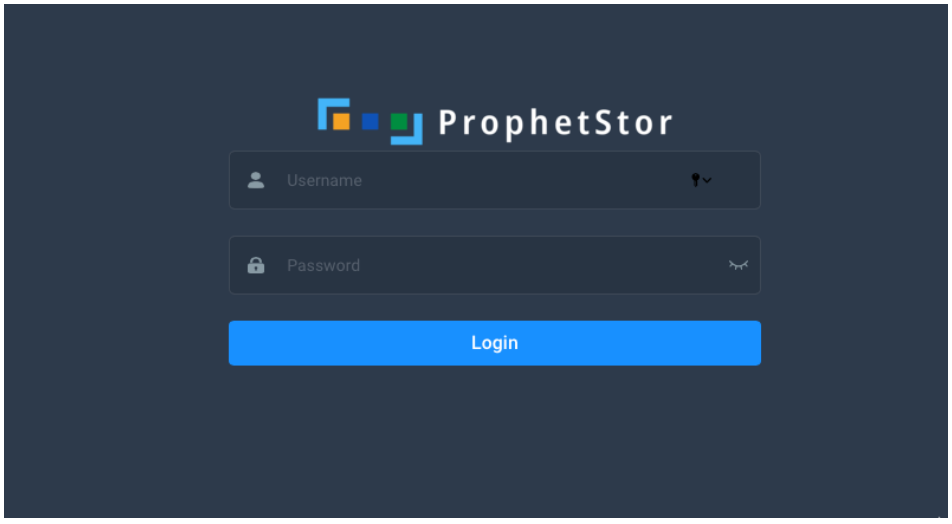
```
Applying 05-clusterrolebinding.yaml...
clusterrolebinding.rbac.authorization.k8s.io/federatorai-operator created
Applying 06-role.yaml...
role.rbac.authorization.k8s.io/federatorai-operator created
Applying 07-rolebinding.yaml...
rolebinding.rbac.authorization.k8s.io/federatorai-operator created

Checking pods...
All federatorai pods are ready.

Install Federator.ai operator datadog successfully

Downloading Federator.ai CR sample files ...
Done
========================================
Which storage type you would like to use? ephemeral or persistent?
[default: ephemeral]: persistent
Specify log storage size [e.g., 10 for 10GB, default: 10]: 10
Specify data storage size [e.g., 10 for 10GB, default: 10]: 10
Specify InfluxDB storage size [e.g., 100 for 100GB, default: 100]: 100
Specify storage class name: managed-nfs-storage
Do you want to expose dashboard and REST API services for external access? [default: y]:

----------------------------------------
install_namespace = federatorai
storage_type = persistent
log storage size = 10 GB
data storage size = 10 GB
InfluxDB storage size = 100 GB
storage class name = managed-nfs-storage
expose service = y
----------------------------------------
Is the above information correct [default: y]:
Processing...
Waiting for datahub(datadog) pod to be ready ...
datahub pod is running.

Checking pods...
Waiting for pods in namespace federatorai to be ready...
Waiting pod alameda-ai-789db8bcfb-mlh7h in namespace federatorai to be ready. phase:
[Running]
Waiting for pods in namespace federatorai to be ready...
Waiting pod alameda-notifier-7767dc597d-bjzvc in namespace federatorai to be ready. phase:
[Running]
Waiting for pods in namespace federatorai to be ready...

All federatorai pods are ready.
Please input Datadog API key: 327a6f7072f93883797270c2ae962xxx
Please input Datadog Application key: bd6b50ff1c108b9e914b12ae4cafea1cbb2aexxx
secret/federatorai-data-adapter-secret patched
secret/federatorai-data-adapter-secret patched
pod "federatorai-data-adapter-68b8dfb9-vld25" deleted

Checking pods...
All federatorai pods are ready.

========================================
You can now access GUI through https://<YOUR IP>:31012
Default login credential is admin/admin

Also, you can start to apply alamedascaler CR for the target you would like to monitor.
Review administration guide for further details.
========================================

========================================
You can now access Federatorai REST API through https://<YOUR IP>:31011
Default login credential is admin/admin
```

```
The REST API online document can be found in https://<YOUR
IP>:31011/apis/v1/swagger/index.html
========================================

Install Federator.ai datadog successfully
Do you want to monitor this cluster? [default: y]:
Use "k8s-4-205" as cluster name and DD_CLUSTER_NAME  ← got it from datadog agent
configuration by script automatically, if it is existing.
Applying file alamedascaler_federatorai.yaml ...
alamedascaler.autoscaling.containers.ai/clusterscaler created
Done

Downloaded YAML files are located under /tmp/install-op

Downloaded script files are located under /tmp/federatorai-scripts/v4.3.datadog-patch1
```

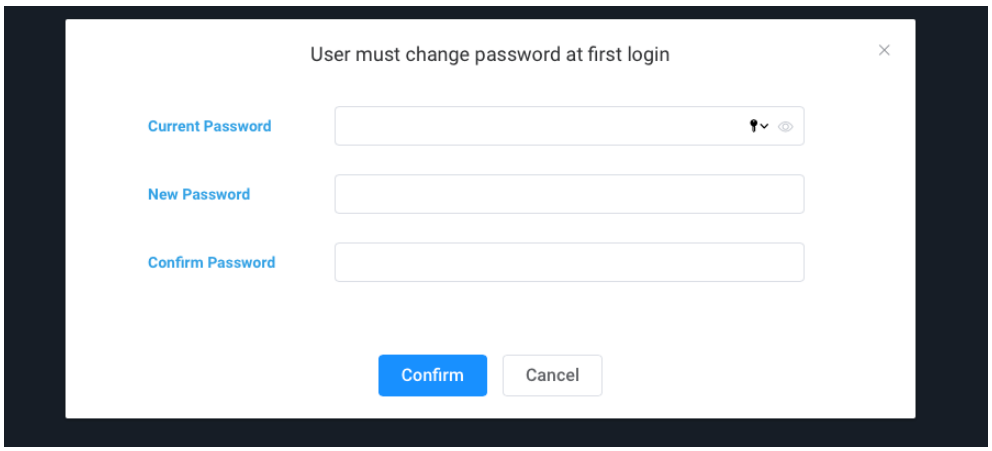
3. Verify Federator.ai pods are running properly

```
~# kubectl get pod -n federatorai
NAME                                          READY   STATUS    RESTARTS   AGE
alameda-ai-789db8bcfb-mlh7h                   1/1     Running   0          31m
alameda-ai-dispatcher-768f869754-4tw27        1/1     Running   1          31m
alameda-analyzer-bc9dffb8f-xkqvq              1/1     Running   0          31m
alameda-datahub-74d8f99865-57xs9              1/1     Running   0          31m
alameda-influxdb-0                            1/1     Running   0          31m
alameda-notifier-7767dc597d-bjzvc             1/1     Running   0          31m
alameda-operator-6fbc794dcb-shd8v             1/1     Running   0          31m
alameda-rabbitmq-666896899d-cdsw6             1/1     Running   0          31m
alameda-recommender-5797d7bc46-8sgrp          1/1     Running   0          31m
fedemeter-api-8459b778bc-6vt5g                1/1     Running   0          31m
fedemeter-influxdb-0                          1/1     Running   0          31m
federatorai-agent-b499cdc44-2c5fv             1/1     Running   0          31m
federatorai-dashboard-backend-6c6db96444-kd24l  1/1     Running   0          31m
federatorai-dashboard-frontend-5df45f7cb6-vc57x  1/1     Running   0          31m
federatorai-data-adapter-68b8dfb9-jvjxg       1/1     Running   0          24m
federatorai-operator-5fd47b6b7f-m92vf         1/1     Running   0          35m
federatorai-rest-748984b79d-fgx4w             1/1     Running   0          31m
```

4. Log on Federator.ai GUI
   Federator.ai GUI URL can be found in the output of Step 2. The default Username and Password are “admin/admin”



Change the default password when you log in Federator.ai for the first time


User must change password at first login
Current Password
New Password
Confirm Password
Confirm
Cancel

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")

```
$ kubectl 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>        9404/TCP
15d
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., "topic0001")

```
$ kubectl 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

$ kubectl -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")

```
$ kubectl 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
...

$ kubectl -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 Data Adapter configuration helper script in "/tmp/federatorai-scripts/v4.3.datadog-patch1/scripts"

```
~# ls -l /tmp/federatorai-scripts/v4.3.datadog-patch1/scripts
-rw-r--r--. 1 root root 18088 Oct 20 00:08 cluster-property-setup.sh
-rw-r--r--. 1 root root 12000 Oct 20 00:08 email-notifier-setup.sh
-rw-r--r--. 1 root root 13505 Oct 20 00:08 federatorai-launcher.sh
-rw-r--r--. 1 root root 42074 Oct 20 00:08 federatorai-setup-for-datadog.sh
-rw-r--r--. 1 root root 45564 Oct 20 00:08 install.sh
-rw-r--r--. 1 root root  4950 Oct 20 00:08 node-label-assignor.sh
-rw-r--r--. 1 root root 34888 Oct 20 00:08 planning-util.sh
-rw-r--r--. 1 root root 49468 Oct 20 00:08 preloader-util.sh
-rw-r--r--. 1 root root  7654 Oct 20 00:08 prepare-private-repository.sh
-rw-r--r--. 1 root root  6255 Oct 20 00:08 uninstall.sh
```

- Change file permission to be executable

```
$ chomd +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-patch1/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  ← see below #note-1 for more details

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: kubectl apply should be used on resource created by either kubectl create --save-
config or kubectl 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
```

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

- Verify configuration result

```
~# ls -l config-result/
-rw-r--r-- 1 root root 35666 Sep 16 12:06 adapter-configmap.yaml
-rw-r--r-- 1 root root   912 Sep 16 12:06 kafka-consumer.yaml
-rw-r--r-- 1 root root   690 Sep 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
~# kubectl 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_wa
termarkpodautoscalers_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 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 #note-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 #note-1 for more details
          kube_deployment: nginx-sample
          kube_namespace: nginx-sample
```

#note-1: *"kube_cluster" value must match with DD_TAGS (value="kube_cluster:**<cluster_name>**"), or if you have only DD_CLUSTER_NAME ("clusterName" field in datadog-values.yaml) configured in Datadog Agent (datadog-values.yaml), not DD_TAGS, then this "kube_cluster: <cluster_name>" field should be replaced with "**kube_cluster_name: <cluster_name>**"*

- Deploy WPA and confirm the status

```
$ kubectl apply -f wpa.yaml
```

4. (Optional) Federator.ai, by default uploads Cost Analysis metrics to Datadog Service to show the information on Datadog Cost Analysis Overview dashboard. You can optionally disable this feature by configuring the default 'AlamedaOrganization' CR, or use the helper script *'/tmp/federatorai-scripts/v4.3.datadog-patch1/scripts/cluster-property-setup.sh'* to disable it.

- Example of 'AlamedaOrganization' CR

```
~# kubectl edit alamedaOrganization -n federatorai
apiVersion: tenant.containers.ai/v1alpha1
kind: AlamedaOrganization
metadata:
  name: default
spec:
  tenant: default
  features:
  - type: costAnalysis
    costAnalysis:
      enabled: true
      mode: localOnly  # replace "uploadResult" with "localOnly"
  watchedNamespace:
    operator: exclude
    names:
... ...
```

- Example of using the helper script ‘cluster-property-setup.sh’

```
~# bash cluster-property-setup.sh
Alameda Organization:
       (a) Display current settings.
       (b) Add/Edit individual cluster settings.
       (c) Remove individual cluster settings.
       (d) Add/Edit global cluster settings.
       (e) Remove global cluster settings.
       (f) Exit.
Please enter your choice: d

Do you want to upload cost analysis metrics to monitoring cloud service (Datadog)?
[default: y]: n
Do you want to configure watched namespaces of this cluster? [default: y]:
Input watched namespace operator [include/exclude]: exclude
Do you want to exclude system namespaces? [default: y]: y
alamedaorganization.tenant.containers.ai/default patched
Done. Press ENTER to continue.

Alameda Organization:
       (a) Display current settings.
       (b) Add/Edit individual cluster settings.
       (c) Remove individual cluster settings.
       (d) Add/Edit global cluster settings.
       (e) Remove global cluster settings.
       (f) Exit.
Please enter your choice: a

======== Global Settings ================================================
cost analysis enabled: true
cost analysis mode: localOnly
watched namespaces operator: exclude
watched namespaces: kube-public,kube-service-catalog,kube-system,management-infra,kube-
node-lease,stackpoint-system,marketplace,openshift,openshift-*
=========================================================================
```

Manage Federator.ai License Keycode

Federator.ai uses a keycode to control the license. A 30-day trial keycode is installed by default. It requires replacing with a valid keycode from ProphetStor 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.

Apply Keycode

1. Get “AlamedaService” CR name

```
~# kubectl get alamedaservice --all-namespaces
NAMESPACE         NAME                              EXECUTION     VERSION       PROMETHEUS
AGE
federatorai   my-alamedaservice   false      v4.3.1031   https://prometheus-k8s.openshift-
monitoring:9091   45d
```

2. Edit the “AlamedaService” CR

```
$ kubectl 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: K4AMOC4TSDXXXXXXXXXXXXXXXXXXX
```

Delete Keycode

1. Get “AlamedaService” CR name

```
~# kubectl get alamedaservice --all-namespaces
NAMESPACE         NAME                              EXECUTION     VERSION       PROMETHEUS
AGE
federatorai   my-alamedaservice   false      v4.3.1031   https://prometheus-k8s.openshift-
monitoring:9091   45d
```

2. Edit the “AlamedaService” CR

```
~# kubectl 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-alamedaservice
......
spec:
  ......
  keycode:
    codeNumber:
```

Activate Keycode

1. Get “AlamedaService” CR name

```
~# kubectl get alamedaservice --all-namespaces
NAMESPACE         NAME                              EXECUTION      VERSION       PROMETHEUS
AGE
federatorai   my-alamedaservice   false      v4.3.1031   https://prometheus-k8s.openshift-
monitoring:9091   45d
```

2. Edit the “AlamedaService” CR

```
~# kubectl edit alamedaservice my-alamedaservice
```

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

```
apiVersion: federatorai.containers.ai/v1alpha1
kind: AlamedaService
metadata:
  name: my-alamedaservice
......
status:
  ......
  keycodeStatus:
    codeNumber: K4AMOC4TSDXXXXXXXXXXXXXXXXXXQ
    lastErrorMessage: ""
    registrationData: H4sICAavJl8C/2ZlZGFpLXJlZ2RhdGEudGd6AO3ad1DTZxjA8R9IoMoe
KkMhZSiKSAYgG0IwjBBkb4kRAgZJwwa3oAiKUPYoSB0E0YooKquADCGmgkiQCigqVoYgURCFiEJtr3
wegjWPwWLNYAt0UfY4hgsAYYrCGCxiBfQExUNC0SjUZoYTQmPYj2r+ciWazo/1vy76W+yNM/B2c3R3
uiEIKEsagUrZnnh3s6lyZ/YfrFk5cVpi86XqYU4E4XqnoFLDYPsOp1xeTw5iR365holofk8dRD7VQP
2prLF+uEPGkmGsIS7AxNoeT1cR2W6u7GekO3Lp/TEBGxrKoUXEP5TmlvF3RNqd6N2UoyPbrr+8Z8Zi
e9613bfzzvvHs+/zz3vXvfu/f5/6UxdKPokbGMQDo1kh6yOYgWjXx2mE8M9fX/mNgtBtg/+5O/Jg6P
...
JJhu2gMPKh7XEl16h4Ojfv5pHrafOCXvxB0zbTXkyjklVgoLsdVXGd1HDARd6sVWbcrBqLP3M2bDP9
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

4. Once ProphetStor received the activation request email and validated the “registrationData”, it returns the activation code, “signatureData”, via an email. Copy the “signatureData” from the email, fill in the “keycode. signatureData” field and save the change.

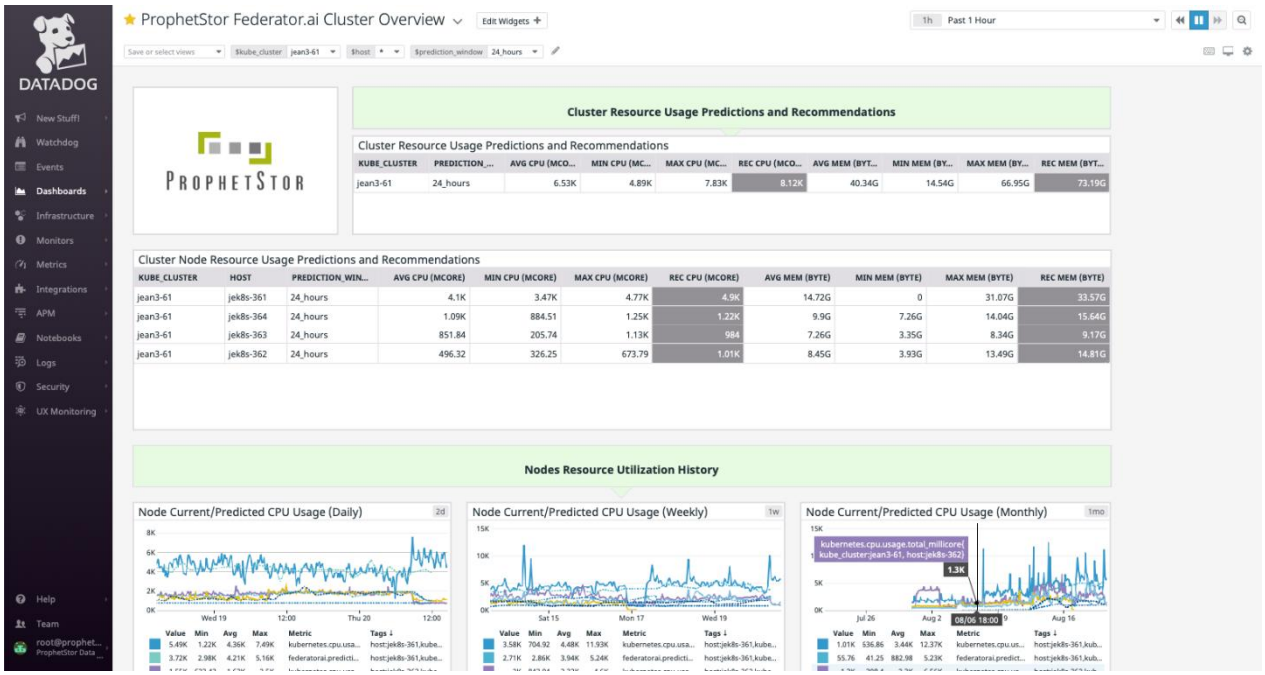
```
apiVersion: federatorai.containers.ai/v1alpha1
kind: AlamedaService
metadata:
  name: my-alamedaservice
......
spec:
  ......
  keycode:
    codeNumber: K4AMOC4TSDXXXXXXXXXXXXXXXXXXQ
    signatureData: F5nmus478ertgnldd430gvsef90gNYAt0UfY4hgsAYYrCGCxiBfQExUNC0S
KkMhZSiKSAYgG pi86XqYU4E4a3oAiKUPYoSB0E0YooKquADCGmgkiQCigYrCGCxergHwernREBo4E
wegjWPwWLNYAt0UfY4hgsAYYrCGCx6UxdKPokG0SjUZoYTQmPYj2r+ciWazo/1vy76W+yNM/B2c3R3
OYgWjXx2mE8M9fh3s6lyZ/YfrFk5cVpixdKPokbGMQDo1khRTNBOp1xeTw5iR365holofk8dRD7VQP
2prLF+uEPGkmGsIS7AxNoeT1cR2W6u7GekO3Lp/TEBGxrKoUXEP5TmlvF3RNqd6N2UoyPbrr+8Z8Zi
e9613bfzzvvHs+/2ZlZGFpLXJlZAO3ad1DTZxjA8R9IoxdKPokbGMQDo1kh6yOYg8M9fX/RwtBVerh
...
JBoerBTR445h4536g456UJdfsheryhryu6JwerJwerYjJKER5zQ6kZrFFhkr6sVWbcrBqLPregUh9
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Appendix

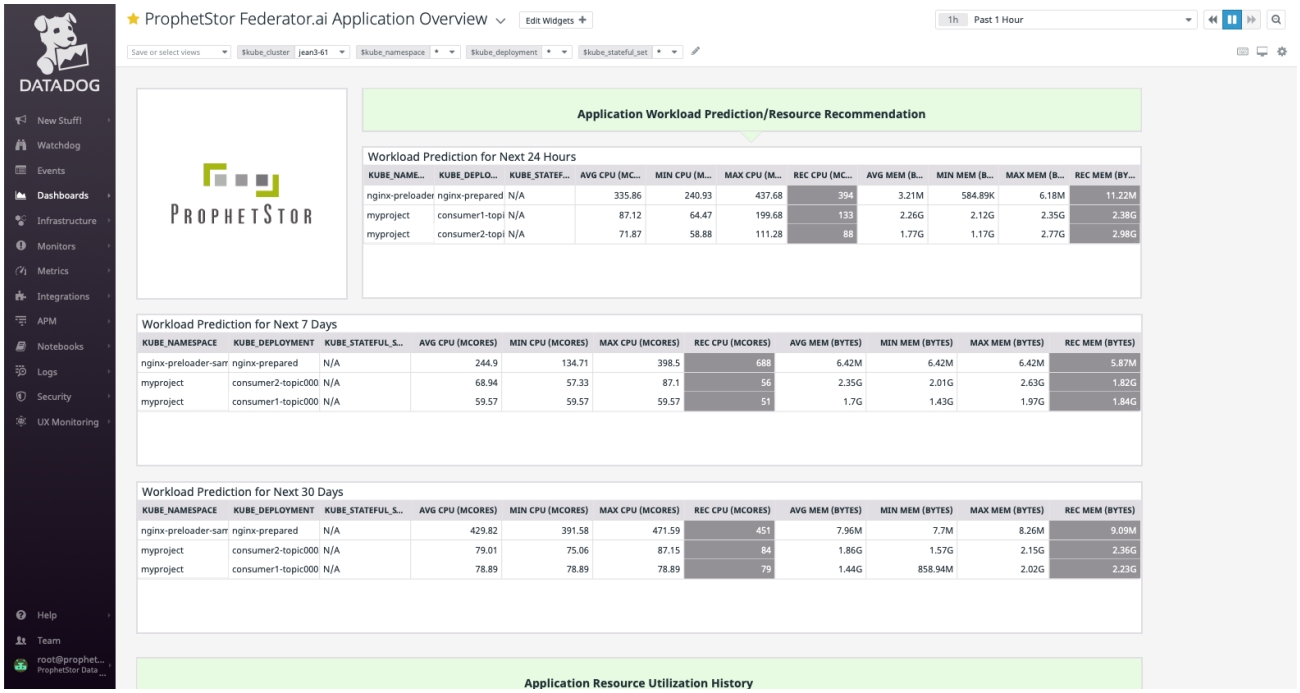
Datadog Dashboards

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

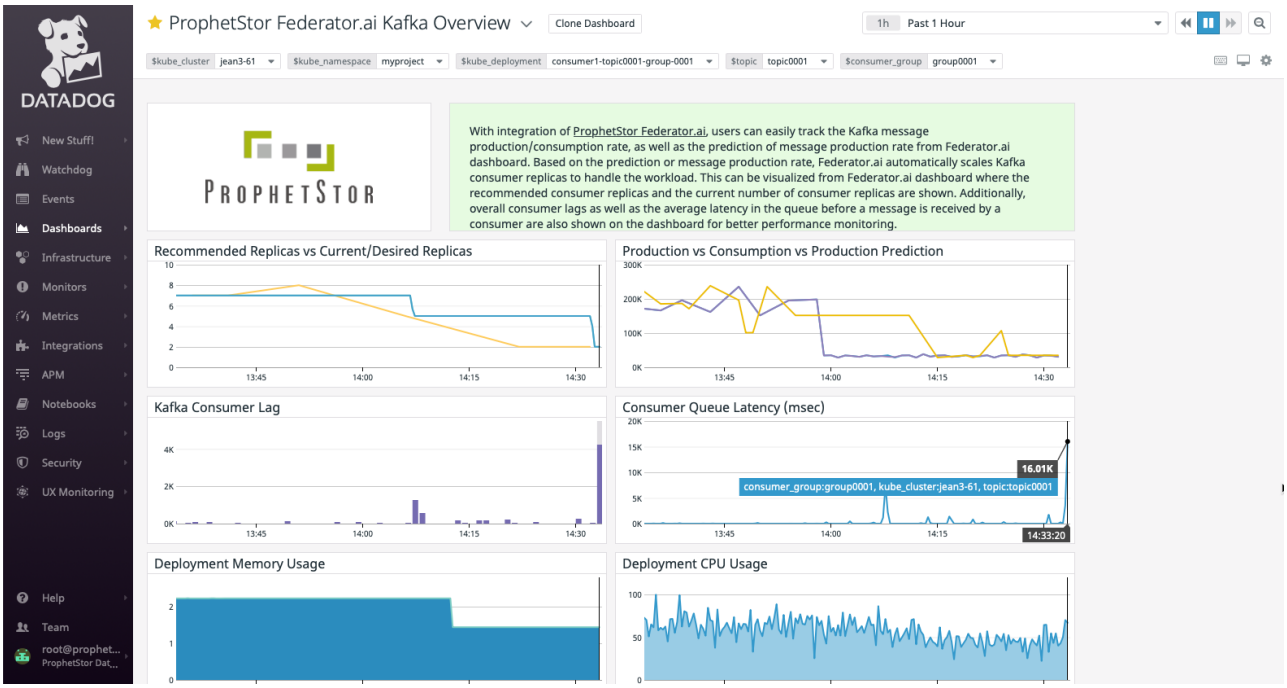
ProphetStor Federator.ai Cluster Overview



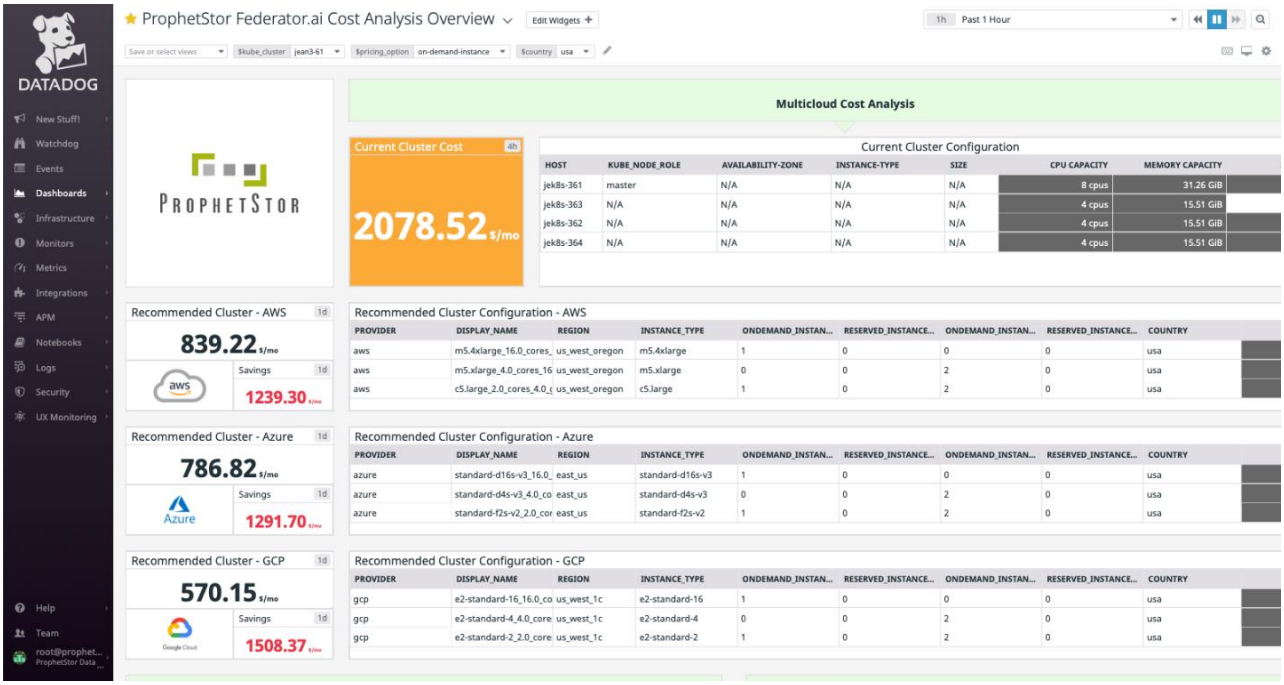
ProphetStor Federator.ai Application Overview



ProphetStor Federator.ai Kafka Overview



ProphetStor Federator.ai Cost Analysis Overview



Cluster Name Configuration for Datadog Agent

Use the commands to confirm Datadog Agent/Cluster Agent installation and cluster name configuration status

```
$ kubectl get pods -n <datadog_agent_namespace>
$ kubectl exec -it <datadog_cluster_agent_pod> -n <datadog_agent_namespace> -- env | grep
DD_CLUSTER_NAME
```

```
~# kubectl get pods -n default
NAME                                                READY   STATUS    RESTARTS   AGE
datadog-monitoring-6j2k4                            2/2     Running   2          26d
datadog-monitoring-8frn7                            2/2     Running   2          26d
datadog-monitoring-cluster-agent-5cdcbb8747-b2br2   1/1     Running   1          26d
datadog-monitoring-kube-state-metrics-769f55fd64-tptz7   1/1     Running   1          26d
datadog-monitoring-mlzdm                            2/2     Running   2          26d
wpacontroller-watermarkpodautoscaler-68484f8dd4-cqx65   1/1     Running   1          68d
~#
~# kubectl exec -it datadog-monitoring-cluster-agent-5cdcbb8747-b2br2 -n default -- env | grep
DD_CLUSTER_NAME
DD_CLUSTER_NAME=jean3-61
```

- If Datadog Agent is not installed, or Datadog Agent is installed by Helm Chart but Cluster Agent is not installed or cluster name is not configured

  1. Configure 'values.yaml' to enable Cluster Agent and cluster name

```
datadog:
... ...
  ## @param clusterName - string - optional
  ## Set a unique cluster name to allow scoping hosts and Cluster Checks easily
  ## The name must be unique and must be dot-separated tokens where a token can be up to
40 characters with the following restrictions:
  ## * Lowercase letters, numbers, and hyphens only.
  ## * Must start with a letter.
  ## * Must end with a number or a letter.
  ## Compared to the rules of GKE, dots are allowed whereas they are not allowed on GKE:
  ## https://cloud.google.com/kubernetes-
engine/docs/reference/rest/v1beta1/projects.locations.clusters#Cluster.FIELDS.name
  #
  clusterName: my-cluster  # <CLUSTER_NAME>
... ...
## @param clusterAgent - object - required
## This is the Datadog Cluster Agent implementation that handles cluster-wide
## metrics more cleanly, separates concerns for better rbac, and implements
## the external metrics API so you can autoscale HPAs based on datadog metrics
## ref: https://docs.datadoghq.com/agent/kubernetes/cluster/
#
clusterAgent:
  ## @param enabled - boolean - required
  ## Set this to true to enable Datadog Cluster Agent
  #
  enabled: true
```

  2. Use command '*helm install -f values.yaml*' to install a new Datadog Agent, or use command '*helm upgrade -f values.yaml*' to install Cluster Agent and configure cluster name

```
$ helm install -f values.yaml
$ helm upgrade -f values.yaml
```

- If running Datadog Agent is not installed by Helm Chart

  1. Configure cluster name by editing Cluster Agent's deployment YAML

```
~# kubectl edit deployment datadog-monitoring-cluster-agent -n default
apiVersion: apps/v1
kind: Deployment
metadata:
  ... ...
  name: datadog-monitoring-cluster-agent
  namespace: default
spec:
  ... ...
  template:
    ... ...
    spec:
      containers:
      - env:
        - name: DD_HEALTH_PORT
          value: "5555“
        ... ...
        - name: DD_CLUSTER_NAME
          value: <cluster_name>
```

Troubleshooting

1. WPA dumps errors during autoscaling

- Error message in WPA Controller

```
~# kubectl get pod -n default
NAME                                                READY   STATUS    RESTARTS   AGE
datadog-agent-2m6kk                                 1/1     Running   2          2d
datadog-agent-8kd54                                 1/1     Running   0          2d
datadog-agent-94rl6                                 1/1     Running   0          2d
datadog-agent-mq4mv                                 1/1     Running   0          2d
datadog-cluster-agent-74f44fdd4d-82tjp              1/1     Running   0          1d
docker-registry-1-vw59s                             1/1     Running   4        324d
prometheus-adapter-799b7dfc4f-rs7zj                 1/1     Running   1          6d
registry-console-2-jxfdl                            1/1     Running   2          6d
router-1-sw78l                                      1/1     Running   4        324d
wpacontroller-watermarkpodautoscaler-7ffbb97f9d-hcbsg 1/1   Running   0          1d

~# kubectl logs wpacontroller-watermarkpodautoscaler-7ffbb97f9d-hcbsg -n default
```

{"level":"info","ts":1589533961.5993037,"logger":"wpa_controller","msg":"Successful rescale","Request.Namespace":"myproject","Request.Name":"consumer1-topic0001-group-0001","currentReplicas":40,"desiredReplicas":40,"rescaleReason":""}

{"level":"error","ts":1589533961.600972,"logger":"wpa_controller","msg":"Error during reconcileWPA","Request.Namespace":"myproject","Request.Name":"consumer1-topic0001-group-0001","error":"the server could not find the requested resource (put watermarkpodautoscalers.datadoghq.com consumer1-topic0001-group-0001)","stacktrace":"github.com/go-logr/zapr.(*zapLogger).Error\n\twatermarkpodautoscaler/vendor/github.com/go-logr/zapr/zapr.go:128\ngithub.com/DataDog/watermarkpodautoscaler/pkg/controller/watermarkpodautoscaler.(*ReconcileWatermarkPodAutoscaler).Reconcile\n\twatermarkpodautoscaler/pkg/controller/watermarkpodautoscaler/watermarkpodautoscaler_controller.go:345\nsigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).reconcileHandler\n\twatermarkpodautoscaler/vendor/sigs.k8s.io/controller-runtime/pkg/internal/controller/controller.go:216\nsigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).processNextWorkItem\n\twatermarkpodautoscaler/vendor/sigs.k8s.io/controller-runtime/pkg/internal/controller/controller.go:192\nsigs.k8s.io/controller-runtime/pkg/internal/controller.(*Controller).worker\n\twatermarkpodautoscaler/vendor/sigs.k8s.io/controller-runtime/pkg/internal/controller/controller.go:171\nk8s.io/apimachinery/pkg/util/wait.JitterUntil.func1\n\twatermarkpodautoscaler/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:152\nk8s.io/apimachinery/pkg/util/wait.JitterUntil\n\twatermarkpodautoscaler/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait.Until\n\twatermarkpodautoscaler/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:88"}

- Reason
  - WPA is incompatible with Kubernetes 1.11
  - Install WPA on Kubernetes 1.11 dumps errors

```
must only have "properties", "required" or "description" at the root if the status
subresource is enabled
```

- Workaround
  - Comment out ‘subresources’ key in WatermarkPodAutoscaler CRD

```
~# cd
datadog_wpa/watermarkpodautoscaler_for_k8s_1.11/chart/watermarkpodautoscaler/templates
~# vi datadoghq.com_watermarkpodautoscalers_crd.yaml
...
...
    shortNames:
    - wpa
    singular: watermarkpodautoscaler
  scope: Namespaced
  #subresources:   ← comment out
  #  status: {}    ← comment out
  validation:
    openAPIV3Schema:
      description: WatermarkPodAutoscaler is the Schema for the watermarkpodautoscalers
        API
      properties:
        apiVersion:
          description: 'APIVersion defines the versioned schema of this representation
...
...
```

Note: It can auto-scale monitored application, but dump some errors during update status

- Related Datadog WPA ticket
  - https://github.com/DataDog/watermarkpodautoscaler/issues/50

2. Data Adapter reports errors

- Error messages in Data Adapter logs

```
~# oc exec -it $(oc get pods|grep federatorai-data-adapter|grep Running|awk '{print $1}') -- cat /var/log/telegraf.log
> telegraf.log
~# cat telegraf.log | grep "E\!"
2020-05-15T09:59:33Z E! [datadog][application_aware] Failed to get kafka consumer spec replicas
2020-05-15T09:59:33Z E! [inputs.datadog_application_aware] Error in plugin:
[url=https://api.datadoghq.com/api/v1/query][kafka]: Failed to get consumer information.
```

- Reason
  - Datadog Agent does not work with 'kube-state-metrics' comes with OpenShift

- Solution
  - Install another compatible 'kube-state-metrics'

```
If there is another kube-state-metrics running on openshift, rename all the clusterrole and clusterrolebinding name
of kube-state-metrics to prevent kube-state-metrics clusterrole name collision

restart datadog agent and make sure agent integrate with kube-state-metrics properly.
check all the node agent status by following command
~# oc exec <datadog-agent-pod-name> agent status
```