

# Federator.ai Release 4.3.1 for Datadog Installation Guide

# 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	6
Installation	7
Configuration	.11
Manage Federator.ai License Keycode	19
Apply Keycode	.19
Delete Keycode	.19
Activate Keycode	.20
Appendix	22
Datadog Dashboards	.22
ProphetStor Federator.ai Cluster Overview	.22
ProphetStor Federator.ai Application Overview	.22
ProphetStor Federator.ai Kafka Overview	.23
ProphetStor Federator.ai Cost Analysis Overview	.23
Cluster Name Configuration for Datadog Agent	.24
Troubleshooting	.26

# **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: <pre>\$ kubectl version Server Version: version.Info{Major:"1", Minor:"17", GitVersion:"v1.17.2", GitCommit:"59603c6e503c87169aea6106f57b9f242f6 4df89", GitTreeState:"clean", BuildDate:"2020- 01-18T23:22:30Z", GoVersion:"go1.13.5", Compiler:"gc", Platform:"linux/amd64"}</pre>
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 Al 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 Al 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</server></server>	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	<ul> <li>The CPU/memory required for Federator.ai depends on the number of clusters and applications being monitored/managed.</li> <li>Suggestion for initial namespace quota is <ul> <li>CPU 8 cores</li> <li>Memory 12G</li> </ul> </li> <li>The quota could be adjusted if number of managed clusters/applications increases.</li> <li>Use the command to get namespace resource quota \$ kubectl get resourcequotaall-namespaces</li> </ul>
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 \$ <b>kubectl exec</b> < <i>datadog-agent-pod&gt;</i> -n < <i>datadog- agent-namespace&gt;</i> agent integration show datadog- kafka-consumer Refer to <u>https://www.datadoghq.com/blog/monitor- kafka-with-datadog/</u> 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 < <i>datadog-cluster-agent-pod&gt;</i> -n < <i>datadog-agent-namespace&gt;</i> 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. <u>https://www.datadoghq.com/</u>
  - 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/

	×	Welcome, ! Get started	You are 50% done setting up.
D	ATADOG	We pull in a	avatars from Gravatar. You can update yours <b>here</b>
ñ	Watchdog		
	Events	Your hosts (so far)	Var have 2 brack constitue
-	Dashboards +	Courts a dashbarrad	data to Datadog.
•0	Infrastructure +	Create a dashboard	See your hosts in more detail in your Expanded host map. Your infrastructure <b>7 hosts</b>
0	Monitors +	Create a monitor	Get a summary of basic metrics at your
(7)	Metrics +	Invite teammates	System Overview dashboard.
<b>#</b> -	Integrations	Integrations	Updated < 1 min ago
	APM F	APIs	
8	Notebooks )	Agent	
10	Logs		
O	Security +	You have 🚺 integration alread	ly installed.
	UX Monitoring +		

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

API Keys Your API keys are unique to your organization. An API key is required by the Datadog Agent to submit metrics and events to Datadog.							
Name Key	Created by	Created at (UTC)					
8a94db9a5e 34 1 89e8a6ặc	kyv745.chen@gmail.com	2020-05-21 08:31:24	Revoke				
New API key							
API key name Create API Key	1						
✓ Application Keys							
Pplication keys in conjunction with your org's APF key, give your full access to bacadeg's programmadic APF. Application keys are associated with the user account that created them and can be named. The application key is used to log all requests made to the API.							
Name 2 Key		Created by					
		los 245 shee from all sam	Reucks				
pod		kyv745.cnen@gmail.com	THE VOICE				
pod New application key		kyvvio.cnenegmail.com	NEYUNE				

## 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
                                             Total
                                                     Spent
                               Dload Upload
                                                               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

#### 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"

<ul> <li>Username</li> <li>Password</li> <li>Login</li> </ul>
Password Login
Login

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

Current Password  New Password  Confirm Password  Confirm Cancel	Us	er must change password at first login	×
New Password       Confirm Password       Confirm       Cancel	Current Password	<b>f</b> ~ ©	
Confirm Password Confirm Cancel	New Password		
Confirm	Confirm Password		
		Confirm	

# 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")

<pre>\$ kubectl get svc -n myprojec</pre>	t			
my-cluster-kafka-bootstrap	ClusterIP	10.107.237.39	<none></none>	
9091/TCP,9092/TCP,9093/TCP,94	04/TCP 15d			
my-cluster-kafka-brokers	ClusterIP	None	<none></none>	
9091/TCP,9092/TCP,9093/TCP	15d			
my-cluster-kafka-exporter	ClusterIP	10.98.96.53	<none></none>	9404/TCP
15d				
<pre>my-cluster-zookeeper-client</pre>	ClusterIP	10.110.115.16	<none></none>	
9404/TCP,2181/TCP	15d			
my-cluster-zookeeper-nodes	ClusterIP	None	<none></none>	
2181/TCP,2888/TCP,3888/TCP	15d			

• Find topic ID of interest (e.g., "topic0001")

<pre>\$ kubectl get pod -n myproject</pre>					
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	
<pre>\$ kubectl -n myproject exec my-cluster-kafka-0 -c kafka bin/kafka-topics.sh bootstrap-server my-cluster-kafka-bootstrap:9092list 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</pre>					

• Find Consumer Group ID (e.g., "group0001")

<pre>\$ 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</pre>						
<pre>\$ kubectl -n myproject exec my-cluster-kafka-0 -c kafka bin/kafka-consumer-groups.sh bootstrap-server my-cluster-kafka-bootstrap:9092list 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</pre>						

- 2. Configure Federator.ai Data Adapter to connect to Datadog Service
  - Use Data Adapter configuration helper script in "/tmp/federatorai-scripts/v4.3.datadogpatch1/scripts"

```
~# ls -1 /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 45564 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"				
<pre>~# DD_NAMEWPA="wpacontroller"</pre>				
<pre>~# helm install \$DD_NAMEWPA -n \$DD_NAMESPACE ./chart/</pre>	watermai	rkpodautoscal	er	
~# kubectl get pods -n default				
NAME	READY	STATUS RES	TARTS	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-b5dn	q 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 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

#### **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: K4AMOC4TSDXXXXXXXXXXXXXXXXXXXXXXXXXX
   lastErrorMessage: ""
   registrationData: H4sICAavJ18C/2Z1ZGFpLXJ1Z2RhdGEudGd6AO3ad1DTZxjA8R9IoMoe
KkMhZSiKSAYgG0IwjBBkb4kRAgZJwwa3oAiKUPYoSB0E0YooKquADCGmgkiQCigqVoYgURCFiEJtr3
wegjWPwWLNYAt0UfY4hgsAYYrCGCxiBfQExUNC0SjUZoYTQmPYj2r+ciWazo/1vy76W+yNM/B2c3R3
uiEIKEsagUrZnnh3s61yZ/YfrFk5cVpi86XqYU4E4XqnoFLDYPs0p1xeTw5iR365holofk8dRD7VQP
2prLF+uEPGkmGsIS7AxNoeT1cR2W6u7Gek03Lp/TEBGxrKoUXEP5TmlvF3RNqd6N2UoyPbrr+8Z8Zi
e9613bfzzvvHs+/zz3vXvfu/f5/6UxdKPokbGMQDo1kh6yOYgWjXx2mE8M9fX/mNgtBtg/+50/Jg6P
JJhu2gMPKh7XE116h40jfv5pHraf0CXvxB0zbTXkyjklVgoLsdVXGd1HDARd6sVWbcrBqLP3M2bDP9
```

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:
             \texttt{signatureData: F5nmus478} ertgnldd 430 gvsef 90 gNYAt0 UfY4 hgsAYYrCGC x i BfQE x UNC0 State of the second state of the se
KkMhZSiKSAYgG pi86XqYU4E4a3oAiKUPYoSB0E0YooKquADCGmgkiQCigYrCGCxergHwernREBo4E
wegjWPwWLNYAt0UfY4hgsAYYrCGCx6UxdKPokG0SjUZoYTQmPYj2r+ciWazo/1vy76W+yNM/B2c3R3
OYgWjXx2mE8M9fh3s6lyZ/YfrFk5cVpixdKPokbGMQDo1khRTNBOp1xeTw5iR365holofk8dRD7VQP
2prLF+uEPGkmGsIS7AxNoeT1cR2W6u7GekO3Lp/TEBGxrKoUXEP5Tm1vF3RNqd6N2UoyPbrr+8Z8Zi
e9613bfzzvvHs+/2ZlZGFpLXJlZAO3ad1DTZxjA8R9IoxdKPokbGMQDo1kh6yOYg8M9fX/RwtBVerh
JBoerBTR445h4536g456UJdfsheryhryu6JwerJwerYjJKER5zQ6kZrFFhkr6sVWbcrBqLPregUh9
```

# Appendix

# **Datadog Dashboards**

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

# ProphetStor Federator.ai Cluster Overview

97	* ProphetSt	tor Feder	ator.ai Clus	ster Overview $  imes $	Edit Widgets +				3	h Past 1 Hour		- 4 <mark>11</mark> ⊮ Q
	Save or select views	<ul> <li>\$kube_clust</li> </ul>	er jean3-61 *	Shost * • Sprediction_wir	dow 24_hours *							∞ ♀ ♦
DATADOG												
📢 New Stuffi						Cluster Resourc	e Usage Predicti	ons and Reco	mmendations			
H Watchdog	- E			Cluster Resource Usa	ige Predictions and	d Recommendatio	ns					
Events	Door		0.0	KUBE_CLUSTER PREDICT	ION AVG CPU (MC	O MIN CPU (MC	MAX CPU (MC R	EC CPU (MCO A	VG MEM (BYT MIN MEI	и (вү MAX MEM (вү	. REC MEM (BYT	
🖮 Dashboards 🔸	LKOP	, H F I 9 I	UK	jean3-61 24_hour	s 6.	53K 4.89K	7.83K	8.12K	40.34G	14.54G 66.950	5 73.19G	
😵 Infrastructure 🔸												
Monitors	Cluster Mede	Decourse Us	ano Dradiction	c and Decommendatio								
(7) Metrics	KUBE_CLUSTER	HOST	PREDICTION_W	IN AVG CPU (MCORE)	MIN CPU (MCORE)	MAX CPU (MCORE)	REC CPU (MCORE)	AVG MEM (BY	TE) MIN MEM (BYTE)	MAX MEM (BYTE)	REC MEM (BYTE)	
n- Integrations	jean3-61	jek8s-361	24_hours	4.1K	3.47K	4.77K	4.9K	14.7	72G 0	31.07G	33.57G	
T APM	jean3-61	jek8s-364	24_hours	1.09K	884.51	1.25K	1.22K	9	.9G 7.26G	14.04G	15.64G	
Notebooks +	jean3-61	jek8s-363	24_hours	851.84	205.74	1.13K	984	7.2	16G 3.35G	8.34G	9.17G	
珍 Logs 💡	Jeans-61	Jekos-302	24_100/5	430.32	320.23	0/3./9	LUIK	0.4	3.334	13,490	14.813	
C Security												
亲 UX Monitoring 。												
					Nodes	Resource Utiliza	tion History					
	Node Current	/Predicted C	PU Usage (Dai	ly) 2d	Node Current/Pro	edicted CPU Usag	e (Weekly)	1w No	ode Current/Predicte	d CPU Usage (Mont	thiy) Imo	
	8K			1.00	156			15	kubernetes.cpu.usage.total	millicore{	917 B	
	4K WOAM	Marman	MA March	Mind and Mind	10K	,	1 1	1	kube_cluster:jean3-61, host;	ek8s-362) 1.3K	I.I. all	
	2K	man	- Antonia - and	and the second second	Sx when have	south (	mandagood	in s	x G	The for the	Mag Ching	
tt Team	OK W	ed 19	12:00 1	Thu 20 12:00	OK Sat 15	Mon 17	Wed 19	0	к јиј 26 л	ug 2 08/06 18:00 9	Aug 16	
root@prophet	Value Min 5.49K 1.22	Avg Max 2K 4.36K 7.49K	Metric kubernetes.cpu.us	Tags↓ a host:jek8s-361,kube	Value Min A 3.58K 704.92	vg Max Metric 1.48K 11.93K kubernet	Tags↓ s.cpu.usa host:jek8s-3	61,kube.,	Value Min Avg M 1.01K 536.86 3.44K 12	ax Metric .37K kubernetes.cpu.us	Tags ↓ host.jek8s-361,kub	
ProphetStor Data	3.72K 2.90	IK 4.21K 5.16K	federatorai.predic	ti host jek8s-361, kube	2.71K 2.86K	3.94K 5.24K federator	ai.predicti host:jek8s-3	61,kube	55.76 41.25 882.98 5	.23K federatoral.predict	host:jek8s-361,kub	

# ProphetStor Federator.ai Application Overview

	0.1	🛧 ProphetSto	or Federator.a	ii Applicatic	on Overview 🗸	Edit Widgets +						1h Past 1	Hour		 <b>Ⅱ</b>
		Save or select views	\$kube_cluster jean3-	61 ¥ \$kube_name	espace * • \$kube_de	ployment * 👻 \$1	kube_stateful_set * •	1							
D/	ATADOG														
1	New Stuff!					Aş	plication Work	load Pred	liction/Re	source Recom	mendation				
	Watchdog														
_			Workload Prediction for Next 24 Hours												
-	Events				AME KUBE_DEPLO	KUBE_STATEF AV	G CPU (MC MIN	CPU (M N	MAX CPU (M	REC CPU (MC	AVG MEM (B	MIN MEM (B	MAX MEM (B	. REC MEM (BY	
	Dashboards >	Прор		nginx-pr	eloader nginx-prepared	N/A	335.86	240.93	437.68	394	3.21M	584.89K	6.18N	11.22M	
•	Infrastructure 💛	PRUP	HEIJIOK	myproje	ct consumer1-top	N/A	87.12	64.47	199.68	133	2.26G	2.12G	2.350	2.38G	
θ	Monitors >			myproje	consumer2-top	N/A	/1.6/	30.00	111.20	00	1.778	1.174	2.770	2.980	
<i>C</i> 4	Metrics >														
÷.	Integrations														
-	APM >	Workload Predi	ction for Next 7 D	avs											
	Notebooks >	KUBE_NAMESPACE	KUBE_DEPLOYMENT	KUBE_STATEFUL_S	AVG CPU (MCORES)	MIN CPU (MCORES)	MAX CPU (MCORES	) REC CPU	(MCORES)	AVG MEM (BYTES)	MIN MEM (B	YTES) MAX M	EM (BYTES)	REC MEM (BYTES)	
	1	nginx-preloader-sar	r nginx-prepared	N/A	244.9	134.71	398.	5	688	6.42M	e	.42M	6.42M	5.87M	
Q.	Logs	myproject	consumer2-topic000	N/A	68.94	57.33	87.	1	56	2.35G		2.01G	2.63G	1.82G	
	Security >	myproject	consumer1-topic000	N/A	59.57	59.57	59.5	7	51	1.7G		.43G	1.97G	1.84G	
C49(	UX Monitoring														
		Workload Predi	ction for Next 30 [	Days											
		KUBE_NAMESPACE	KUBE_DEPLOYMENT	KUBE_STATEFUL_S	AVG CPU (MCORES)	MIN CPU (MCORES)	MAX CPU (MCORES	) REC CPU	(MCORES)	AVG MEM (BYTES)	MIN MEM (B	YTES) MAX M	EM (BYTES)	REC MEM (BYTES)	
		nginx-preloader-sar	r nginx-prepared	N/A	429.82	391.58	471.5		451	7.96M		/./M	8.26M	9.09M	
		myproject	consumer2-topic000	N/A	79.01	75.06	78.8	9	79	1.80G	858	.94M	2.02G	2.36G 2.23G	
0	Help >		constructive receptode		10.03	10.05	10.0			1.440	0.0		1.520	1.230	
<u>2</u> 2	Team														
	root@prophet ProphetStor Data					Application	Resource Utilia	ation His	story						

## ProphetStor Federator.ai Kafka Overview

978	★ ProphetStor Federator.ai Kafka Overview ∨ Clone Das	shboard 1h Past 1 Hour	- <b>≪ Ⅱ</b> ≫ Q
	Skube_cluster jean3-61 v Skube_namespace myproject v Skube_deployment consumer1	-topic0001-group-0001 v Stopic topic0001 v Sconsumer_group group0001 v	···· 📮 💠
DATADOG			
<ul> <li>✓ New Stuff! →</li> <li>Matchdog</li> <li>Events</li> <li>Dashboards →</li> </ul>	With integration of Prop production/consumption dashboard. Based on the consumer replicas to har recommended consume overall consumer rare also show	hetStor Federator.ai, users can easily track the Kafka message n rate, as well as the prediction of message production rate from Federator.ai prediction or message production rate, Federator.ai automatically scales Kafka and le the workdoad. This can be visualized from Federator.ai dashboard where the r replicas and the current number of consumer replicas are shown. Additionally, s well as the average latency in the quee before a message is received by a n on the dashboard for better performance monitoring.	
¶ Infrastructure →	Recommended Replicas vs Current/Desired Replicas	Production vs Consumption vs Production Prediction	
Monitors	8	2006	
APM	2 0	0K-13:45 14:00 14:15 14:30	
🖉 Notebooks 🕨	Kafka Consumer Lag	Consumer Queue Latency (msec) 20K	
	4K	15K 10K	
🎄 UX Monitoring →	28	consumer_group:group0001, kube_cluster:jean3-61, topic:topic0001	
	0K 13:45 14:00 14:15 14:30	0K 13:45 14:00 14:15 14:33:20	
	Deployment Memory Usage	Deployment CPU Usage	
♥     Help       ▲     Team       ▲     root@prophet       ProphetStor Dat	2	20 May May May May Mark Mark Mark Mark Mark Mark Mark Mark	

# ProphetStor Federator.ai Cost Analysis Overview

	✤ ProphetSto	or Federator.ai C	ost Analysis	Overview ~	dit Widgets +				1h Past 1 Hour		- 44 II 🕨
	Save or select views	\$kube_cluster jean3-61	spricing_option on	demand-instance * Scour	ntry usa 💌	/					<b>200</b>
ADOG							Multiclou	d Cost Analysis			
w Stuffi							marciciou	a cost Analysis			
chdog			Current Clurt	r Cort				Current Cluste	r Configuration		
			Current Cluste	n cost Elle	HOST	KUBE NODE ROLE	AVAILABILITY-ZONE	INSTANCE-TYPE	SIZE	CPU CAPACITY	MEMORY CAPACITY
са.					jek8s-361	master	N/A	N/A	N/A	8 cpus	31.26 GiB
boards +	Рворн	FTSTOR			jek8s-363	N/A	N/A	N/A	N/A	4 cpus	15.51 GiB
structure 👌			2078	52	jek8s-362	N/A	N/A	N/A	N/A	4 cpus	15.51 GiB
itors			2070	••••••••••••••••••••••••••••••••••••••	jek8s-364	N/A	N/A	N/A	N/A	4 cpus	15.51 GiB
rics +											
arations											
gradure	Recommended C	luster - AWS 1d	Recommende	d Cluster Configurati	on - AWS						
			PROVIDER	DISPLAY_NAME	REGION	INSTANCE_TYPE	ONDEMAND_INSTAN	RESERVED_INSTANCE	ONDEMAND_INSTAN	RESERVED_INSTANCE	COUNTRY
ibooks +	839	.22 s/mo	aws	m5.4xlarge_16.0_cor	es_ us_west_or	regon m5.4xlarge	1	0	0	0	usa
i 8	$\sim$	Savings 1d	aws	m5.xlarge_4.0_cores	16 us_west_or	regon m5.xlarge	0	0	2	0	usa
irity +	aws	1239.30	aws	c5.large_2.0_cores_4	.0_ç us_west_or	regon c5.large	1	0	2	0	usa
tonitoring i											
	Recommended Cluster - Azure 11		Recommende	d Cluster Configurati	on - Azure						
	706	02	PROVIDER	DISPLAY_NAME	REGION	INSTANCE_TYPE	ONDEMAND_INSTAN	RESERVED_INSTANCE	ONDEMAND_INSTAN	RESERVED_INSTANCE	COUNTRY
	/00	.02 s/mo	azure	standard-d16s-v3_16	5.0_ east_us	standard-d16s-v3	1	0	0	0	usa
	^	Savings 1d	azure	standard-d4s-v3_4.0	_co_east_us	standard-d4s-v3	0	0	2	0	usa
	Azure	1291.70	azure	standard-f2s-v2_2.0_	cor east_us	standard-f2s-v2	1	0	2	0	usa
	Recommended C	luster - GCP 1d	Recommende	d Cluster Configurati	on - GCP						
		4.5	PROVIDER	DISPLAY_NAME	REGION	INSTANCE_TYPE	ONDEMAND_INSTAN	RESERVED_INSTANCE	ONDEMAND_INSTAN	RESERVED_INSTANCE	COUNTRY
	570	.15 s/mo	gcp	e2-standard-16_16.0	_co_us_west_1	e2-standard-16	1	0	0	0	usa
	•	Savings 1d	gcp	e2-standard-4_4.0_co	ore us_west_1	e2-standard-4	0	0	2	0	usa
n			gcp	e2-standard-2_2.0_cd	ore us west 1	e2-standard-2	1	0	2	0	usa

# **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
                                                                READY STATUS RESTARTS AGE
NAME
                                                                2/2
                                                                        Running 2
                                                                                               26d
datadog-monitoring-6j2k4
datadog-monitoring-8frn72/2datadog-monitoring-cluster-agent-5cdcbb8747-b2br21/1datadog-monitoring-kube-state-metrics-769f55fd64-tptz71/1
                                                                        Running 2
Running 1
Running 1
Running 2
                                                                                               26d
                                                                                               26d
                                                                                               26d
                                                                2/2
datadog-monitoring-mlzdm
                                                                                               26d
                                                                        Running 1
wpacontroller-watermarkpodautoscaler-68484f8dd4-cqx65 1/1
                                                                                               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
```

 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

<pre>~# kubectl get pod -n default</pre>				
NAME	READY	STATUS	RESTART	S AGE
datadog-agent-2m6kk	1/1	Running	2	2d
datadog-agent-8kd54	1/1	Running	0	2d
datadog-agent-94r16	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

{"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./capr.go: 128\ngithub.com/DataDog/watermarkpodautoscaler/pkg/controller/watermarkpodautoscaler.(\*ReconcileWa termarkPodAutoscaler).Reconcile\n\twatermarkpodautoscaler/pkg/controller/watermarkpodautoscaler.(\*ReconcileWa termarkPodAutoscaler).Reconcile\n\twatermarkpodautoscaler/pkg/controller/watermarkpodautoscaler.(\*Action of the second termarkpodautoscaler).Reconcile\n\twatermarkpodautoscaler.

runtime/pkg/internal/controller.(\*Controller).reconcileHandler\n\twatermarkpodautoscaler/vendor/sigs.k8s.io/controller.runtime/pkg/internal/controller.go:216\nsigs.k8s.io/controller-

runtime/pkg/internal/controller.(\*Controller).processNextWorkItem\n\twatermarkpodautoscaler/vendor/sigs.k8s.io/controller-runtime/pkg/internal/controller.go:192\nsigs.k8s.io/controller-

runtime/pkg/internal/controller.(\*Controller).worker\n\twatermarkpodautoscaler/vendor/sigs.k8s.io/controller-

runtime/pkg/internal/controller.go:171\nk8s.io/apimachinery/pkg/util/wait.JitterUntil.func1\n\twatermarkpodautoscal er/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:152\nk8s.io/apimachinery/pkg/util/wait.JitterUntil\n\twatermarkpodautoscal ler/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait.Until\n\twatermarkpodautoscal ler/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait.Until\n\twatermarkpodautoscal ler/vendor/k8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait.Until\n\twatermarkpodautoscal ler/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:153\nk8s.io/apimachinery/pkg/util/wait.Until\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:153\nk8s.io/apimachinery/pkg/util/wait.Until\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:153\nk8s.io/apimachinery/pkg/util/wait.Until\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:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pkg/util/wait/wait.go:153\nk8s.io/apimachinery/pk

- 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: \leftarrow comment out
                 ← comment out
  # status: {}
  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