Federator.ai Release v5.0
Installation Guide

Contents

Overview

Federator.ai

ProphetStor Federator.ai is an AI-based solution that helps enterprises manage and optimize resources for applications on Kubernetes and virtual machines (VMs) in VMware or AWS EC2 clusters. Using advanced machine learning algorithms to predict application workloads, Federator.ai offers:

- AI-based workload prediction for containerized applications in Kubernetes clusters and VMs in VMware or AWS EC2 clusters
- Resource recommendations based on workload prediction, application, Kubernetes, and other related metrics
- Correlation and causality analysis of microservices/controllers of Kubernetes applications
- Automatic scaling of Kubernetes application containers, Kafka consumer groups, and Ingress upstream
- Multicloud cost analysis and recommendations based on workload predictions for Kubernetes clusters and VM clusters
- Actual cost and potential savings based on recommendations for clusters, Kubernetes applications, VMs, and Kubernetes namespaces
- Correlation and causality analysis of microservices/controllers of Kubernetes applications
- Statistical analysis and predictions based on the correlation between resource usage and application workload

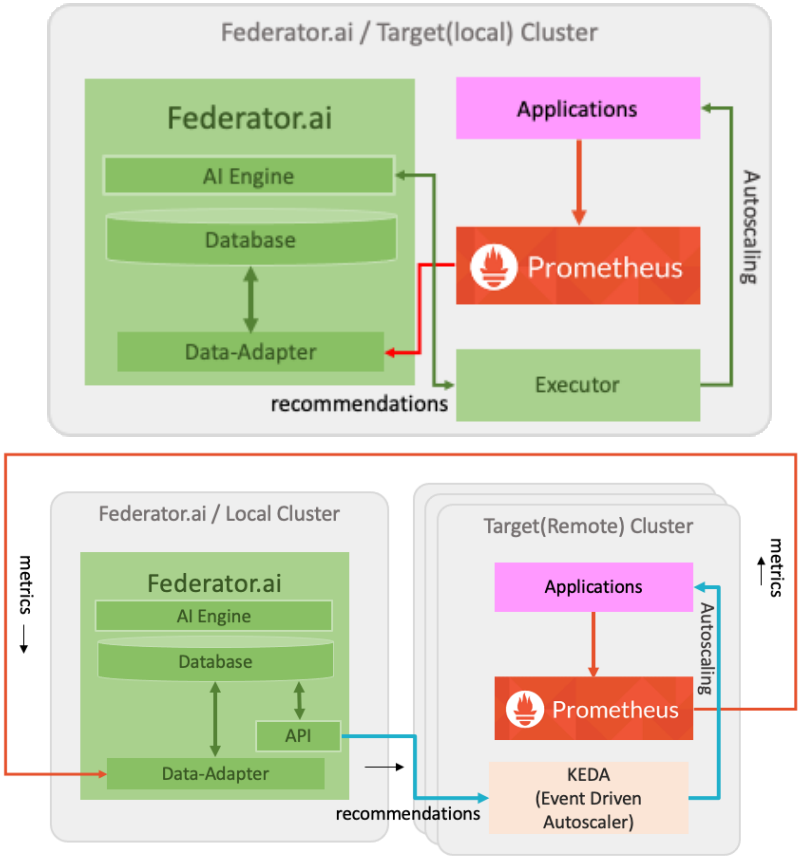
Supported Metrics Data Sources

There are five different metrics data sources supported in released v5.0.0-ga: Prometheus, Datadog, Sysdig, VMWare vCenter, and AWS CloudWatch.

Prometheus

Prometheus is a free and open-source event monitoring tool for containers or microservices. It uses scraping to collect numerical data based on time series. Metrics are collected in regular timestamps and stored locally. Federator.ai supports using Prometheus gathering Kubernetes cluster metrics and leveraging collected data for workload predictions, recommendations for resource planning, autoscaling containers/pods, and cost analysis for clusters deployed in a multicloud environment.

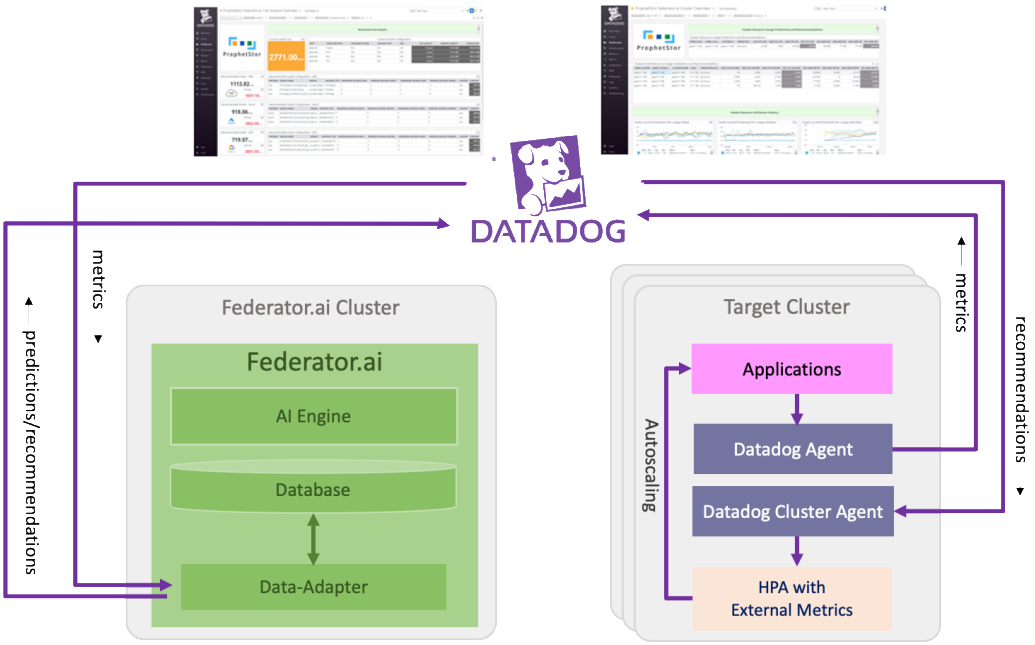
The following diagram shows how the metrics are collected from Prometheus by Federator.ai in a Kubernetes environment.



Datadog

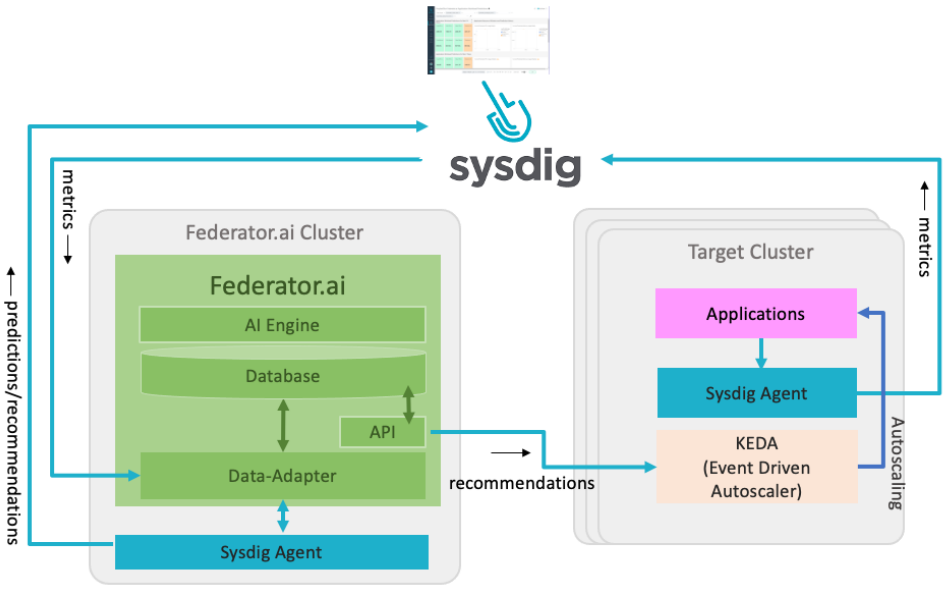
Fedederator.ai has integrated with Datadog and utilizes the metrics collected by Datadog Agent for workload predictions. The following diagram shows how application metrics are used by Federator.ai to predict workload and to scale applications for better performance automatically. Specifically,

- Datadog Agent sends cluster/applications metrics to Datadog Services
- Federator.ai's Data-adapter queries cluster/applications metrics from Datadog Services and forwards 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
- Datadog Dashboard displays cluster/applications metrics and prediction/recommendation/plan by Federator.ai



Sysdig

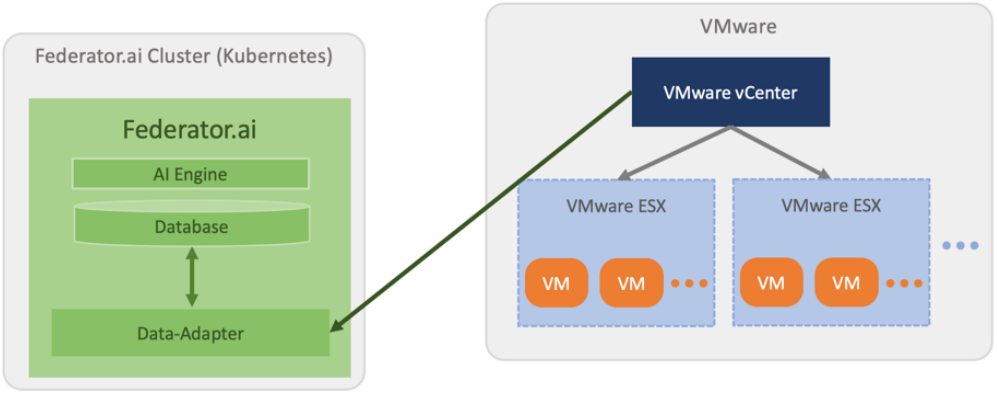
Fedederator.ai has integrated with Sysdig and utilizes the metrics collected by Sysdig Agent for workload predictions. The following diagram shows how application metrics are used by Federator.ai to predict workload and to autoscale applications for better performance and saving resources.



VMWare vCenter

VMware vCenter Server provides integrated management of all hosts and virtual machines in the data center from a single console, allowing IT administrators to improve control, simplify daily work, and reduce the complexity and cost of managing the IT environment.

Federator.ai data adapter connects to VMware vCenter servers via VMware SDK to retrieve VMs workload metrics data for predictions, recommendations, and cost analysis for VM clusters.



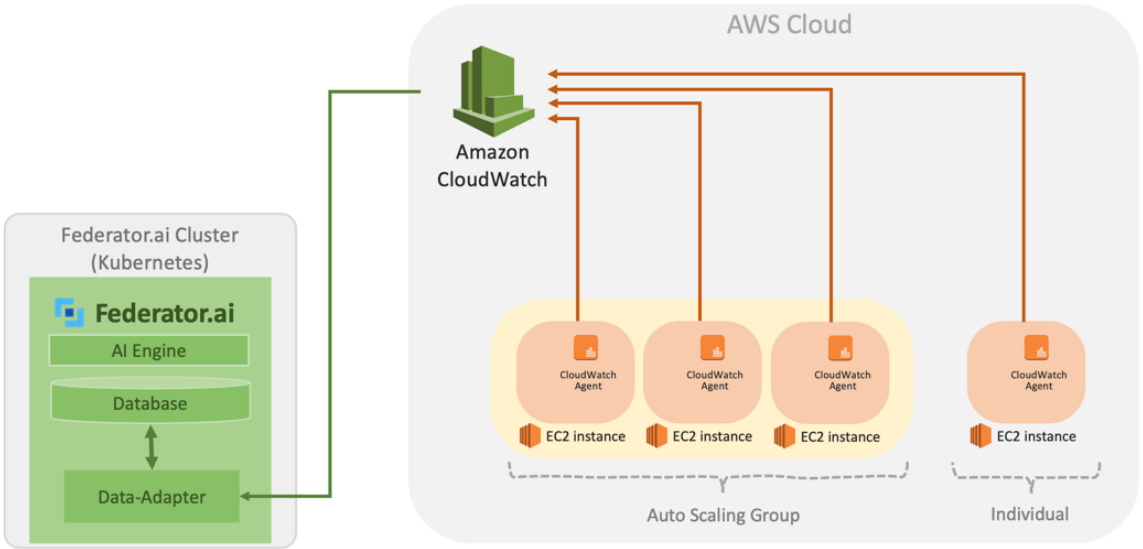
AWS CloudWatch

AWS CloudWatch is a monitoring service for AWS cloud resources and the applications running on the AWS cloud. It provides visibility into resource utilization, operational performance, and overall demand patterns.

The metrics collected by CloudWatch by default do not include memory usage of EC2 instances. Therefore, CloudWatch agent is required for Federator.ai to collect memory usage metrics.

Federator.ai supports two types of AWS VM clusters:

- Auto Scaling Group
- Individual VM



Requirements and Recommended Resource Configuration

Supported Platform

- OpenShift : 4.6~4.9
- Kubernetes : 1.11 ~ 1.22
- Rancher v2.4.8,v2.5.8,v2.5.9
- EKS/AKS/GKE

Data Source

- Datadog
- Sysdig
- Prometheus
- VMWare vCenter 5.5/6.0/6.5/6.7/7.0
- AWS CloudWatch

Federator.ai Resource Requirements

- Total Resource Requirments
  - Request: 5.1 CPU cores (Limit: 22 cores)
  - Request: 5.1 GB Memory (Limit: 42GB)
  - StorageClass: 176GB (require ReadWriteOnce access mode)
- Resource requirements for AI Engine
  - There must be at least one worker node with at least 2 CPU(Limit: 8 cores) cores and 1 GB of memory available
  - The 2 CPU cores and 1 GB memory are included in the total 5.1 CPU cores and 5.1 GB memory requirements
- Persistent volumes
  - The StorageClass that provides the persistent volumes must support RWO (ReadWriteOnce) access mode.
  - It is recommended to use persistent volumes instead of ephemeral storage to store the data in the production environment. Any data on ephemeral storage will be lost after Federator.ai pods are restarted.

Federator.ai Version

- Version: Release 5.0-ga
- Tag : v5.0.0-ga

Datadog Agent Version(reference)

- Datadog Agent helm chart version: v2.4.24, v2.13.0
- Datadog Agent version: v7.21.1, v7.27.0
- Datadog Cluster Agent version: v1.8.0, v1.12.0

- Datadog Watermark Pod Autoscaler version: v0.1.0

Prometheus Version(reference)

- OpenShift
  - Default installed Prometheus
- SUSE Rancher
  - Cattle-Prometheus
- Kubernetes
  - prometheus-operator-8.5.11
  - kube-prometheus-release-0.6
  - kube-prometheus-stack-12.5.0/15.4.6/17.0.3

Sysdig Agent Version(reference)

- Sysdig agent: 11.2.0/11.3.0

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: Preparation

- For Datadog, obtain API Key, Application Key of Datadog Cloud Service account. Instructions are provided below.
- For Prometheus, obtain Prometheus service URL (ex : *http://<prometheus_svc_name>.<namesapce>:9090*)
- For Sysdig, obtain Sysdig API URL and Token.
- For VMware vCenter, obtain adminstrator login credential and vCenter IP or FQDN.
- For AWS CloudWatch, obtain Access Key ID and Secret Access Key of AWS account.

Step 2:

- For Datadog, install and configure Datadog Agent/Cluster Agent if they have not been installed. Please follow the Datadog documentation on how to install Datadog Agent and Cluster Agent.
- For Sysdig, install and configure Sysdig Agent. Please follow Sysdig documentation on how to install Sysdig Agent.

Step 3: Install Federator.ai.

Step 4: Configure Federator.ai Data Adapter for the external metrics data source via Federator.ai Initial Setup Wizard.

Step 5: Check integrated Federator.ai dashboard on Datadog/Sysdig Cloud.

Pre-installation Check List

Kubernetes Access Management Requirement

- ClusterRole/ClusterRoleBinding, Role/RoleBinding

Federator.ai Feature	Component	Scope	Operation
Installation, Lifecycle Management	federatorai-operator	Cluster	admin
Data Collection/Controller	alameda-datahub	Cluster	edit
	alameda-operator	Cluster	edit
	federatorai-data-adapter	Cluster	view
Dashboard	federatorai-dashboard-frontend	Local	view
	federatorai-dashboard-backend	Cluster	view

Prediction	alameda-ai	Local	view
	ai-dispatcher	Local	view
Recommendation	alameda-recommender	Local	view
Autoscaling Execution	alameda-executor	Cluster	edit
Workload Anomaly Notification	alameda-analyzer	Cluster	view
	alameda-rabbitmq	Local	view
	alameda-notifier	Cluster	edit
Cost Analysis	fedemeter-api	Local	view
	federatorai-agent	Local	view
Database	alameda-influxdb	Local	view
	federmeter-influxdb	Local	view
REST API	federatorai-rest	Cluster	edit

Installation Script Run Permission Requirement

- Installation scripts do NOT require root permissions to run in the host where "kubectl" command is set up
  - The default script download path is "/opt", if the non-root user does not have the permission to write "/opt", input a different download path during installation.

```
$ curl -s https://raw.githubusercontent.com/containers-ai/prophetstor/master/deploy/federatorai-launcher.sh | bash
Please enter Federator.ai version tag [default: latest]:
Federator.ai version = v5.0.0-ga
Please enter the path of Federator.ai directory [default: /opt]:
```

- Installataion scripts support Linux OS only
  - macOS and Windows are not supported

Kubernetes:

#	Checklist Item	Requirement	Details
1	What is the Kubernetes version?	Server Version: 1.11~1.22	Use the command below to get the 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 this Kubernetes cluster require a private image repository?	If a 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 the Federator.ai installation script asks for the information.
3	StorageClass and Persistent Volumes requirement	StorageClass supports ReadWriteOnce access mode. Available storage size is larger than 176GB.	The minimum storage size for Federator.ai Release v5.0 is 176GB, including database, data, and logs.
4	Kubernetes cluster CPU/memory requirement	Minimum CPU/mem/storage: - CPU: 5,100 (mcores) - Memory: 5.1 (GB) - Storage Class Capacity: 168GB At least one worker node with - CPU: 2 Cores - Memory: 1GB	To be able to run the AI Engine pod, there must be at least one worker node that has more than 2 CPU cores and 1 GB of memory available. 2 CPU Cores and 1GB for AI Engine are included in the total 5.1 CPU Cores and 5.1 GB memory requirements.
5	Is this 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 the 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 the minimum resource requirement - CPU: 5.1 Cores - Memory: 5.1 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 the number of managed clusters/applications increases. Use the command to get namespace resource quota $ **kubectl get resourcequota --all-namespaces**
7	Does this deployment require 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

Prometheus:

#	Checklist Item	Requirement	Details
1	What is the Prometheus version? (for Kubernetes)	Recommended version - Prometheus operator helm chart version: 8.5.11 - Prometheus operator version: 0.34.0 - Prometheus server version: 2.13.1	Use the command below to get Prometheus version: $ **helm ls -A |grep -i prometheus** prometheus-adapter monitoring 1 2020-03-13 15:35:05.28963154 +0800 CST deployed prometheus-adapter-2.1.3 v0.6.0 prometheus-operator monitoring 1 2020-03-13 14:34:16.132479221 +0800 CST deployed prometheus-operator-8.12.1 0.37.0

			$ **kubectl get deployment -A -o custom-columns=IMAGE:.spec.template.spec.containers[0].image |grep -i prometheus** directxman12/k8s-prometheus-adapter-amd64: v0.6.0 quay.io/coreos/prometheus-operator: v0.37.0

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 for the HPA autoscaling feature	Cluster Agent provides metrics to HPA Autoscaler for autoscaling.
3	Is Datadog WPA controller installed? (Option)	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?(Option)	Datadog Kafka Consumer integration is mandatory if user wants to use Kafka consumer optimization feature	Use the command to confirm Kafka integration is enabled $ **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	An 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	An application key is mandatory for connecting Datadog Service	Follow the steps described in the "Before You Start" session to obtain the Application key.
7	Is one of cluster name is configurated for the Datadog agent/cluster agent? 1.>DD_TAGS with value ="kube_cluster:<cluster_name>" in values.yaml or 2.>"cluster_name" in values.yaml, or 3.>"DD_CLUSTER_NAME" in Datadog cluster agent deployment	"kube_cluster","cluster_name","kube_cluster_name(DD_CLSUTER_NAME)" one of them is required for Federator.ai to identify Kubernetes cluster.	Case 1.>New Datadog Agent installation: Install Datadog agent and cluster agent by "helm install -f values.yaml", in values.yaml. ... **clusterName: <cluster-name>** ... clusterAgent: enabled: ~~false~~ true Case 2.> In Datadog Agent installed environment, with no Cluster Agent and no cluster_name setting Update Datadog Agent to enable Cluster agent by "helm upgrade -f values.yaml", in values.yaml - assign a cluster name ... datadog: clusterName: <cluster-name> - enable cluster agent ... clusterAgent: enabled: ~~false~~ true ... - $helm upgrade ...

			- Check “DD_Cluster_Name” **$kubectl get daemonset *<datadog_agent_daemonset_name>* -n *<datadog_agnet_namespace>* -o yaml** - name: DD_CLUSTER_NAME value: <cluster-name> 3.>In Datadog Agent and Cluster Agent installed environment, with no cluster_name setting Update Datadog Agent by “helm upgrade” - assign a cluster name datadog: clusterName: <cluster-name> - $helm upgrade ... - Check “DD_Cluster_Name” **$kubectl get daemonset *<datadog_agent_daemonset_name>* -n *<datadog_agnet_namespace>* -o yaml** As: - name: DD_CLUSTER_NAME value: <cluster-name> 4.> In Datadog Agent and Cluster Agent installed environment, with cluster_name setting Use the command below to confirm DD_Cluster_Name - $**kubectl get daemonset *<datadog_agent_daemonset_name>* -n *<datadog_agnet_namespace>* -o yaml** As: - name: DD_CLUSTER_NAME value: <cluster-name>

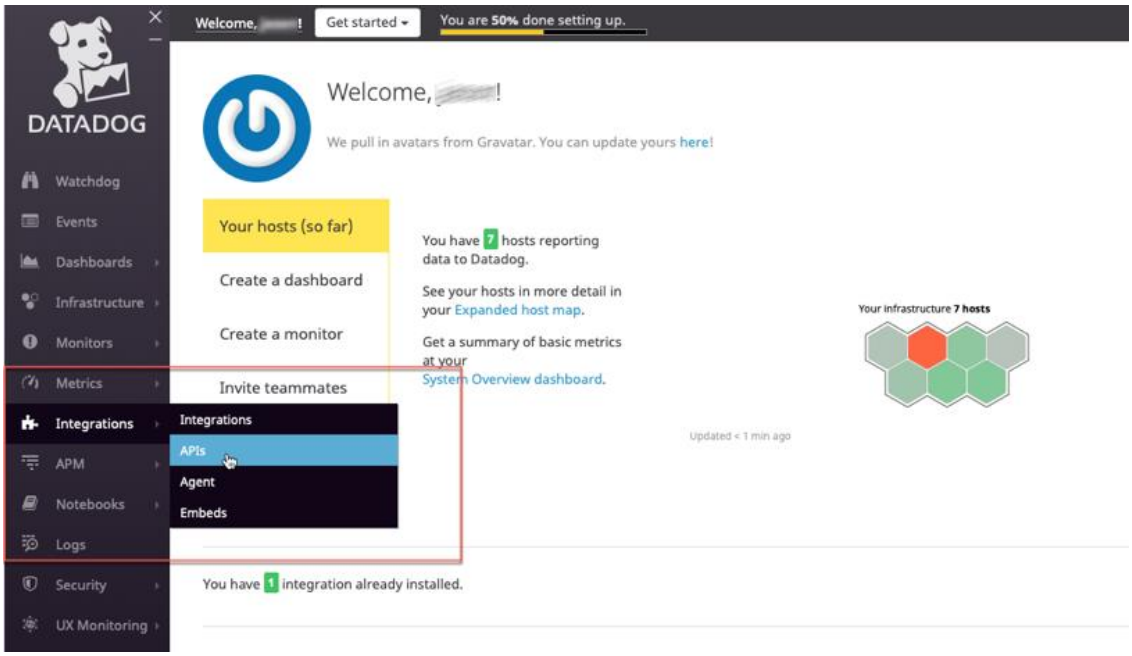
Before You Start

Datadog

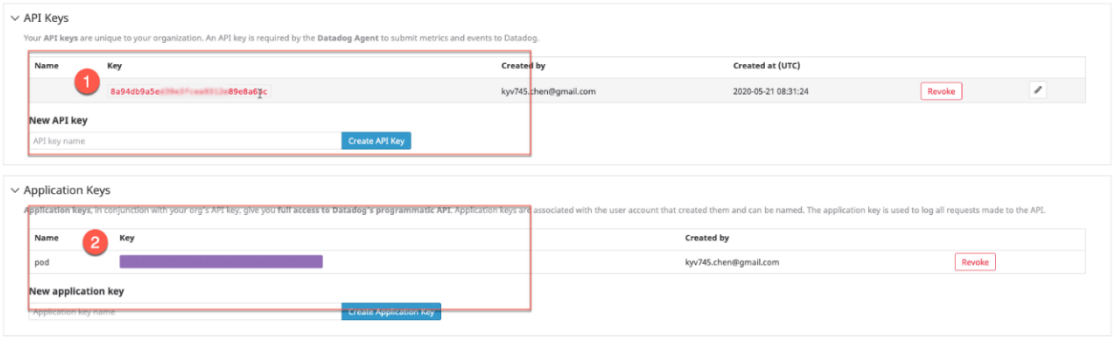
- The admin role for installing Fedeator.ai is "Cluster Admin."
- Datadog agent must be ready if Federator.ai runs in the same Kubernetes cluster monitored.
- Obtain Datadog account API Key, Application Key.
  1. A Datadog account is required for connecting and using Datadog Cloud Service. If you don't 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/

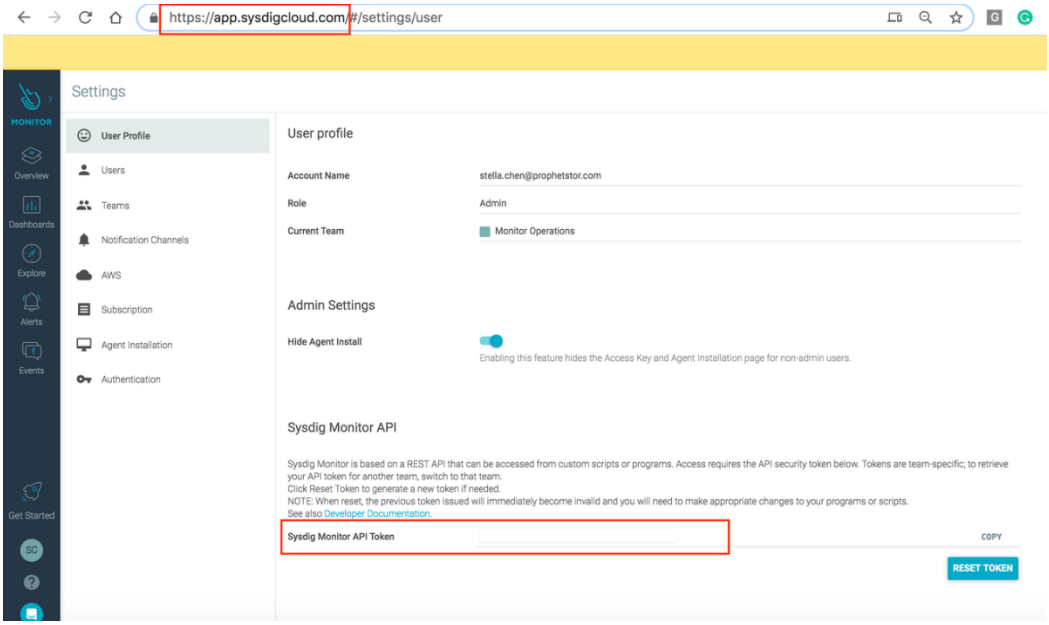


Copy the API Key and Application Key for Federator.ai metrics data source configuration



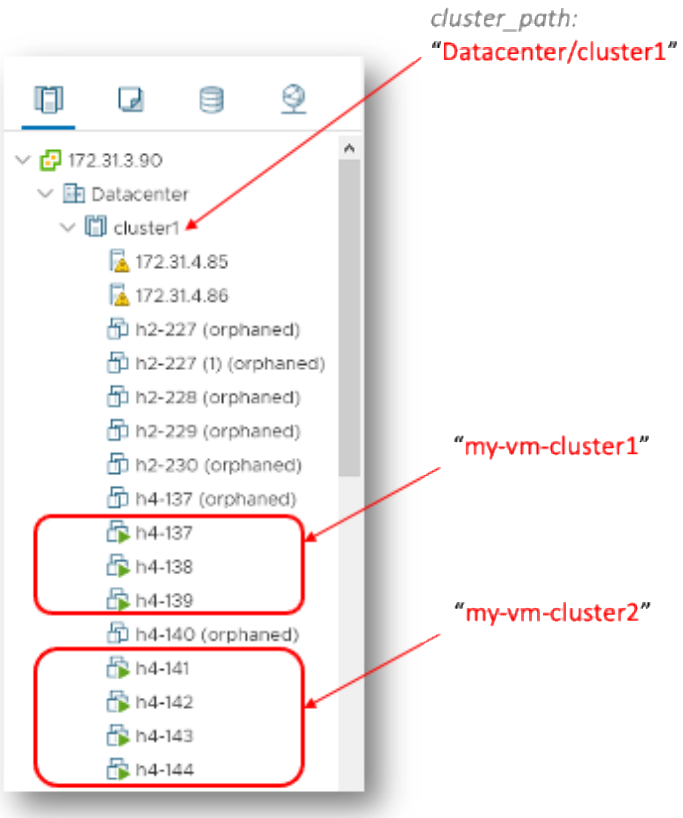
Sysdig

- Different Sysdig API URL is needed for different regions :
  - For US East, Sysdig API URL is https://app.sysdigcloud.com
  - For US West, Sysdig API URL is https://us2.app.sysdig.com
  - For European Union, Sysdig API URL is https://eu1.app.sysdig.com
- Copy Sysdig Monitor API Token for Federator.ai metrics data source configuration



VMware vCenter

- You can define a VM cluster from any VMs under the same cluster path. See below for an example of cluster path on vCenter.

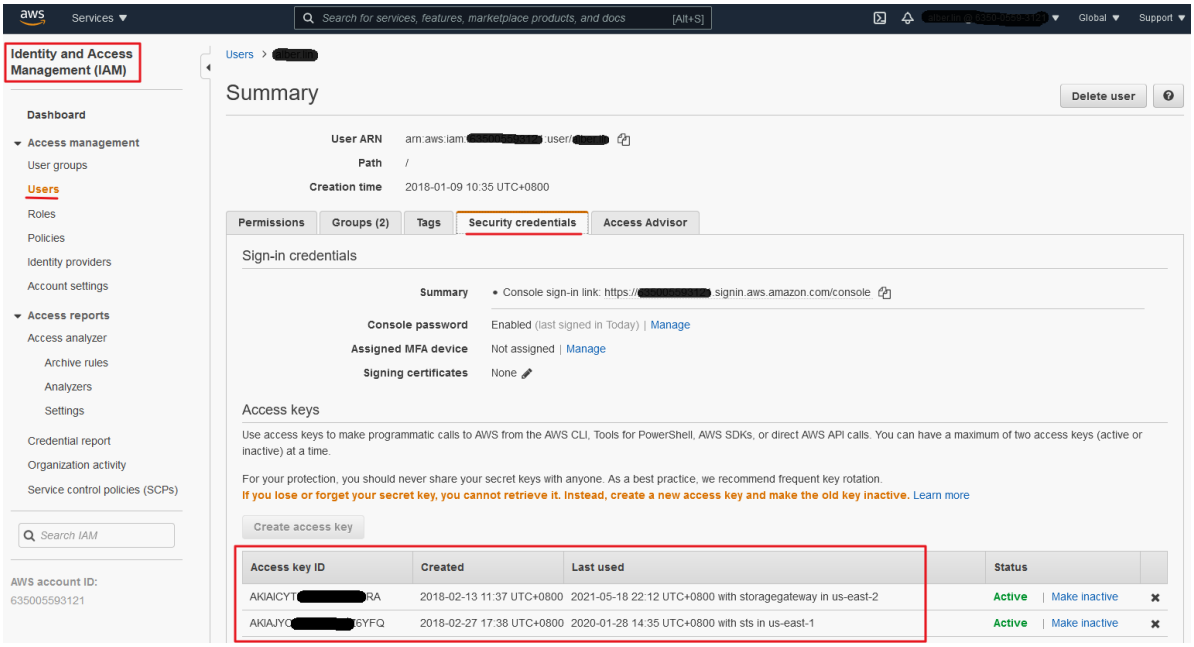


AWS CloudWatch

- Obtain CloudWatch Account Key ID and Secret Access Key.

1. Use your AWS account ID or account alias, your IAM username, and your password to sign in to the [IAM console](#).

2. Go to "Access management > Users > Security credentials" to get Access Key ID and Secret Access Key



https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Brand New Installation

1. Connect to Kubernetes cluster

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

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

```
~$ curl https://raw.githubusercontent.com/containers-
ai/prophetstor/master/deploy/federatorai-launcher.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 17101  100 17101    0     0  30118      0 --:--:-- --:--:-- --:--:-- 30107
Please enter Federator.ai version tag [default: latest]:
Federator.ai version = v5.0.0-ga
Please enter the path of Federator.ai directory [default: /opt]:

Downloading v5.0.0-ga tgz file ...
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 = v5.0.0-ga
install_namespace = federatorai
----------------------------------------
Is the above information correct? [default: y]:

Downloading v5.0.0-ga tgz file ...
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
Applying 08-service.yaml...
```

```
service/federatorai-operator-service created
Applying 09-secret.yaml...
secret/federatorai-operator-service-cert created
Applying 10-mutatingwebhook.yaml...
mutatingwebhookconfiguration.admissionregistration.k8s.io/federatorai-operator-
servicesmutation created
Applying 11-validatingwebhook.yaml...
validatingwebhookconfiguration.admissionregistration.k8s.io/federatorai-operator-
servicesvalidation created

Checking pods...
Waiting for pod federatorai-operator-669566b7c-rmphp in namespace federatorai to be ready.
phase: [Running]
Waiting for pods in namespace federatorai to be ready...

All pods under namespace(federatorai) are ready.

Install Federator.ai operator v5.0.0-ga successfully

Downloading Federator.ai alamedaservice sample file ...
Done

Downloading Federator.ai alamedascaler sample files ...
Done
========================================
Which storage type you would like to use? ephemeral or persistent?
[default: persistent]:
Specify log storage size [e.g., 2 for 2GB, default: 2]:
Specify AI engine storage size [e.g., 10 for 10GB, default: 10]:
Specify InfluxDB storage size [e.g., 100 for 100GB, default: 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 = 2 GB
AI engine 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(v5.0.0-ga) pod to appear ...

datahub pod is present.


Checking pods...
Waiting for pod alameda-ai-66f5c7b6b4-rx87j in namespace federatorai to be ready. phase:
[Pending]
Waiting for pods in namespace federatorai to be ready...
Waiting for pod alameda-ai-66f5c7b6b4-rx87j in namespace federatorai to be ready. phase:
[Pending]
Waiting for pods in namespace federatorai to be ready...
Waiting for pod alameda-ai-66f5c7b6b4-rx87j in namespace federatorai to be ready. phase:
[Pending]
Waiting for pods in namespace federatorai to be ready...
```

```
Waiting for pod alameda-ai-66f5c7b6b4-rx87j in namespace federatorai to be ready. phase:
[Running]
Waiting for pods in namespace federatorai to be ready...
Waiting for pod alameda-operator-7ff69f4bb5-v22ws in namespace federatorai to be ready.
phase: [Running]
Waiting for pods in namespace federatorai to be ready...

All pods under namespace(federatorai) are ready.

The default alamedaorganization under namespace federatorai is ready.

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

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

========================================
You can now access Federatorai REST API through https://<YOUR IP>:31011
The 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 v5.0.0-ga successfully

Downloaded YAML files are located under /opt/federatorai/installation

Downloaded files are located under /opt/federatorai/repo/v5.0.0-ga
```

3. Verify Federator.ai pods are running properly

```
~# kubectl get pod -n federatorai
NAME                                                READY   STATUS    RESTARTS   AGE
alameda-ai-66f784f79d-xphsb                         1/1     Running   0          7m37s
alameda-ai-dispatcher-694474f89-njmjb               1/1     Running   0          7m37s
alameda-datahub-679875645c-cc6kt                    1/1     Running   0          8m12s
alameda-executor-bf699b5fb-26r4n                    1/1     Running   0          7m37s
alameda-influxdb-0                                  1/1     Running   0          8m12s
alameda-notifier-b87744cdc-wwrgt                    1/1     Running   0          7m37s
alameda-rabbitmq-7f5869cdc8-5wt57                   1/1     Running   0          8m12s
fedemeter-api-64665c6ff4-gbfsn                      1/1     Running   0          8m12s
fedemeter-influxdb-0                                1/1     Running   0          8m12s
federatorai-agent-774cdfb9dd-bblkh                  1/1     Running   0          7m37s
federatorai-dashboard-backend-67b749597c-sqhlq      1/1     Running   0          7m37s
federatorai-dashboard-frontend-856458bf5c-pb9kb     1/1     Running   0          7m37s
federatorai-data-adapter-9f977cb5b-jbtqz            1/1     Running   0          7m37s
federatorai-operator-858cf75ffc-2tn6v               1/1     Running   0          11m
federatorai-postgresql-0                            1/1     Running   0          8m12s
federatorai-recommender-dispatcher-5c4d6c965f-6mfwf 1/1     Running   0          7m37s
federatorai-recommender-worker-585ccc7f55-q7lf7     1/1     Running   0          7m25s
federatorai-rest-855bfb6956-lcprj                   1/1     Running   0          7m37s
```

Federator.ai internal/external communication ports

- If NodePort is not permitted due to security reason, disable NodePorts from Federator.ai Operator
  - edit AlamesaService CR and remove "serviceExposures session "

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

...
serviceExposures:
  - name: federatorai-dashboard-frontend
    nodePort:
      ports:
      - nodePort: 31012
        port: 9001
    type: NodePort
  - name: federatorai-rest
    nodePort:
      ports:
      - nodePort: 31011
        port: 5056
    type: NodePort
  storages:
....
```

  - Or during installing process: "Expose dashboard and REST API services: (default:y)" step to answer 'N' to disable NodePort

```
~# curl https://raw.githubusercontent.com/containers-
ai/prophetstor/master/deploy/federatorai-launcher.sh | bash
...
...
Which storage type you would like to use? ephemeral or persistent?
[default: persistent]:
Specify log storage size [e.g., 2 for 2GB, default: 2]:
Specify AI engine storage size [e.g., 10 for 10GB, default: 10]:
Specify InfluxDB storage size [e.g., 100 for 100GB, default: 100]:
Specify storage class name: managed-nfs-storage
Do you want to expose dashboard and REST API services for external access? [default: y]:
...
...
```

**NGINX Ingress or LB tools may be used for external access to Federator.ai GUI if disabled NodePort.*

Connecting to Federator.ai Web portal

- Kubernetes/Rancher Cluster
  In a Kubernetes environment, use the kubectl command to find the administration portal service port number and node IP address.
  `$kubectl get svc -n federatorai |grep federatorai-dashboard-frontend-node-port`
  The output will look something like this:
  federatorai-dashboard-frontend-node-port NodePort 10.103.181.133 <none> 9001:31012/TCP
  Get the node's IP to access (INTERNAL-IP).
  `$kubectl get nodes -o wide`

```
# kubectl get nodes -o wide
NAME     STATUS   ROLES    AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION          CONTAINER-RUNTIME
h7-130   Ready    master   35d   v1.18.5   172.31.7.130   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
h7-131   Ready    <none>   35d   v1.18.5   172.31.7.131   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
h7-132   Ready    <none>   35d   v1.18.5   172.31.7.132   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
h7-133   Ready    <none>   35d   v1.18.5   172.31.7.133   <none>        CentOS Linux 7 (Core)
3.10.0-957.el7.x86_64   docker://19.3.13
```

(example) The URL will be **https://172.31.7.130:31012**

- OpenShift

  In an OpenShift environment, use the `oc get route` command to find the URL.

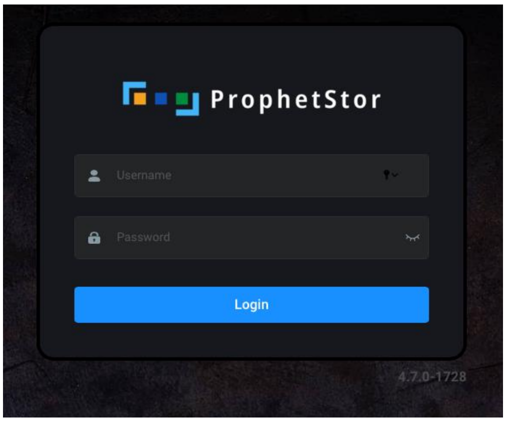
```
~# oc get route -n federatorai|grep federatorai-dashboard-frontend
```

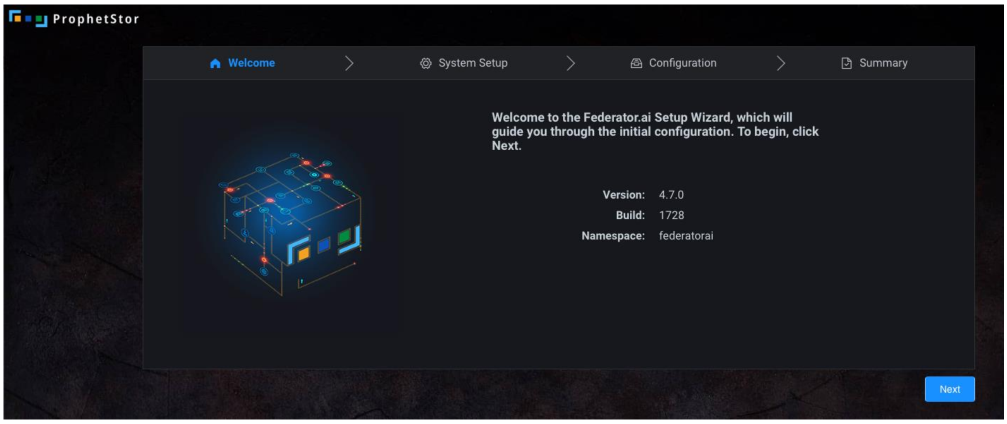
The output will look something like this:

```
NAME                           HOST/PORT                                                            PATH   SERVICES
                     PORT           TERMINATION    WILDCARD
federatorai-dashboard-frontend   federatorai-dashboard-frontend-federatorai.apps.ocp4.172-31-2-86.nip.io          federatorai-
dashboard-frontend   frontend-http   edge/Redirect   None
federatorai-rest                 federatorai-rest-federatorai.apps.ocp4.172-31-2-86.nip.io                        federatorai-
rest                 restapi-http    edge/Redirect   None
```

(example) The URL will be **https://federatorai-dashboard-frontend-federatorai.apps.ocp4.172-31-2- 86.nip.io**

Federator.ai Web Portal



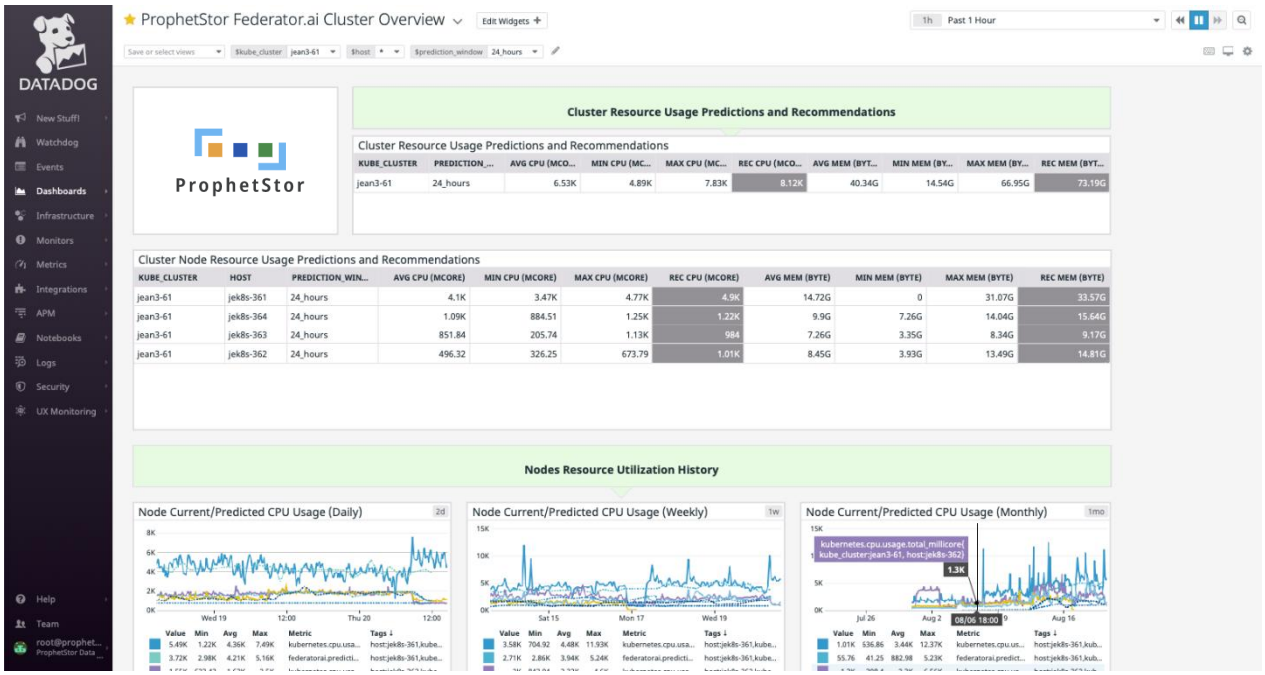


Appendix

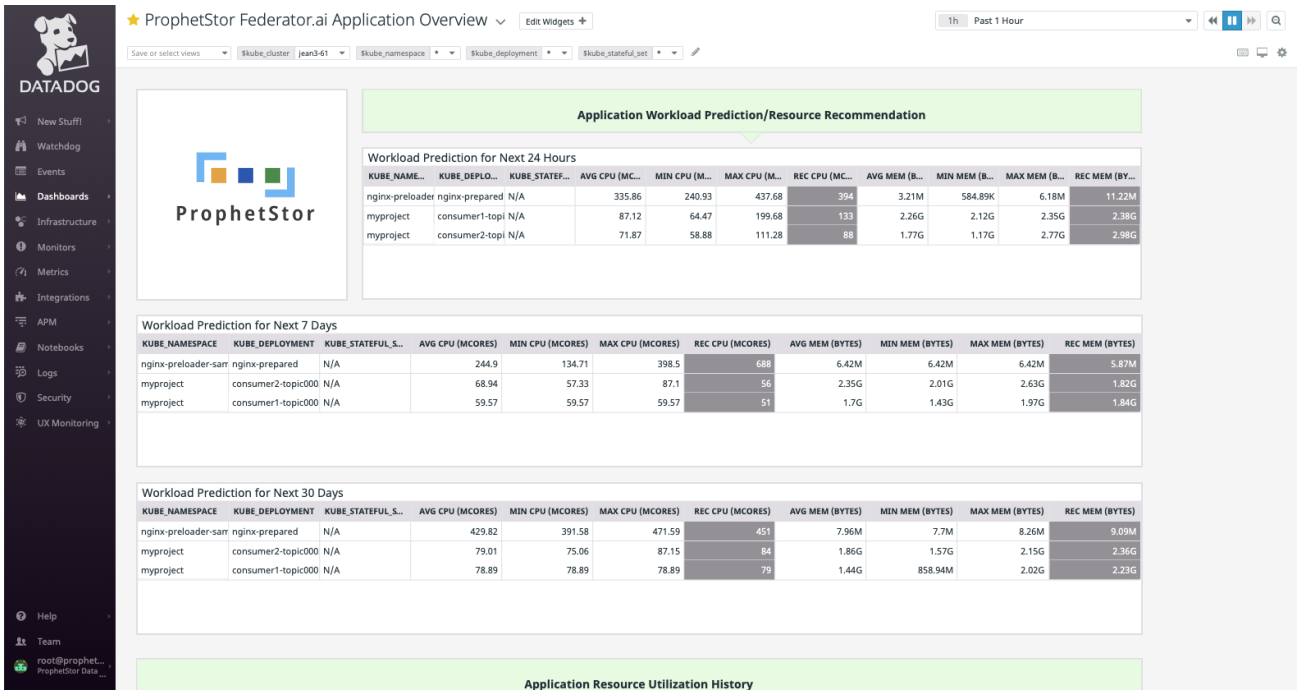
Datadog Dashboards Overview

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

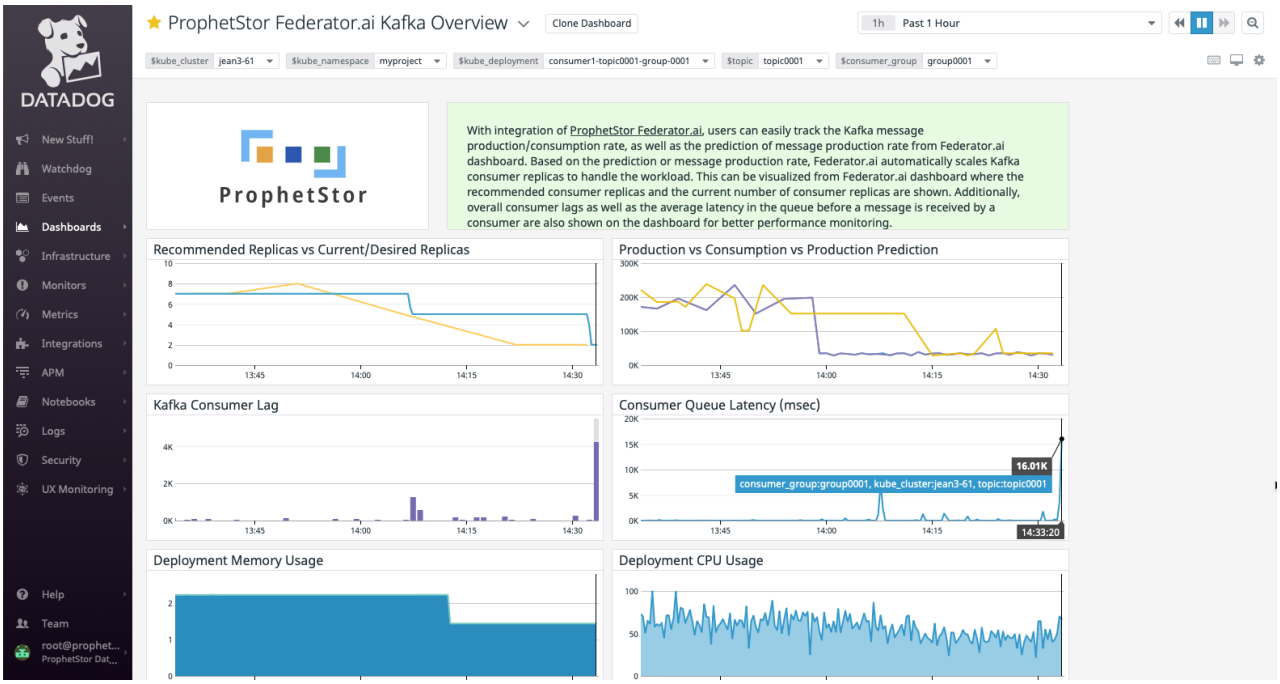
ProphetStor Federator.ai Cluster Overview



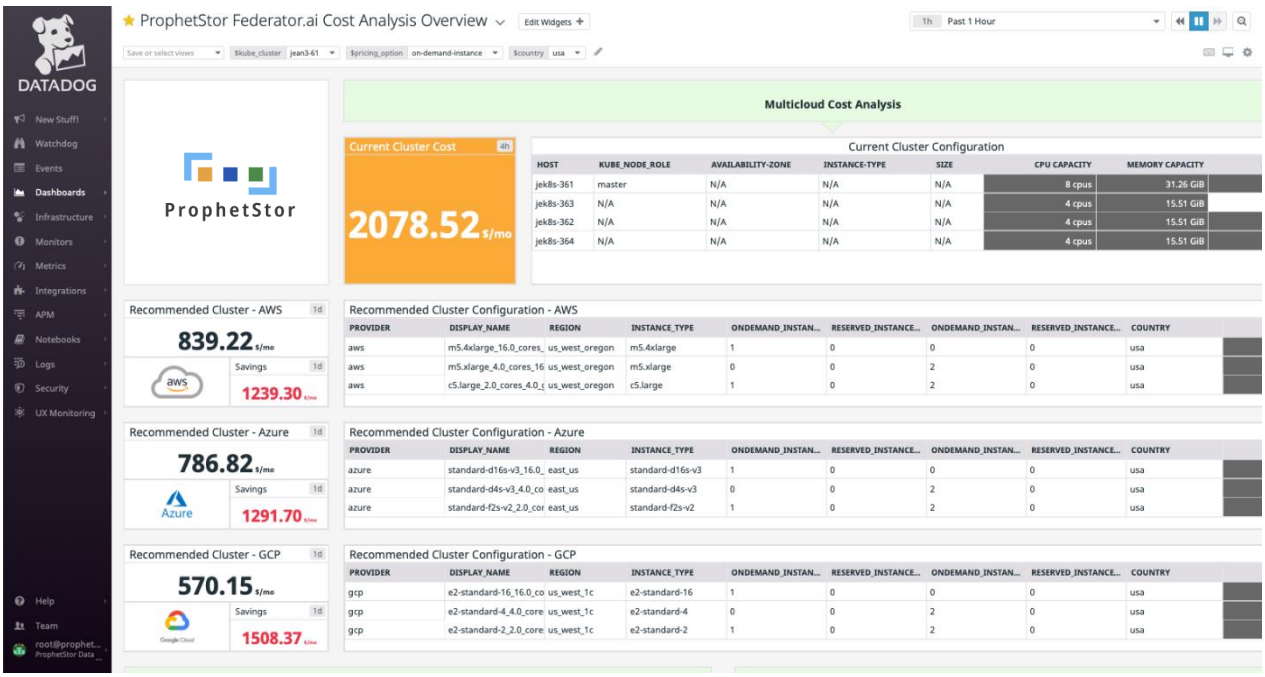
ProphetStor Federator.ai Application Overview



ProphetStor Federator.ai Kafka Overview



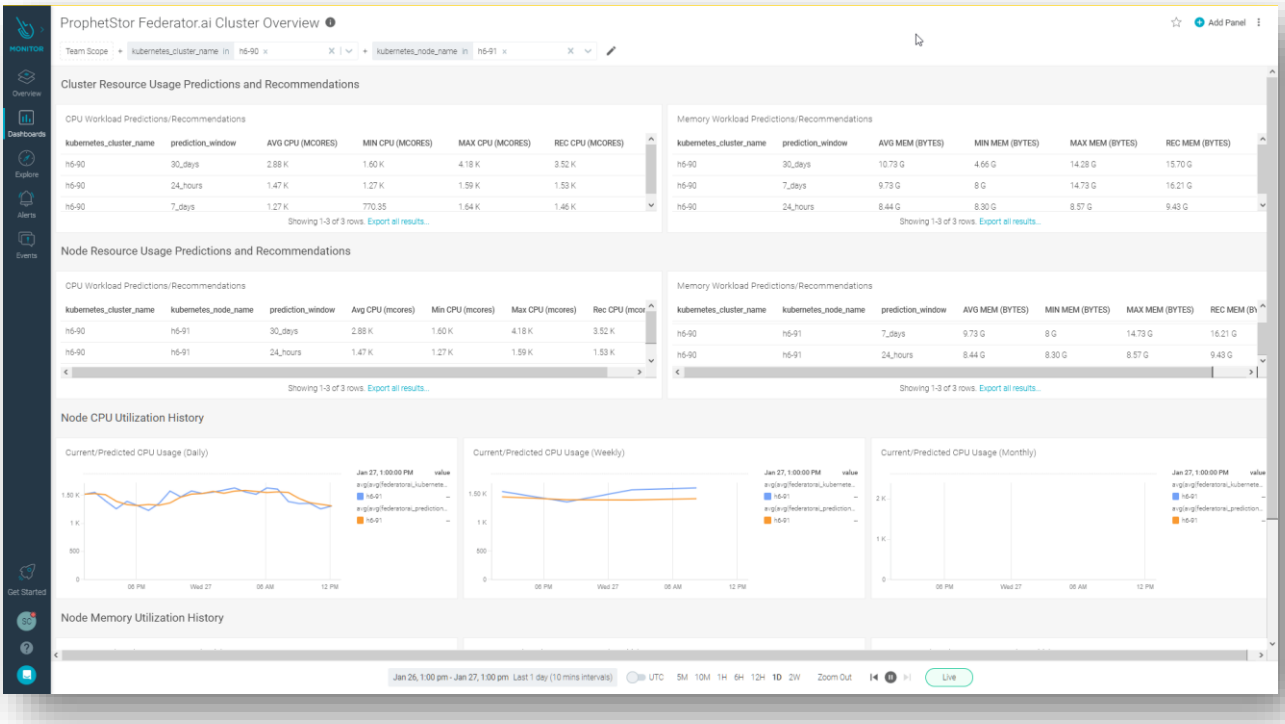
ProphetStor Federator.ai Cost Analysis Overview



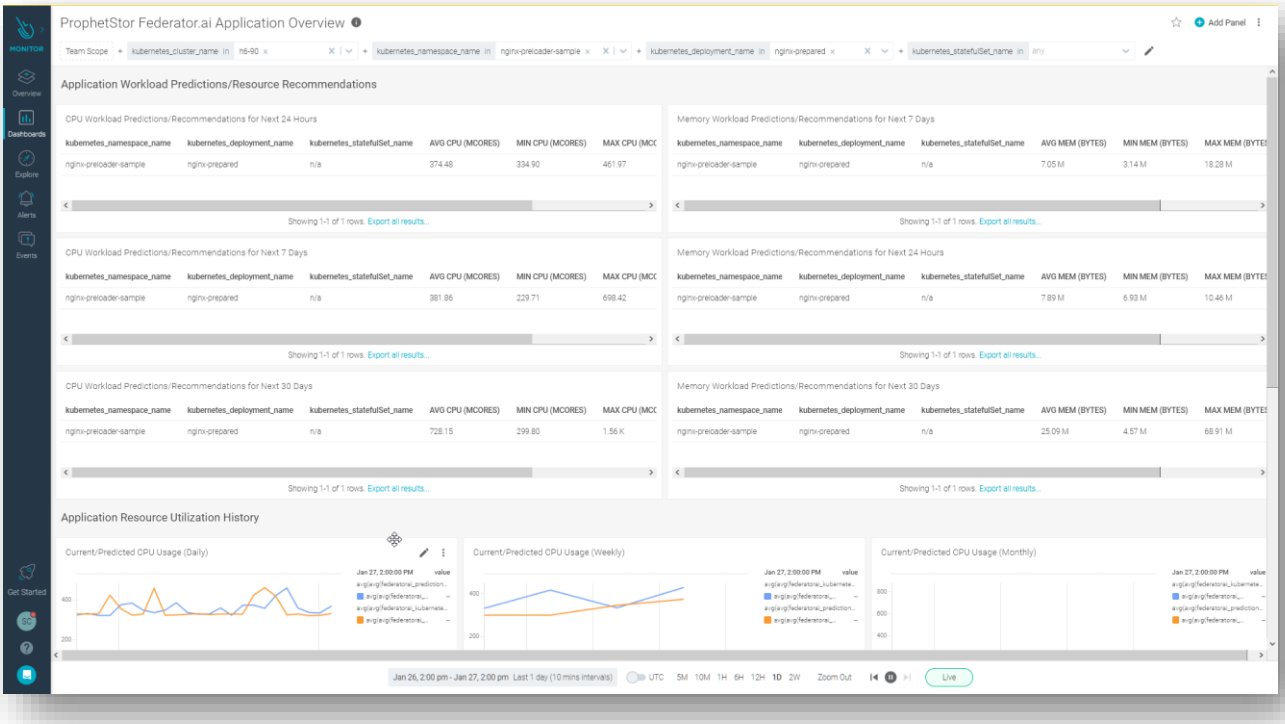
Sysdig Dashboard Overview

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

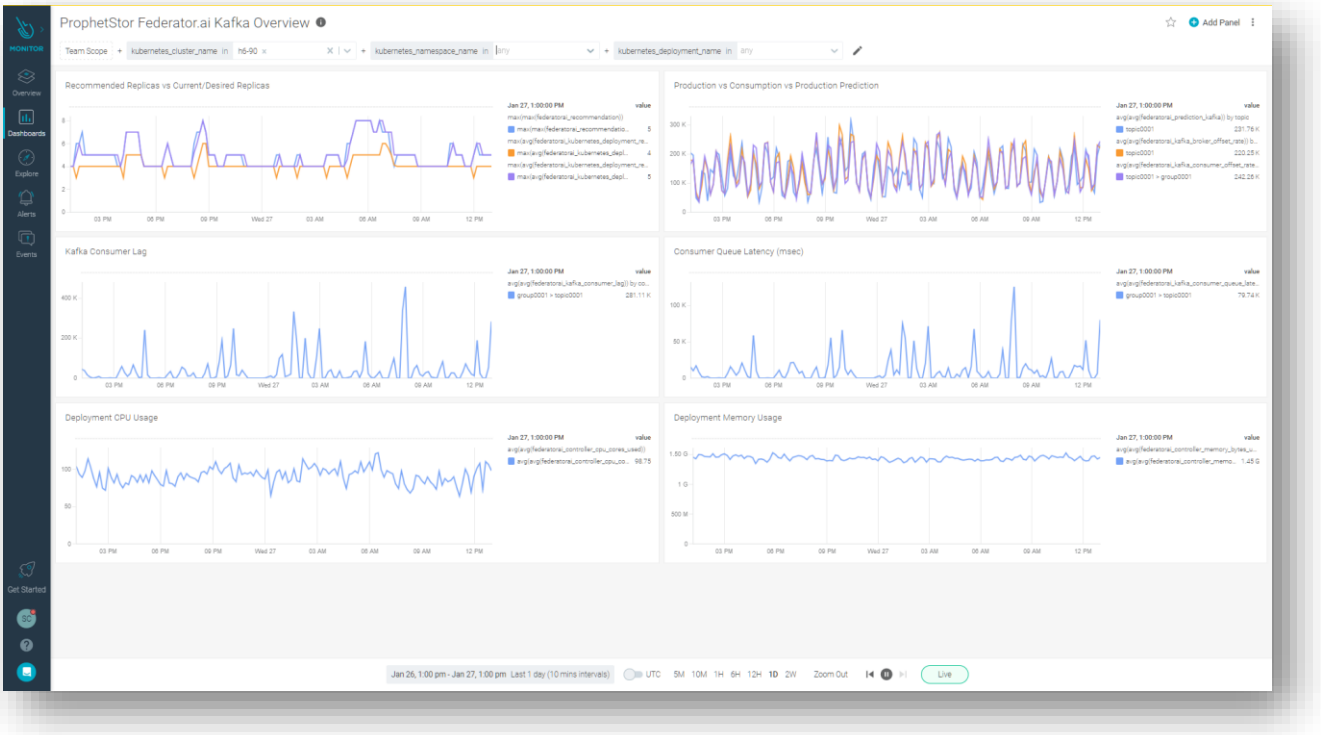
Federator.ai Cluster Overview



Federator.ai Application Overview



Federator.ai Application Overview



Federator.ai installation/uninstallation using Helm Chart

Prerequisites

- Kubernetes version 1.18 or later
- OpenShift version 4.x.x or later
- Helm version is 3.x.x or later

Add Helm chart repository

```
~# helm repo add prophetstor https://prophetstor-ai.github.io/federatorai-operator-helm/
```

Test the Helm chart repository

```
~# helm search repo federatorai
```

Installing with the release name my-name:

```
~# helm install `my-name` prophetstor/federatorai --namespace=federatorai --create-namespace
```

To uninstall/delete the my-name deployment:

```
~# helm ls --all-namespaces
helm delete `my-name` --namespace=federatorai
```

Configuration

The following table lists the configurable parameters of the chart and their default values are specified in values.yaml.

Parameter	Description
`image.pullPolicy`	Container pull policy
`image.repository`	Image for Federator.ai operator
`image.tag`	Image Tag for Federator.ai operator
`federatorai.imageLocation`	Image Location for services containers
`federatorai.version`	Image Tag for services containers
`federatorai.persistence.enabled`	Enable persistent volumes
`federatorai.persistence.storageClass`	Storage Class Name of persistent volumes
`federatorai.persistence.storages.logStorage.size`	Log volume size
`federatorai.persistence.aiCore.dataStorage.size`	AICore data volume size
`federatorai.persistence.influxdb.dataStorage.size`	Influxdb data volume size
`federatorai.persistence.fedemeterInfluxdb.dataStorage.size`	Fedemeter influxdb data volume size
`services.dashboardFrontend.nodePort`	Port of the Dashboard service

Specify each parameter using the --set key=value[,key=value] argument to helm install

Tip: You can use the default values.yaml

Sample :valume.yaml

```
## Default values for Federator.ai
## This is a YAML-formatted file.
## Declare variables to be passed into your templates.
##
image:
  pullPolicy: IfNotPresent
  repository: quay.io/prophetstor/federatorai-operator-ubi
  tag: v5.0.0-ga

## Set default values
##
federatorai:
  imageLocation: quay.io/prophetstor
  version: v5.0.0-ga
```

```
  ## If the persistence is enabled, a default StorageClass
  ## is needed in the k8s cluster to provision volumes.
  persistence:
    enabled: true
    storageClass: "standard"
    storages:
      logStorage:
        size: 2Gi
    aiCore:
      dataStorage:
        size: 10Gi
    influxdb:
      dataStorage:
        size: 100Gi
    fedemeterInfluxdb:
      dataStorage:
        size: 10Gi

services:
  dashboardFrontend:
    ## Specify the nodePort value for the dashboard frontend
    ## Comment out the following line to disable nodePort service
    nodePort: 31012
  rest:
    ## Specify the nodePort value for the REST service
    ## Comment out the following line to disable nodePort service
    nodePort: 31011
```

Alternative installation with configuration file

A YAML file that specifies the values for the parameters can be provided while installing the chart. For example

```
~# helm install `my-name` prophetstor/federatorai -f values.yaml --
namespace=federatorai --create-namespace
```

Federator.ai installation/uninstallation using Ansible

Only support Federator.ai since v4.4.0 or later

Prerequisite

Ansible Control Node

Software:	Version:	Query Command:
Ansible	2.10.2 or later	ansible --version
Ansible Collection - community.kubernetes	1.1.1 or later	ansible-galaxy collection list or ansible-galaxy collection install community.kubernetes -vvv
Python	3.7 or later	python3 --version

OpenShift python client (Required by community.kubernetes collection)	0.11.2 or later	pip3 list | grep openshift
kubeconfig file (Need to copy target cluster's kubeconfig file to the Ansible Control Node)		e.g. file is put on /root/.kube/config.135

Preparation (Ansible Control Node):

1. Install Ansible
   https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html
2. Install collection “community.kubernetes“

```
~# ansible-galaxy collection install community.kubernetes
```

3. Install python & pip
   https://www.python.org/downloads/
4. Install OpenShift python client if you are using OpenShift clusters

```
~# pip3 install openshift
```

5. Download Ansible playbook for Federator.ai
6. Modify user_variable.yaml file for customizing needed info.

Installing Federator.ai

Variables for in user_variable.yml

Group	Variable Name	Sample value	Description	Mandatory
Federator.ai env	federatorai_version	v5.0.0-ga	Federator.ai version tag	Y
Storage for Federator.ai pods	storage_type	ephemeral or persistent	Using ephemeral persistent volume type	Y
Storage info (Only be used when storage_type is persistent)	log_storage_size	10	Log size reserved for every pod. 10 means 10GB	N
Private repo	enable_private_repo	y	Using private repo to pull the Federator.ai required docker images	N
Pod resource	enable_resource_requirement	y	Add pod resource requirement (limits & requests) for every Federator.ai pod	N

Expose services (Only be used when openshift_env is "n ")	expose_dashboard_and_rest_api_services	y	Expose the dashboard and API services in the Kubernetes cluster.	Y
Cluster type	openshift_env	n	Input "y "if installed cluster is OpenShift cluster	Y
	installed_namespace	federatorai	namespace where Federator.ai will be installed	N
	image_url_prefix	""	Input the private repo URL	N
	ai_engine_size	10	Storage size reserved for Alameda AI engine.	N
	influxdb_storage_size	100	Data size reserved for InfluxDB pod.	N
	storage_class_name	scname	To specifying storage class name for provisioning persistent volumes	Y

Steps:

1. Go to Ansible playbook folder

```
~# cd ansible_for_federatorai
```

2. Modify user_variable.yaml (under uninstaller folder) file for customizing needed info.
3. Export K8S_AUTH_KUBECONFIG to specify kubeconfig file path for Ansible collection (community.kubernetes).

```
~# export K8S_AUTH_KUBECONFIG=/root/.kube/config.135
```

4. Run Ansible playbook

```
~# ansible-playbook federtorai_installation.yaml
```

Uninstalling Federator.ai

For Uninstallation, please use the file under ansible_for_federatorai/uninstaller directory.

Variables in user_variable.yml.

Group	Variable Name	Sample value	Description	Mandatory
Storage for Federator.ai pods	storage_type	ephemeral or persistent	Specify current Federator.ai storage type (ephemeral or persistent)	Y
Preserve current persistent volume (Only be used when storage_type is persistent)	preserve_pv	Y	Specify whether to preserve Federator.ai PVs	

Steps:

1. Get to Ansible playbook uninstallation folder

```
~# cd ansible_for_federatorai/uninstaller
```

2. Modify user_variable.yaml (under uninstaller folder) file for customizing needed info.
3. Export **K8S_AUTH_KUBECONFIG** to specify kubeconfig file path for Ansible collection (community.kubernetes)

```
~# export K8S_AUTH_KUBECONFIG=/root/.kube/config.135
```

4. Run Ansible playbook

```
~# ansible-playbook federatorai_uninstaller.yaml
```

Configure nativeHPA using external metrics for Datadog

Sample YAML file

```
#!Sample YAML
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: <app1-hpa>
  namespace: <namespace>
spec:
  minReplicas: <num>
  maxReplicas: <num>
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: <deployment_name>
  metrics:
  - type: External
    external:
      metricName: federatorai.recommendation
      metricSelector:
```

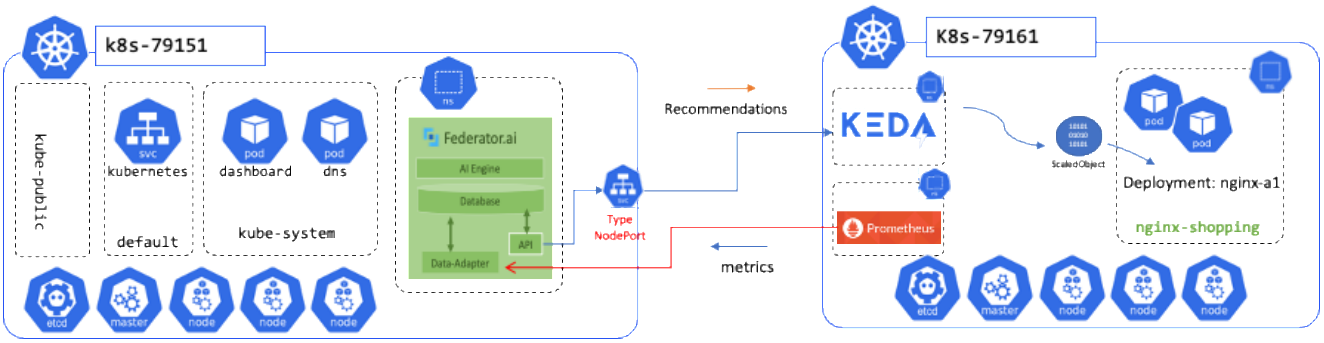
```
      matchLabels:
        resource: replicas
        kube_cluster: <clusterName>
        kube_deployment: <deployment_name>
        kube_namespace: <namespace>
    targetAverageValue: 1
```

Deploy KEDA for Remote and Sysdig monitored Cluster Pod Autoscaling

Step 1: Deploy KEDA.

```
$helm repo add kedacore https://kedacore.github.io/charts
$helm repo update
$kubectl create namespace keda
$helm install keda kedacore/keda --namespace keda
```

Example: Configure Generic Application HPA using KEDA ScaledObject



Environment

- Kind: **deployment**
- Deployment Name: **nginx-a1**
- Namespace: **nginx-shopping**
- Ingress for Federator.ai API URL: “http://172.31.79.151:31011”

Apply updated YAML

This file can be divided into 3 parts.

1. Secrets: storage for access token

Federatorai REST provide “basic” authentication mode, so we need to create “Secret” with encoded base64 string (base64), in the example above, “**YWRtaW4=**” and “**cGFzc3dvcmQ=**” are “**admin**” and “**password**” respectively.

```
$vi secret-shopping.yaml
apiVersion: v1
kind: Secret
metadata:
  name: keda-metric-api-secret
  namespace: nginx-shopping
data:
  username: "YWRtaW4="
```

```
  password: "cGFzc3dvcmQ="
$kubectl apply -f secret-shopping.yaml -n nginx-shopping
```

Get Federator.ai API certificated and patch secret

```
$CACRT=$(echo | openssl s_client -showcerts -connect 172.31.79.151:31011
2>/dev/null | openssl x509 | sed -n -e '/BEGIN\ CERTIFICATE/,/END\ CERTIFICATE/
p' | base64 | tr -d '\n')
```

*” 31011” is Federator.ai API port access from external of cluster

```
$ kubectl -n nginx-shopping patch secret keda-metric-api-secret -p
"{\"data\":{\"ca.crt\":\"${CACRT}\"}}"
```

2. TriggerAuthentication: target to trigger authentication
3. ScaledObject: target to scale and how to trigger it with provided authentication method
4. URL format in ScaledObject configuration YAML:

   "https://<clusterIP>:<federator_api

   _port/apis/v1/recommendations/clusters/<cluster_name>/namespaces/<namespace_name

   >/deployments/<demployment_name>/metrics/<metric_name>?limit=1&order=desc"

metric_name:

- “controller_desired_replicas”
- “kafka_consmer_group_desired_replicas”
- “ingress_upstream_desired_replicas”

	pollingInterval	**cooldownPeriod**
generic application	90	180
kafka consumer group	150	300
ingress upstream	30	60

```
$ vi app-scaledobject.yaml
#! – sample YAML
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: keda-metric-api-creds
  namespace: nginx-shopping
spec:
  secretTargetRef:
    - parameter: username
      name: keda-metric-api-secret
      key: username
    - parameter: password
      name: keda-metric-api-secret
      key: password
    - key: ca.crt
      name: keda-metric-api-secret
```

```
      parameter: ca
---
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: http-scaledobject
  namespace: nginx-shopping
  labels:
    deploymentName: nginx-a1
spec:
  minReplicaCount: 1
  maxReplicaCount: 10
  pollingInterval: 150
  cooldownPeriod: 300
  scaleTargetRef:
    name: nginx-a1   ←deployment name
  triggers:
    - type: metrics-api
      metadata:
        targetValue: “1”
        url: "https://172.31.79.151:31011/apis/v1/recommendations/clusters/k8s-
79161/namespaces/nginx-shopping/deployments/nginx-a1/metrics/
controller_desired_replicas?limit=1&order=desc"
        valueLocation: 'data.0.value'
        authMode: "basic"
      authenticationRef:
        name: keda-metric-api-creds
```

Apply YAML file to create KEDA scaled object

```
$kubectl apply -f app-scaledobject.yaml
$kubectl get ScaledObject -n nginx-shopping
```

NAME	SCALETARGETKIND	SCALETARGETNAME	MIN	MAX	TRIGGERS	AUTHENTICATION	READY	ACTIVE	FALLBACK	AGE
http-scaledobject	apps/v1.Deployment	nginx-a1	1	10	metrics-api	keda-metric-api-creds	True	True	False	25h

Make sure Ready, Active status is “True”. If status is false, describe ScaledObject to check error message.