Federator.ai Release v5.1.0
Installation Guide
Contents

Overview ....................................................................................................................... 3
Federator.ai .................................................................................................................. 3
Supported Metrics Data Sources .................................................................................. 3
  Prometheus .................................................................................................................. 3
  Datadog ...................................................................................................................... 4
  Sysdig .......................................................................................................................... 5
  VMWare vCenter ....................................................................................................... 5
  AWS CloudWatch ...................................................................................................... 6

Requirements and Recommended Resource Configuration .................................... 7
Supported Platform ...................................................................................................... 7
Data Source .................................................................................................................. 7
Federator.ai Resource Requirements .......................................................................... 7
Federator.ai Version ..................................................................................................... 7
Datadog Agent Version (reference) ............................................................................. 8
Prometheus Version (reference) ................................................................................ 8
Sysdig Agent Version (reference) ............................................................................... 8

Federator.ai Installation and Configuration ................................................................ 9
Summary of Installation Steps .................................................................................. 9
Pre-installation Check List ....................................................................................... 9
Before You Start ........................................................................................................ 13
  Datadog .................................................................................................................... 13
  Sysdig ....................................................................................................................... 14
  VMWare vCenter .................................................................................................... 15
  AWS CloudWatch ................................................................................................. 15
Brand New Installation .............................................................................................. 17
Federator.ai internal/external communication ports .............................................. 20

Appendix ..................................................................................................................... 26
Datadog Dashboards Overview ................................................................................ 26
  ProphetStor Federator.ai Cluster Overview ............................................................ 26
  ProphetStor Federator.ai Application Overview ................................................... 26
  ProphetStor Federator.ai Kafka Overview .............................................................. 27
  ProphetStor Federator.ai Cost Analysis Overview ............................................... 27
Sysdig Dashboard Overview ...................................................................................... 28
  Federator.ai Cluster Overview ............................................................................... 28
  Federator.ai Application Overview ....................................................................... 28
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federator.ai Application Overview</td>
<td>29</td>
</tr>
<tr>
<td>Federator.ai installation/uninstallation using Helm Chart</td>
<td>29</td>
</tr>
<tr>
<td>Federator.ai installation/uninstallation using Ansible</td>
<td>31</td>
</tr>
<tr>
<td><strong>Prerequisite</strong></td>
<td>31</td>
</tr>
<tr>
<td><strong>Preparation (Ansible Control Node):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Installing Federator.ai</strong></td>
<td>32</td>
</tr>
<tr>
<td>Configure nativeHPA using external metrics for Datadog</td>
<td>34</td>
</tr>
<tr>
<td>Deploy KEDA for Remote and Sysdig monitored Cluster Pod Autoscaling</td>
<td>35</td>
</tr>
<tr>
<td>Example: Configure Generic Application HPA using KEDA ScaledObject</td>
<td>35</td>
</tr>
</tbody>
</table>
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 v5.1.0-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

Federator.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 Metrics Data Sources

- Prometheus
- Datadog
- Sysdig
- VMWare vCenter
- AWS CloudWatch

Supported Platforms

- Kubernetes v1.11.x – 1.22x
- Red Hat OpenShift v4.6-4.9
- Amazon AWS/EKS
- Google GCP/GKE
- Microsoft Azure/AKS
- Rancher v2.4.8, 2.5.8, 2.5.9, 2.6.3
- VMware vCenter 5.5, 6.0, 6.5, 6.7, 7.0

Federator.ai Resource Requirements

- Total Resource Requirements
  - 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.

Permission Request:

- “Federator-operator” required Kubernetes Cluster Access Management RBAC privilege with Cluster-Full Control (Admin) permission (Scope-Operation).
- Installation scripts do NOT require root permissions to run in the host where “kubecti” command is set up.
- All of Federator.ai pods are running as “non-root” user in cluster.
Federator.ai Version

- Version: Release 5.1.0-ga
- Tag: v5.1.0-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>.<namespace>:9090)
- For Sysdig, obtain Sysdig API URL and Token.
- For VMware vCenter, obtain administrator 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

<table>
<thead>
<tr>
<th>Federator.ai Feature</th>
<th>Component</th>
<th>Scope</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation, Lifecycle</td>
<td>federatorai-operator</td>
<td>Cluster</td>
<td>admin</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Collection/Controller</td>
<td>alameda-datahub</td>
<td>Cluster</td>
<td>edit</td>
</tr>
<tr>
<td></td>
<td>alameda-operator</td>
<td>Cluster</td>
<td>edit</td>
</tr>
<tr>
<td></td>
<td>federatorai-data-adapter</td>
<td>Cluster</td>
<td>view</td>
</tr>
<tr>
<td>Dashboard</td>
<td>federatorai-dashboard-frontend</td>
<td>Local</td>
<td>view</td>
</tr>
<tr>
<td></td>
<td>federatorai-dashboard-backend</td>
<td>Cluster</td>
<td>view</td>
</tr>
</tbody>
</table>
### 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 = v5.1.0-ga
Please enter the path of Federator.ai directory [default: /opt]:

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

### Kubernetes:

<table>
<thead>
<tr>
<th>#</th>
<th>Checklist Item</th>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
</table>
| 1 | What is the Kubernetes version? | Server Version: 1.11~1.22 | Use the command below to get the Kubernetes version:
$ kubectl version ...
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 v5.1.0 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:

<table>
<thead>
<tr>
<th>Checklist Item</th>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
</table>
| #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-operator monitoring 2020-03-13 15:35:05.28963154 +0800 CST deployed prometheus-operator-2.1.3 v0.6.0

prometheus-operator monitoring 2020-03-13 14:34:16.132479221 +0800 CST deployed prometheus-operator-8.12.1 0.37.0 |
Datadog Agent:

<table>
<thead>
<tr>
<th>#</th>
<th>Checklist Item</th>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Is Datadog Agent installed?</td>
<td>Datadog Agent is mandatory</td>
<td>Kubernetes resources and workload metrics are collected by Datadog Agent.</td>
</tr>
<tr>
<td>2</td>
<td>Is Datadog Cluster Agent installed?</td>
<td>Cluster Agent is mandatory for the HPA autoscaling feature</td>
<td>Cluster Agent provides metrics to HPA Autoscaler for autoscaling.</td>
</tr>
<tr>
<td>3</td>
<td>Is Datadog WPA controller installed? (Option)</td>
<td>Datadog WPA is required if autoscaling is done by WPA</td>
<td>Datadog WPA is the HPA Autoscaler developed by Datadog. Users can use Datadog WPA or Kubernetes native HPA to do autoscaling.</td>
</tr>
<tr>
<td>4</td>
<td>Datadog Kafka Consumer integration is enabled? (Option)</td>
<td>Datadog Kafka Consumer integration is mandatory if user wants to use Kafka consumer optimization feature</td>
<td>Use the command to confirm Kafka integration is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$ kubectl exec &lt;datadog-agent-pod&gt; -n &lt;datadog-agent-namespace&gt; -- agent integration show datadog-kafka-consumer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Refer to <a href="https://www.datadoghq.com/blog/monitor-kafka-with-datadog/">https://www.datadoghq.com/blog/monitor-kafka-with-datadog/</a> for Kafka Consumer integration installation</td>
</tr>
<tr>
<td>5</td>
<td>Datadog account API key</td>
<td>An API key is mandatory for connecting Datadog Service</td>
<td>Follow the steps described in the &quot;Before You Start&quot; session to obtain the API key.</td>
</tr>
<tr>
<td>6</td>
<td>Datadog account Application key</td>
<td>An application key is mandatory for connecting Datadog Service</td>
<td>Follow the steps described in the &quot;Before You Start&quot; session to obtain the Application key.</td>
</tr>
<tr>
<td>7</td>
<td>Is one of cluster name is configured for the Datadog agent/cluster agent?</td>
<td>&quot;kube_cluster&quot;,&quot;cluster_name&quot;,&quot;kube_cluster_name(DD_CLUSTER_NAME)&quot; one of them is required for Federator.ai to identify Kubernetes cluster.</td>
<td>Case 1. &gt; New Datadog Agent installation: Install Datadog agent and cluster agent by &quot;helm install -f values.yaml&quot;, in values.yaml.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>... clusterName: &lt;cluster-name&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>... clusterAgent: enabled: false true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Case 2. &gt; In Datadog Agent installed environment, with no Cluster Agent and no cluster_name setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Update Datadog Agent to enable Cluster agent by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;helm upgrade -f values.yaml&quot;, in values.yaml</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- assign a cluster name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>... datadog:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>... clusterName: &lt;cluster-name&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- enable cluster agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>... clusterAgent: enabled: false true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- $helm upgrade ...</td>
</tr>
</tbody>
</table>
Before You Start

**Datadog**

- The admin role for installing Federator.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/](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
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](https://app.sysdigcloud.com)
  - For US West, Sysdig API URL is [https://us2.app.sysdig.com](https://us2.app.sysdig.com)
  - For European Union, Sysdig API URL is [https://eu1.app.sysdig.com](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

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

   Please enter Federator.ai version tag [default: latest]:
   Federator.ai version = v5.1.0-ga

Please enter the path of Federator.ai directory [default: /opt]:
   Downloading v5.1.0-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 = v5.1.0-ga
   install_namespace = federatorai
   ----------------------------------------
   Is the above information correct? [default: y]:

   Downloading v5.1.0-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 v5.1.0-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(v5.1.0-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-d7c78595f-hj82d 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 Federator.ai 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 v5.1.0-ga successfully
Downloaded YAML files are located under /opt/federatorai/installation
Downloaded files are located under /opt/federatorai/repo/v5.1.0-ga

3. Verify Federator.ai pods are running properly

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>alameda-ai-d7c78595f-hj82d</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-ai-dispatcher-68d448f99c-chhsz</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>alamedadatahub-6997fcb8-fzjlz</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>alameda-executor-57958db45-8zm9m</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-influxdb-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-notifier-c95bbf686-plfk7</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>alamedarabbitmq-76dfd4b77c-wstvh</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>fedemeter-api-5878cfdd7-ng6dg</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>fedemeter-influxdb-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>federatorai-agent-845f86c4df-r4ck5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>federatorai-alert-detector-56c694fd9c-z7fnv</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>3m59s</td>
</tr>
<tr>
<td>federatorai-alert-manager-6c845744-qn6gm</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-dashboard-backend-9794b4b9d-kpcw</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-dashboard-frontend-68999c8958-752cl</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>federatorai-data-adapter-7688d589cb-qksb4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-operator-668c45f7d9-6ghbg</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>6m26s</td>
</tr>
<tr>
<td>federatorai-postgresql-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m19s</td>
</tr>
<tr>
<td>federatorai-recommender-dispatcher-dd455cd79-g8xmm</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-recommender-worker-9564c54d5-dnqzv</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>3m53s</td>
</tr>
<tr>
<td>federatorai-rest-7f9656775c-plf2r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
</tbody>
</table>
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"

```bash
~# 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

```bash
~# 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.

```bash
$ kubectl get svc -n federatorai | grep federatorai-dashboard-frontend-node-port

The output will look something like this:

```
[root@sandbox2a ~]# kubectl get svc -n federatorai | grep federatorai-dashboard-frontend-node-port
federatorai-dashboard-frontend-node-port  NodePort  10.43.221.238 <none> 9001:31012/TCP 9m1s
```
Get the node's IP to access (INTERNAL-IP).

```
$[root@k8s-c1-n1 ~]# kubectl get node -o wide
NAME        STATUS   ROLES              AGE     VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME
k8s-c1-n1   Ready    controlplane,etcd,worker 294d   v1.18.20 172.31.79.151 <none>        CentOS Linux 7 (Core) 3.10.0-957.el7.x86_64   docker://20.10.12
172.31.79.151 <none> worker                     294d   v1.18.20 172.31.79.152 <none>        CentOS Linux 7 (Core) 3.10.0-957.el7.x86_64   docker://18.9.9
k8s-c1-n2   Ready    worker                     294d   v1.18.20 172.31.79.153 <none>        CentOS Linux 7 (Core) 3.10.0-957.el7.x86_64   docker://18.9.9
172.31.79.152 <none> worker                     294d   v1.18.20 172.31.79.153 <none>        CentOS Linux 7 (Core) 3.10.0-957.el7.x86_64   docker://18.9.9
``` 

(example) The URL will be `https://172.31.79.151-153:31012`

- **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
federatorai-dashboard-frontend frontend-http edge/Redirect None
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
Upgrade from a previous version

Federator.ai v5.1.0 supports upgrade from v5.0.0 versions. The Federator.ai installation script automatically detects previously installed Federator.ai. When the installation script prompts if a backup of the previous configuration is needed, just enter yes to save a copy of the configuration if roll back to the previous version is needed.

Prerequisite

1. Federator.ai version is 5.0.0
2. Federator.ai installed and running with Persistent Volume

Upgrade

1. Log into Kubernetes cluster
2. Install the Federator.ai for Kubernetes by using the following command

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

   ```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 16783  100 16783    0     0  25155      0      --:--:-- --:--:-- --:--:-- 25161
   Please enter Federator.ai version tag [default: latest]:
   Federator.ai version = v5.1.0-ga
   Please enter the path of Federator.ai directory [default: /opt]:
   
   Downloading v5.1.0-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
   Previous build with tag v5.0.0-p2 detected in namespace federatorai
   
   ----------------------------------------
   Upgrade:
   tag_number = v5.1.0-ga
   install_namespace = federatorai
   ----------------------------------------
   Is the above information correct? [default: y]:
   Please enter the path for storing backup configuration: [default: /opt/federatorai/configuration_backup]
   Please enter the dashboard login account: admin
   Please enter the dashboard login password:
   Backup configuration...
   backup file saved to folder /opt/federatorai/configuration_backup/federatorai-v5.0.0-p2-backup-20220512-140419.tgz
   
   Downloading v5.1.0-ga tgz file ...
   ```
Done

Applying Federator.ai operator yaml files...
deployment.apps "federatorai-operator" deleted
Applying 00-namespace.yaml...
namespace/federatorai unchanged
Applying 01-serviceaccount.yaml...
serviceaccount/federatorai-operator unchanged

Waiting for federatorai-operator(v5.1.0-ga) pod to appear ...

federatorai-operator pod is present.

federatorai-operator pod is ready.

Downloading Federator.ai alamedaservice sample file ...
Done

Update alamedaservice...
alamedaservice.federatorai.containers.ai/my-alamedaservice patched
Done.
Processing...
Waiting for datahub(v5.1.0-ga) pod to appear ...
Waiting for datahub(v5.1.0-ga) pod to appear ...

datahub pod is present.

Checking pods...
All pods under namespace(federatorai) are ready.
The default alamedaoorganization under namespace federatorai is ready.

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

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 v5.1.0-ga successfully

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

Downloaded files are located under /opt/federatorai/repo/v5.1.0-ga

3. Verify Federator.ai pods are running properly
```bash
~# kubectl get pod -n federatorai

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>alameda-ai-d7c78595f-hj82d</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-ai-dispatcher-68d448f99c-chhsz</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-datahub-6997fcb8-fzjlz</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>alameda-executor-57958dc845-8zm9m</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-influxdb-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>alameda-notifier-c95bbf686-plfk7</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>alameda-rabbitmq-76dfdb77c-wstvh</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>federatorai-api-5878cfd27-ng6dg</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>federatorai-influxdb-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m20s</td>
</tr>
<tr>
<td>federatorai-agent-845f86c4df-r4ck5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>federatorai-alert-detector-56c694fd9c-27fnav</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>3m59s</td>
</tr>
<tr>
<td>federatorai-alert-manager-6c845744-qn6gm</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-dashboard-backend-c974b4b9d-kpbcw</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-dashboard-frontend-68999c8958-752cl</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
<tr>
<td>federatorai-data-adapter-7688d589cb-qksb4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-operator-668c45f7d9-6ghbg</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>6m26s</td>
</tr>
<tr>
<td>federatorai-postgresql-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m19s</td>
</tr>
<tr>
<td>federatorai-recommender-dispatcher-dd455cd79-g8xmm</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m</td>
</tr>
<tr>
<td>federatorai-recommender-worker-9564c54d5-dnqzv</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>3m53s</td>
</tr>
<tr>
<td>federatorai-rest-7f9656775c-plf2r</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>4m1s</td>
</tr>
</tbody>
</table>
Appendix

Datadog Dashboards Overview

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

ProphetStor Federator.ai Cost Analysis Overview

With integration of ProphetStor Federator.ai, users can easily track the Kafka message production and consumption rate, as well as the prediction of message production rate from the dashboard. Based on the prediction of message production rate, Federator.ai automatically scales Kafka consumer replicas to handle the workload. This can be visualized from the dashboard where the recommended consumer replicas and the current number of consumer replicas are shown. Additionally, overall consumer lag as well as the average latency in the queue before a message is processed by a consumer are also shown on the dashboard for better performance monitoring.

Federator.ai v5.1.0 Installation Guide
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`.

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

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

Tip: You can use the default values.yaml

Sample: `values.yaml`

```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: v5.1.0-ga

## Set default values
##
federatorai:
imageLocation: quay.io/prophetstor
version: v5.1.0-ga
```
## If the persistence is enabled, a default StorageClass is needed in the k8s cluster to provision volumes.

```yaml
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

```bash
# 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**

<table>
<thead>
<tr>
<th>Software</th>
<th>Version:</th>
<th>Query Command:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible</td>
<td>2.10.2 or later</td>
<td>ansible --version</td>
</tr>
<tr>
<td>Ansible Collection - community.kubernetes</td>
<td>1.1.1 or later</td>
<td>ansible-galaxy collection list or ansible-galaxy collection install community.kubernetes -vvv</td>
</tr>
<tr>
<td>Python</td>
<td>3.7 or later</td>
<td>python3 --version</td>
</tr>
</tbody>
</table>
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

### Preparation (Ansible Control Node):

1. Install Ansible
   
   [https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html](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/](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

<table>
<thead>
<tr>
<th>Group</th>
<th>Variable Name</th>
<th>Sample value</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Federator.ai env</strong></td>
<td>federatorai_version</td>
<td>v5.1.0-ga</td>
<td>Federator.ai version tag</td>
<td>Y</td>
</tr>
<tr>
<td><strong>Storage for Federator.ai pods</strong></td>
<td>storage_type</td>
<td>ephemeral or persistent</td>
<td>Using ephemeral persistent volume type</td>
<td>Y</td>
</tr>
<tr>
<td><strong>Storage info</strong></td>
<td>log_storage_size</td>
<td>10</td>
<td>Log size reserved for every pod. 10 means 10GB</td>
<td>N</td>
</tr>
<tr>
<td><strong>Private repo</strong></td>
<td>enable_private_repo</td>
<td>y</td>
<td>Using private repo to pull the Federator.ai required docker images</td>
<td>N</td>
</tr>
<tr>
<td><strong>Pod resource</strong></td>
<td>enable_resource_requirement</td>
<td>y</td>
<td>Add pod resource requirement (limits &amp; requests) for every Federator.ai pod</td>
<td>N</td>
</tr>
</tbody>
</table>
Exposé des services
(Seulement utilisé lorsqu'openshift_env est "n")

<table>
<thead>
<tr>
<th>Expose_dashboard_and_rest_api_services</th>
<th>expose_dashboard_and_rest_api_services</th>
<th>y</th>
<th>Expose the dashboard and API services in the Kubernetes cluster.</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster type</td>
<td>openshift_env</td>
<td>n</td>
<td>Input &quot;y&quot; if installed cluster is OpenShift cluster</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>installed_namespace</td>
<td>federatorai</td>
<td>namespace where Federator.ai will be installed</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>image_url_prefix</td>
<td>&quot;&quot;</td>
<td>Input the private repo URL</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>ai_engine_size</td>
<td>10</td>
<td>Storage size reserved for Alameda AI engine.</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>influxdb_storage_size</td>
<td>100</td>
<td>Data size reserved for InfluxDB pod.</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>storage_class_name</td>
<td>scname</td>
<td>To specifying storage class name for provisioning persistent volumes</td>
<td>Y</td>
</tr>
</tbody>
</table>

**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 federatorai_installation.yaml
   ```

**Uninstalling Federator.ai**

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

**Variables in user_variable.yml.**
Federator.ai v5.0 Installation Guide

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

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

```yaml
#!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:
```
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
```

Example: Configure Generic Application HPA using KEDA ScaledObject

Environment
- Kind: deployment
- Deployment Name: nginx-a1
- Namespace: nginx-shopping

Apply updated YAML
This file can be divided into 3 parts.
1. Secrets: storage for access token

Federator.ai 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

$ CACERT=$(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 '
')

*”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\":\"${CACERT}\"}"

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

metric_name:
- “controller_desired_replicas”
- “kafka_consumer_group_desired_replicas”
- “ingress_upstream_desired_replicas”

<table>
<thead>
<tr>
<th>pollingInterval</th>
<th>cooldownPeriod</th>
</tr>
</thead>
<tbody>
<tr>
<td>generic application</td>
<td>90</td>
</tr>
<tr>
<td>kafka consumer group</td>
<td>150</td>
</tr>
<tr>
<td>ingress upstream</td>
<td>30</td>
</tr>
</tbody>
</table>

$ 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
---

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
  triggers:
    - type: metrics-api
      metadata:
        targetValue: "1"
        valueLocation: 'data.0.value'
        authMode: "basic"
        authenticationRef:
          name: keda-metric-apicreds

Apply YAML file to create KEDA scaled object

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>SCALETARGET_KIND</th>
<th>SCALETARGET_NAME</th>
<th>MIN</th>
<th>MAX</th>
<th>TRIGGERS</th>
<th>AUTHENTICATION</th>
<th>READY</th>
<th>ACTIVE</th>
<th>FALLBACK</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>http-scaledobject</td>
<td>apps/v1.Deployment</td>
<td>nginx-a1</td>
<td>1</td>
<td>10</td>
<td>metrics-api</td>
<td>keda-metric-apicreds</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>25h</td>
</tr>
</tbody>
</table>

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