

Federator.ai v5.1.5 Installation Guide by Using Helm Chart

Contents

- Overview 2**
 - Federator.ai 2
 - Supported Platforms 2
 - Supported Metrics Data Sources 2
 - Resource Requirements 2
- Before You Start 3**
 - Ensure the environment and required resources are available 3
 - Obtain the credentials required to connect to your metrics data source 3
 - Confirm the metrics source agent is installed 4
- Installation 4**
 - Prerequisites 4
 - Installation Steps 4
 - Uninstallation 8
- Offline Installation 8**
 - Workflow 8
 - Bootstrap Host 8
 - Installation Steps 9
- Parameter (optional) 12**

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, AWS EC2, Google GCE, or Azure VM 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, AWS EC2, Google GCE, or Azure VM 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
- Statistical analysis and predictions based on the correlation between resource usage and application workload

Supported Platforms

- OpenShift: 4.6~4.9
- Kubernetes (K8S): 1.16 ~ 1.28
- Rancher v2.4.8, v2.5.8, v2.5.9
- Cloud Environment AKS/EKS/GKE
- VMware Tanzu

Supported Metrics Data Sources

- Prometheus
- Datadog
- Sysdig
- VMWare vCenter
- AWS CloudWatch
- Google Cloud Operations Suite
- Azure Monitor

Resource Requirements

- Total resource requirements:

- CPU : 9 cores
- Memory : 16 GB Memory
- Storage : 156 GB (ReadWriteOnce access mode)
- Resource requirements for the time-series database:
 - It requires to have at least one worker node with more than 3 CPU cores and 8 GB of memory available.
- Persistent volumes:
 - It is strongly 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.

Before You Start

Ensure the environment and required resources are available

- Total resource requirements:
 - CPU : 9 cores
 - Memory : 16 GB Memory
 - Storage : 156 GB
- ❗ *If your environment for testing does not have enough CPU/memory resources to install Federator.ai, you can disable resource requests for Federator.ai by setting the installation parameter '--set global.resourcesRequestsEnabled=false'.*

Obtain the credentials required to connect to your metrics data source

- For Datadog, the API Key and Application Key of your Datadog Cloud Service account are required. Refer to [Datadog documentation](#) for information about how to obtain the API Key and Application Key.
- For Prometheus (Kubernetes, Rancher), obtain the Prometheus service URL (e.g., `http://<prometheus_svc_name>.<namespace>:9090`).

```
# Kubernetes Environment
$ kubectl get svc -n monitoring
```

- For Prometheus (OpenShift), obtain the Prometheus route URL and Token.

```
# OpenShift Environment
$ oc get route -n openshift-monitoring
$ oc get-token Prometheus-k8s -n openshift-monitoring
```

- For Sysdig, the Sysdig API URL and API Token are required. Refer to [Sysdig SaaS Regions and IP Range](#) and [Sysdig API Token](#) for information about how to obtain the Sysdig API URL and API Token.

- For VMware vCenter, obtain administrator login credentials and vCenter IP or FQDN.
- For AWS CloudWatch, the Access Key ID/Secret Access Key or IAM role ARN of your AWS account are required. Refer to [AWS documentation](#) for information about how to obtain the Access Key ID/Secret Access Key or IAM role ARN. CloudWatch Agent installation on each EC2 instance is required.
- For Google Cloud Operations Suite, the service account key file is required. Refer to [Google Cloud Service Account Setup Guide](#) for more information.

Confirm the metrics source agent is installed

- For Prometheus, install Prometheus using helm charts. (OpenShift is enabled by default.)
- For Datadog, install and configure the Datadog Agent/Cluster Agent if not previously installed. Refer to the Datadog documentation for information about how to install the Datadog Agent and Cluster Agent.
- For Sysdig, install and configure the Sysdig Agent. Refer to the Sysdig documentation for information about how to install the Sysdig Agent.

Installation

Prerequisites

- Kubernetes version 1.16 or later.
- Helm version is 3.x.x or later.

Installation Steps

1. Add Federator.ai helm chart repository

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

2. Test the helm chart repository

```
$ helm search repo federatorai
```

3. Install Federator.ai helm chart with a release name (e.g., '**federatorai**') in the 'federatorai' namespace.

i *The default persistent storage class is 'default', use '--set global.storageClassName="**<name>**"' to configure an alternative storage class.*

```
$ helm install federatorai prophetstor/federatorai --namespace=federatorai --create-namespace

NAME: federatorai
LAST DEPLOYED: Wed Sep  2 09:06:28 2023
NAMESPACE: federatorai
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

NOTES:

CHART NAME: federatorai

CHART VERSION: 5.1.5

APP VERSION: 5.1.5-2296

**** Please be patient while the chart is being deployed ****

Watch the Federator.ai pods status using the command:

```
kubectl get pods -w --namespace federatorai -l app.kubernetes.io/part-of=federatorai
```

4. Confirm Federator.ai pods are running and ready

```
$ kubectl get pods -n federatorai
```

NAME	READY	STATUS	RESTARTS	AGE
alameda-ai-bf95746cc-wfjfl	1/1	Running	0	22h
alameda-ai-dispatcher-7f875f78bf-f4lts	1/1	Running	0	22h
alameda-datahub-578779bcdd-pdzz7	1/1	Running	0	22h
alameda-influxdb-0	1/1	Running	0	22h
alameda-notifier-7d6698fdb9-f5wr2	1/1	Running	0	22h
alameda-rabbitmq-95b8b859b-ngntb	1/1	Running	0	22h
fedemeter-api-dcff7786-rd984	1/1	Running	0	58m
fedemeter-influxdb-0	1/1	Running	0	22h
federatorai-agent-747d58f656-wd942	1/1	Running	0	22h
federatorai-alert-detector-665fffd45b-qwxjp	1/1	Running	0	22h
federatorai-alert-manager-75c5bdfdc9-lwpkx	1/1	Running	0	22h
federatorai-dashboard-backend-6ffb94bf8-gtgrf	1/1	Running	0	22h
federatorai-dashboard-frontend-ddd558559-8djfz	1/1	Running	0	22h
federatorai-data-adapter-77db9685c5-xz7mc	1/1	Running	0	22h
federatorai-postgresql-0	1/1	Running	0	22h
federatorai-recommender-dispatcher-5cdc6d76b5-x8mxb	1/1	Running	0	22h
federatorai-recommender-worker-57d64fdb8d-zbpkl	1/1	Running	0	22h
federatorai-rest-75ccb6f44-gl8cv	1/1	Running	0	22h

5. By default, Federator.ai helm chart creates a NodePort service for Federator.ai dashboard. If your environment does not allow creating NodePort services, you'll need to manually configure Kubernetes to expose the Federator.ai dashboard service ("*federatorai-dashboard-frontend*") for external access.

```
$ kubectl get svc federatorai-dashboard-frontend-public -n federatorai
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
federatorai-dashboard-frontend-public	NodePort	10.101.65.187	<none>	9001:31012/TCP	9d

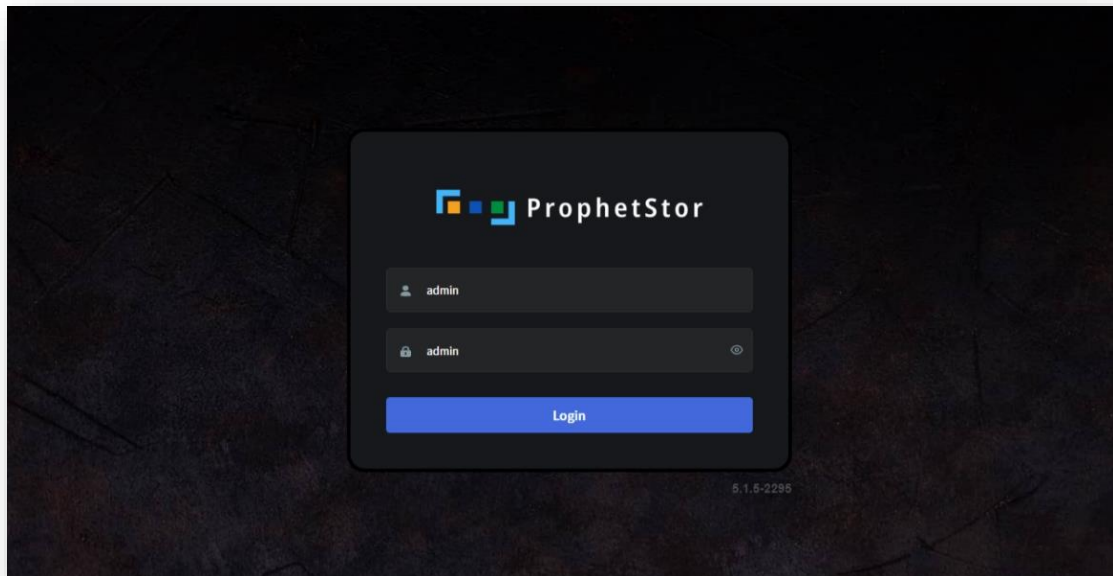
```
$ kubectl get svc federatorai-dashboard-frontend -n federatorai
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
federatorai-dashboard-frontend	ClusterIP	10.96.52.210	<none>	9000/TCP,9001/TCP	9d

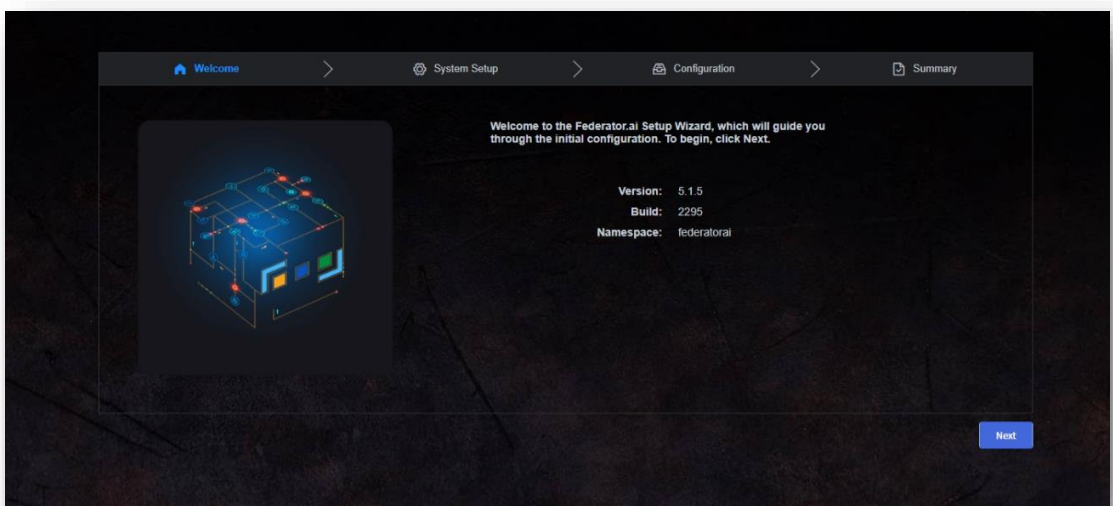
6. Connect to Federator.ai dashboard

```
https://<Kubernetes_IP>:31012
```

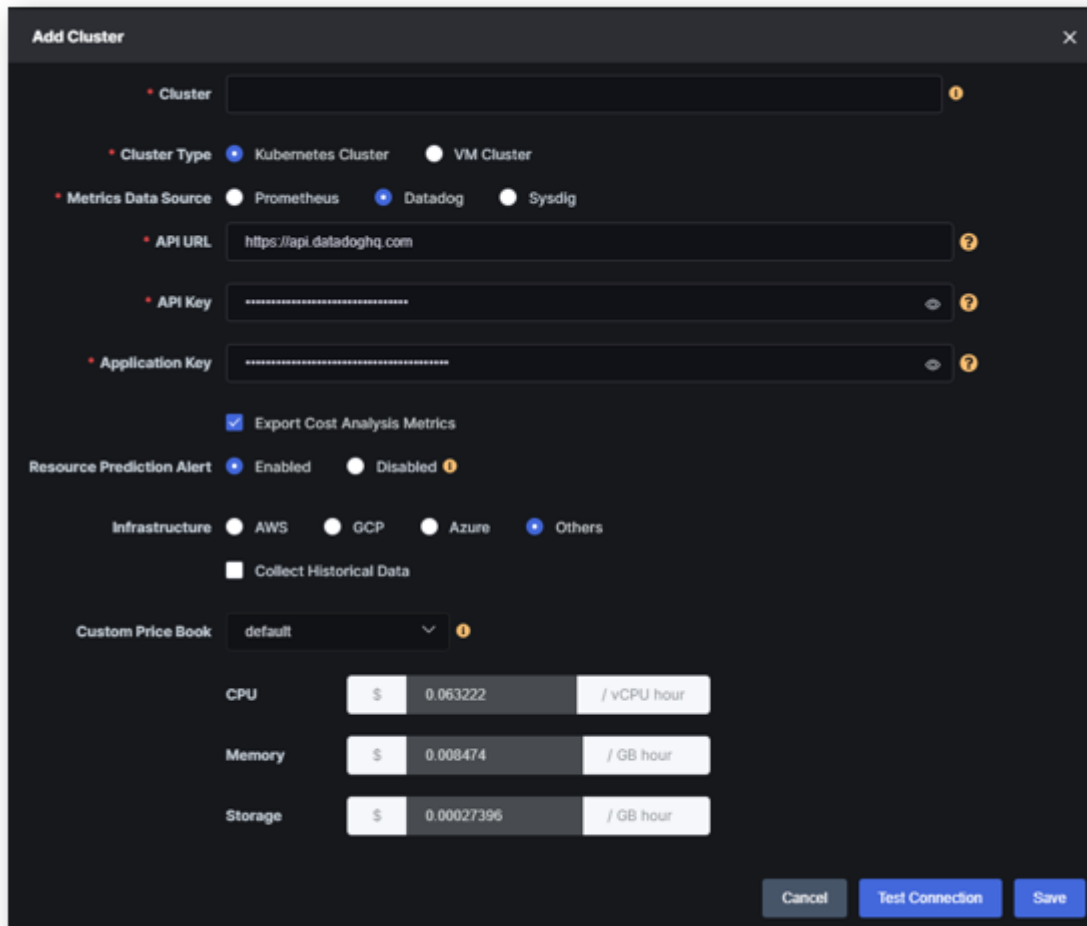
The default login username/password is "admin/admin".



7. After logging into the Federator.ai dashboard, you can follow the initial setup wizard to configure your first cluster and application (optional).



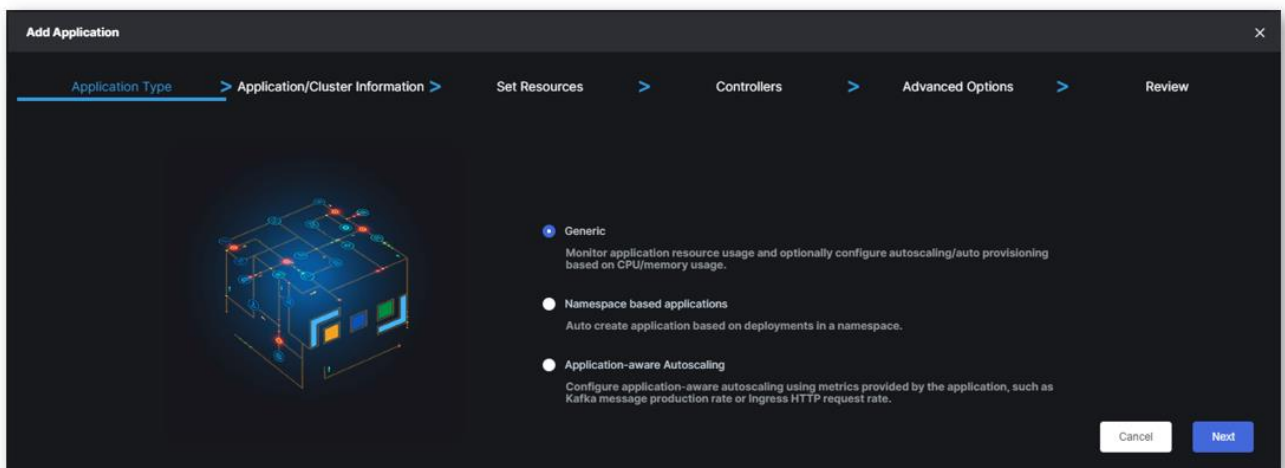
8. To manage additional clusters or applications, navigate to 'Configuration > Clusters' or 'Configuration > Applications', respectively, and click on 'Add Cluster' or 'Add Application' located in the upper-right corner.



The 'Add Cluster' dialog box is a dark-themed form with a close button (X) in the top right corner. It contains the following fields and options:

- Cluster:** A text input field with a yellow information icon (i) to its right.
- Cluster Type:** Two radio buttons: 'Kubernetes Cluster' (selected) and 'VM Cluster'.
- Metrics Data Source:** Three radio buttons: 'Prometheus', 'Datadog' (selected), and 'Sysdig'.
- API URL:** A text input field containing 'https://api.datadoghq.com' with a yellow information icon (i) to its right.
- API Key:** A password input field with a yellow information icon (i) to its right.
- Application Key:** A password input field with a yellow information icon (i) to its right.
- Export Cost Analysis Metrics:** A checked checkbox.
- Resource Prediction Alert:** Two radio buttons: 'Enabled' (selected) and 'Disabled' with a yellow information icon (i) to its right.
- Infrastructure:** Four radio buttons: 'AWS', 'GCP', 'Azure', and 'Others' (selected).
- Collect Historical Data:** An unchecked checkbox.
- Custom Price Book:** A dropdown menu showing 'default' with a yellow information icon (i) to its right.
- Resource Pricing Table:**

Resource	Unit	Price
CPU	/ vCPU hour	\$ 0.063222
Memory	/ GB hour	\$ 0.008474
Storage	/ GB hour	\$ 0.00027396
- Buttons:** 'Cancel', 'Test Connection', and 'Save' at the bottom right.



The 'Add Application' dialog box is a dark-themed form with a close button (X) in the top right corner. It features a multi-step navigation bar at the top: 'Application Type' (selected), 'Application/Cluster Information', 'Set Resources', 'Controllers', 'Advanced Options', and 'Review'. The main content area includes:

- Diagram:** A 3D wireframe cube with internal nodes and connections, representing a network or application structure.
- Application Type Selection:**
 - Generic:** Selected. Description: Monitor application resource usage and optionally configure autoscaling/auto provisioning based on CPU/memory usage.
 - Namespace based applications:** Description: Auto create application based on deployments in a namespace.
 - Application-aware Autoscaling:** Description: Configure application-aware autoscaling using metrics provided by the application, such as Kafka message production rate or Ingress HTTP request rate.
- Buttons:** 'Cancel' and 'Next' at the bottom right.

Uninstallation

- Uninstall/Delete the Federator.ai release (e.g., 'federatorai')

```
$ helm ls --all-namespaces  
$ helm delete federatorai --namespace=federatorai
```

Offline Installation

For certain restricted environments, pulling images from public image repository, and downloading any installation helper scripts from Internet are disallowed. Installing Federator.ai in such restricted environments using original installation process is not feasible.

This offline installation guide provides a two-stage manual installation procedure to assist users in installing Federator.ai from a private image repository in an environment with no Internet access.

Workflow

The two-stage offline installation procedure includes, at the first stage, preparing the installation package by downloading installation materials (Federator.ai images, helper scripts, helm chart) online, and at the second stage, uploading Federator.ai images to the private image repository and installing Federator.ai by using the downloaded helm chart.

- Set up a standalone bootstrap host (or VM) with necessary tools for preparing installation media.
- Run installation helper script to download images/scripts/helm chart files from Internet and save the materials in the bootstrap host.
- Connect the bootstrap host to the Kubernetes cluster, or if it is allowed, copy the installation materials to another installation host that is connected to the Kubernetes cluster.
- Run installation helper script on the bootstrap host or the installation host to upload Federator.ai images to the private image repository.
- Install Federator.ai by using the downloaded helm chart.

Bootstrap Host

A standalone bootstrap host is used for running the two-stage installation procedure. This host needs to have Internet access for preparing installation materials. Once installation materials are ready, connect the bootstrap host to the Kubernetes cluster if it is allowed for the second stage of installation.

The hardware and software requirements of the bootstrap host are:

- 2 vCPU and 4 GB memory
- 20 GB disk
- CentOS 7.5 and above

- Docker 1.13.1 and above
- Bash (part of CentOS 7.5 by default)
- ‘kubectf’ command (If the bootstrap is allowed to connect to the OpenShift/Kubernetes cluster, and used for installing Federator.ai without another installation host)

One possible approach is to create a VM on your laptop and run CentOS 7.5 in the VM.

Installation Steps

• Stage 1 – Download installation materials

1. Connect the bootstrap host to the Internet. Use the helper script on Github to prepare installation materials.

```
# Download helper script
$ curl -sOL https://raw.githubusercontent.com/containers-ai/prophetstor/v5.1.5-b2296/deploy/prepare-private-repository.sh
$
# Download helm chart file
$ curl -sOL https://raw.githubusercontent.com/prophetstor-ai/federatorai-operator-helm/master/federatorai-5.1.5-2296.tgz
$
$ bash ./prepare-private-repository.sh --pull --tag v5.1.5-b2296
Pulling image quay.io/prophetstor/alameda-ai-dispatcher:v5.1.5-b2296 ...
Saving alameda-ai-dispatcher image to file ...
alameda-ai-dispatcher.v5.1.5-b2296.tgz saved.

Pulling image quay.io/prophetstor/alameda-ai:v5.1.5-b2296 ...
Saving alameda-ai image to file ...
alameda-ai.v5.1.5-b2296.tgz saved.

Pulling image quay.io/prophetstor/alameda-datahub-ubi:v5.1.5-b2296 ...
Saving alameda-datahub-ubi image to file ...
alameda-datahub-ubi.v5.1.5-b2296.tgz saved.

Pulling image quay.io/prophetstor/alameda-executor-ubi:v5.1.5-b2296 ...
Saving alameda-executor-ubi image to file ...
alameda-executor-ubi.v5.1.5-b2296.tgz saved.

----- snipped -----

Pulling image quay.io/prophetstor/federatorai-telemetry:v5.1.5-b2296 ...
Saving federatorai-telemetry image to file ...
federatorai-telemetry.v5.1.5-b2296.tgz saved.

Pulling image quay.io/prophetstor/federatorai-alert-manager:v5.1.5-b2296 ...
Saving federatorai-alert-manager image to file ...
federatorai-alert-manager.v5.1.5-b2296.tgz saved.

$ ls
alameda-ai-dispatcher.v5.1.5-b2296.tgz    federatorai-agent-preloader.v5.1.5-b2296.tgz
alameda-ai.v5.1.5-b2296.tgz             federatorai-agent-ubi.v5.1.5-b2296.tgz
alameda-datahub-ubi.v5.1.5-b2296.tgz    federatorai-alert-manager.v5.1.5-b2296.tgz
alameda-executor-ubi.v5.1.5-b2296.tgz   federatorai-dashboard-backend.v5.1.5-b2296.tgz
```

alameda-influxdb.v5.1.5-b2296.tgz	federatorai-dashboard-frontend.v5.1.5-b2296.tgz
alameda-notifier-ubi.v5.1.5-b2296.tgz	federatorai-data-adapter.v5.1.5-b2296.tgz
alameda-rabbitmq.v5.1.5-b2296.tgz	federatorai-operator-ubi.v5.1.5-b2296.tgz
alameda-recommender-ubi.v5.1.5-b2296.tgz	federatorai-postgresql.v5.1.5-b2296.tgz
federatorai-rest-ubi.v5.1.5-b2296.tgz	fedemeter-api-ubi.v5.1.5-b2296.tgz
federatorai-telemetry.v5.1.5-b2296.tgz	fedemeter-influxdb.v5.1.5-b2296.tgz
prepare-private-repository.sh	federatorai-5.1.5-2296.tgz

● Stage 2 – Upload images and install

1. Connect the bootstrap host to the Kubernetes cluster or copy the installation package file from the bootstrap host to another installation host if the bootstrap host is not allowed to connect to the Kubernetes cluster.
2. Run the help script to upload Federator.ai images to the private image repository (e.g., “172.31.2.39:5000/prophetstor”).

```
$ bash ./prepare-private-repository.sh --push --repo-url 172.31.2.39:5000/prophetstor
Loading image file alameda-ai-dispatcher.v5.1.5-b2296.tgz ...
Loaded image: quay.io/prophetstor/alameda-ai-dispatcher:v5.1.5-b2296
Tagging alameda-ai-dispatcher:v5.1.5-b2296 image ...
Pushing alameda-ai-dispatcher:v5.1.5-b2296 image to 172.31.2.39:5000/prophetstor ...
Done

Loading image file alameda-ai.v5.1.5-b2296.tgz ...
101581272bf5: Loading layer 2.048kB/2.048kB
5e02fdb5d5fa: Loading layer 398.5MB/398.5MB
25bacd14017f: Loading layer 36.72MB/36.72MB
eab7589e4432: Loading layer 2.048kB/2.048kB
49809805b244: Loading layer 3.584kB/3.584kB
bd8457541a07: Loading layer 884.7kB/884.7kB
5a3bd211a328: Loading layer 61.95kB/61.95kB
c9025b12d853: Loading layer 67.82MB/67.82MB
Loaded image: quay.io/prophetstor/alameda-ai:v5.1.5-b2296
Tagging alameda-ai:v5.1.5-b2296 image ...
Pushing alameda-ai:v5.1.5-b2296 image to 172.31.2.39:5000/prophetstor ...
Done

Loading image file alameda-datahub-ubi.v5.1.5-b2296.tgz ...
6ddea03ae42e: Loading layer 2.048kB/2.048kB
8003c41db1e2: Loading layer 4.608kB/4.608kB
b3d389c8d6ab: Loading layer 26.62kB/26.62kB
0d5017885f1b: Loading layer 4.096kB/4.096kB
f09a2bc36048: Loading layer 155.4MB/155.4MB
ad775ea528c1: Loading layer 17.71MB/17.71MB
85207c17366a: Loading layer 2.396MB/2.396MB
Loaded image: quay.io/prophetstor/alameda-datahub-ubi:v5.1.5-b2296
Tagging alameda-datahub-ubi:v5.1.5-b2296 image ...
Pushing alameda-datahub-ubi:v5.1.5-b2296 image to 172.31.2.39:5000/prophetstor ...
Done

----- snipped -----

Loading image file federatorai-telemetry.v5.1.5-b2296.tgz ...
```

```

7dd57975a88c: Loading layer 3.072kB/3.072kB
2c23e78bcd9a: Loading layer 16.2MB/16.2MB
7da531939940: Loading layer 51.56MB/51.56MB
c0167e92b109: Loading layer 23.24MB/23.24MB
eab1b0f68acd: Loading layer 4.096kB/4.096kB
68d5d40a47ca: Loading layer 7.168kB/7.168kB
Loaded image: quay.io/prophetstor/federatorai-telemetry:v5.1.5-b2296
Tagging federatorai-telemetry:v5.1.5-b2296 image ...
Pushing federatorai-telemetry:v5.1.5-b2296 image to 172.31.2.39:5000/prophetstor ...
Done

Loading image file federatorai-alert-manager.v5.1.5-b2296.tgz ...
15cb503b1f0a: Loading layer 17.7MB/17.7MB
65319a53f067: Loading layer 2.048kB/2.048kB
82bff72aa794: Loading layer 4.608kB/4.608kB
87db0178d539: Loading layer 2.56kB/2.56kB
5f8f965c41bd: Loading layer 22.53kB/22.53kB
8031b7680542: Loading layer 49.86MB/49.86MB
6c2ddfdcef5d: Loading layer 48.23MB/48.23MB
a7aedcd7b8fd: Loading layer 45.69MB/45.69MB
a750d199d9d6: Loading layer 2.56kB/2.56kB
c352fa627d20: Loading layer 143.8MB/143.8MB
d16d5b70de02: Loading layer 3.072kB/3.072kB
Loaded image: quay.io/prophetstor/federatorai-alert-manager:v5.1.5-b2296
Tagging federatorai-alert-manager:v5.1.5-b2296 image ...
Pushing federatorai-alert-manager:v5.1.5-b2296 image to 172.31.2.39:5000/prophetstor ...
Done

```

3. Install Federator.ai by using the downloaded helm chart.

```

$ helm upgrade federatorai federatorai-5.1.5-2296.tgz --install --namespace=federatorai --
create-namespace --set global.imageRegistry=172.31.2.39:5000/prophetstor
Release "federatorai" does not exist. Installing it now.
NAME: federatorai
LAST DEPLOYED: Wed Nov  8 06:38:01 2023
NAMESPACE: federatorai
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: federatorai
CHART VERSION: 5.1.5-2296
APP VERSION: 5.1.5-2296

** Please be patient while the chart is being deployed **

Watch the Federator.ai pods status using the command:
    kubectl get pods -w --namespace federatorai -l app.kubernetes.io/part-of=federatorai
$
$ kubectl get pods -w --namespace federatorai -l app.kubernetes.io/part-of=federatorai

```

NAME	READY	STATUS	RESTARTS	AGE
alameda-ai-8578fbd98d-l4wvb	1/1	Running	0	8m
alameda-ai-dispatcher-85bd6786b7-4mckr	1/1	Running	0	8m
alameda-datahub-5fdb96b874-j2tfv	1/1	Running	0	8m
alameda-influxdb-0	1/1	Running	0	8m
alameda-notifier-785dc44dcc-hwtjt	1/1	Running	0	8m
alameda-rabbitmq-5ccd7f996c-82h4z	1/1	Running	0	8m
fedemeter-api-54f4458df7-x57dv	1/1	Running	0	8m
fedemeter-influxdb-0	1/1	Running	0	8m

federatorai-agent-5d87ccdc5b-12482	1/1	Running	0	8m
federatorai-alert-detector-dc487b6bb-kjdpj	1/1	Running	0	8m
federatorai-alert-manager-67c74766d5-nv5nb	1/1	Running	0	8m
federatorai-dashboard-backend-5689c9cd44-zcjb7	1/1	Running	0	8m
federatorai-dashboard-frontend-56848bfcbb-x6z7r	1/1	Running	0	8m
federatorai-data-adapter-7dcf59698f-97jvb	1/1	Running	0	8m
federatorai-postgresql-0	1/1	Running	0	8m
federatorai-recommender-dispatcher-7d44fc5cf7-fjsmv	1/1	Running	0	8m
federatorai-recommender-worker-65d5b68d9d-7tltw	1/1	Running	0	8m
federatorai-rest-6d5fcf45c9-mcnzb	1/1	Running	0	8m

Parameter (optional)

The tables below list the adjustable parameters of the helm chart. These parameters are optional, and you do not need to specify them unless you have specific requirements in your environment.

Global parameters

Parameter	Description
global.imageRegistry	Image registry
global.imageTag	Image tag of Federator.ai
global.imagePullPolicy	Container image pull policy
global.imagePullSecrets	Array of optional imagePullSecrets
global.storageClassName	Persistent Volume Storage Class
global.commonAnnotations	Common annotations to be added to resources
global.commonLabels	Common labels to be added to resources
global.podAnnotations	Annotations to be added to pods
global.podLabels	Labels to be added to pods
global.resourcesLimitsEnabled	Enable/Disable resource limits
global.resourcesRequestsEnabled	Enable/Disable resource requests

alamedaAi Parameters

Parameter	Description
alamedaAi.persistence.dataStorageSize	Persistent volume size of data volume

alamedaExecutor Parameters

Parameter	Description
alamedaExecutor.enabled	Enable deployment of alameda-executor

alamedaInfluxdb Parameters

Parameter	Description
alamedaInfluxdb.persistence.dataStorageSize	Persistent volume size of data volume

fedemeterInfluxdb Parameters

Parameter	Description
fedemeterInfluxdb.persistence.dataStorageSize	Persistent volume size of data volume

federatoraiDashboardFrontend Parameters

Parameter	Description
<code>federatoraiDashboardFrontend.ingress.enabled</code>	Enable/Disable ingress controller resource
<code>federatoraiDashboardFrontend.ingress.pathType</code>	Ingress Path type
<code>federatoraiDashboardFrontend.ingress.hostname</code>	Default host for the ingress resource
<code>federatoraiDashboardFrontend.ingress.path</code>	The Path of REST. You may need to set this to '/' in order to use this with ALB ingress controllers
<code>federatoraiDashboardFrontend.ingress.annotations</code>	Additional annotations for the Ingress resource. To enable certificate autogeneration, place here your cert-manager annotations.
<code>federatoraiDashboardFrontend.ingress.ingressClassName</code>	IngressClass that will be used to implement the Ingress (Kubernetes 1.18+)
<code>federatoraiDashboardFrontend.ingress.tls</code>	Enable TLS configuration for the hostname defined at <code>federatoraiDashboardFrontend.ingress.hostname</code> parameter
<code>federatoraiDashboardFrontend.ingress.extraHosts</code>	The list of additional hostnames to be covered with this ingress record.
<code>federatoraiDashboardFrontend.ingress.extraPaths</code>	Additional arbitrary path/backend objects
<code>federatoraiDashboardFrontend.ingress.extraTls</code>	The tls configuration for additional hostnames to be covered with this ingress record.
<code>federatoraiDashboardFrontend.ingress.secrets</code>	If you're providing your own certificates, please use this to add the certificates as secrets
<code>federatoraiDashboardFrontend.service.type</code>	Kubernetes service type, valid value: LoadBalancer, NodePort
<code>federatoraiDashboardFrontend.service.port</code>	Public service port
<code>federatoraiDashboardFrontend.service.targetPort</code>	Container port of services, use 9000 for accessing over HTTP and 9001 for accessing over HTTPS
<code>federatoraiDashboardFrontend.service.clusterIP</code>	Specific cluster IP when service type is cluster IP. Use None for headless service
<code>federatoraiDashboardFrontend.service.nodePort</code>	Kubernetes Service nodePort if service type is NodePort
<code>federatoraiDashboardFrontend.service.loadBalancerIP</code>	Load Balancer IP Address if service type is LoadBalancer
<code>federatoraiDashboardFrontend.service.loadBalancerSourceRanges</code>	Address that are allowed when svc is LoadBalancer
<code>federatoraiDashboardFrontend.service.externalTrafficPolicy</code>	Enable client source IP preservation
<code>federatoraiDashboardFrontend.service.healthCheckNodePort</code>	Specifies the health check node port (numeric port number) for the service if externalTrafficPolicy is set to Local.
<code>federatoraiDashboardFrontend.service.annotations</code>	Additional annotations for REST service

federatoraiPostgresql Parameters

Parameter	Description
<code>federatoraiPostgresql.persistence.dataStorageSize</code>	Persistent volume size of data volume

You can specify each parameter by using the "--set key=value[,key=value]" argument when running "helm install".

Alternatively, you can provide a YAML file during the chart installation, which specifies the values for the parameters. For example,

```
$ helm install federatorai prophetstor/federatorai -f values.yaml --namespace=federatorai --create-namespace
```

The YAML file, "values.yaml", can be downloaded by the command,

```
$ helm show values prophetstor/federatorai > values.yaml
```