IBM Tivoli Netcool/OMNIbus Gateway for IBM Cloud Event Management Helm Chart 1.1.0

Reference Guide
March 20, 2020

SC28-3108-00

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 19.

Edition notice

This edition (SC28-3108-00) applies to version 1.1.0 of IBM Tivoli Netcool/OMNIbus Gateway for IBM Cloud Event Management Helm Chart and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC28-3108-01.

© Copyright International Business Machines Corporation 2019, 2020.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this guide

The following sections contain important information about using this guide.

Document control page

Use this information to track changes between versions of this guide.

The Gateway for IBM Cloud Event Management Helm Chart documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/helms/common/Helms.html

Table 1. Document modification history

Document version	Publication date	Comments
SC28-3108-00	October 31, 2019	First IBM publication.
SC28-3108-01	March 20, 2020	Guide updated for version 1.1.0. "Installing IBM Cloud Platform Common Services" on page 3 added. The following topics updated: • "Installing the chart" on page 9 • "Pre-installation tasks" on page 5 • "Configurable parameters for the CEM Gateway" on page 11 • "Configuring ObjectServer connection parameters" on page 15 • "Limitations" on page 17 • "Troubleshooting" on page 17

© Copyright IBM Corp. 2019, 2020

Chapter 1. Gateway for IBM Cloud Event Management Helm Chart

The Gateway for IBM Cloud Event Management Helm Chart allows you to deploy the IBM Netcool/OMNIbus Gateway for Cloud Event Management onto Kubernetes. This gateway processes events and alerts from IBM Netcool/OMNIbus ObjectServer and forwards them to the IBM Cloud Event Management (CEM) dashboard.

Note : This Helm Chart is soon to be deprecated. You should use instead, or migrate to, the IBM Netcool Operations Insight Event Integrations Operator when running on Red Hat OpenShift Container Platform. For details see https://www.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/operators/noi_operator/wip/reference/noiop_intro_noi_operator.html. There will be no updates to the deprecated chart.

This guide contains the following sections:

The Knowledge Center contains the following additional topics that contain information that is common to all Helm Charts:

- Specifying the image repository
- Loading PPA packages to IBM Cloud Private
- Exposing the service
- Upgrading to a new version of the helm chart
- Changing the service type during a helm upgrade
- Role-Based Access Control page

Obtaining the PPA package

You can download the installation package from the IBM Passport Advantage website.

Use the **Find by part number** field to search for the following part number: CC5J2EN

Chart details

The chart deploys an IBM Netcool/OMNIbus Gateway for IBM Cloud Event Management (CEM) onto Kubernetes to forward NOI events into CEM dashboard.

This chart can be deployed more than once on the same namespace.

© Copyright IBM Corp. 2019, 2020

Prerequisites

This solution requires the following applications:

- IBM Tivoli Netcool/OMNIbus ObjectServer to be created and running on Red Hat OpenShift Container Platform (OCP) 4.3 with IBM Cloud Platform Common Services 3.2.4 prior to installing the chart. IBM Netcool Operations Insight 1.6.0.3 Helm Chart version 2.1.3 is required to create IBM Tivoli Netcool/OMNIbus ObjectServer on OCP 4.3. See the following topic on the IBM Knowledge Center: IBM Knowledge Center - Installing on Red Hat OpenShift.
- Kubernetes 1.16
- IBM Netcool Operations Insight 1.6.0.3 Helm Chart version 2.1.3
- IBM Cloud Event Management Helm Chart version 2.5.0

Note : Administrator role is a minimum requirement to install this chart.

The chart must be installed by a Administrator to perform the following tasks:

- Retrieve and edit sensitive information from a secret such as the credentials to use to authenticate with the ObjectServer or replace the key database files for secure communications with the ObjectServer.

The chart must be installed by a Cluster Administrator to perform the following tasks in addition to those listed above:

- Create a new namespace with custom SecurityContextConstraints if necessary. For details see "Red Hat OpenShift security context constraints" on page 3.
- Create a service account in the namespace for this chart. Perform one of the following actions:
  - Have the Cluster Administrator pre-create the custom service account in the namespace. This installation requires the service account name to specified to install the chart and can be done by an Administrator.
  - Have the Cluster Administrator perform the installation without specifying a service account name so that the chart generates a service account and use it. When the Helm release is deleted, the service account will also be deleted.
- If secured communication is required or enabled on your Netcool/OMNIbus ObjectService, a pre-created secret is required for this chart to establish a secured connection with the ObjectServer.
- Additional ObjectServer fields are required in the `alerts.status` table for IBM CEM integration. Refer to Integrating IBM Cloud Event Management (CEM) with Netcool Operations Insight section for details of the Structured Query Language (SQL) needed to add the required fields. For NOI on ICP, the required fields are already added.
- The Cloud Event Management must have a valid signed certificate. See the **Global properties** section in Installing in IBM Cloud Private add Ingress TLS secret.
- The secured communication between Cloud Event Management and the gateway is required. If the CEM is using a self-signed certificate then a pre-created secret is required for this chart to establish a secure trusted connection with the Cloud Event Management. The secret must contain `tls.crt` which is the CEM TLS certificate file in PEM format. You must set **`cemgateway.cemTlsSecretName`** parameter to this secret name.
- You must store CEM gateway sensitive information in a pre-created secret. The secret contains **`CEMWebhookURL`**, **`NewKeystorePassword`** and **`HttpAuthenticationPassword`**. You must set **`cemgateway.cemSecretName`** to this secret name.

  `CEMWebhookURL` is the CEM webhook URL and the required key (mandatory parameter) in the secret.

  `NewKeystorePassword` is the new password to access truststore used by the gateway and the optional key in the secret.

  `HttpAuthenticationPassword` is the HTTP basic authentication password string which is required by the test pod (`helm test`). This is only used by an internal API to check the liveness of the CEM Gateway.

- IBM Cloud Platform Common Services 3.2.4 must be installed prior to install this chart. See Install IBM Cloud Platform Common Services for details.
- If you opt to install the chart through the command line, the Helm CLI must be installed. See the following topic on the IBM Knowledge Center: Installing Helm CLI (helm).

Resources required

This solution requires the following resources:

- CPU Requested : 100m (100 millicpu)
- Memory Requested : 128Mi (~ 134 MB)

Installing IBM Cloud Platform Common Services

Ensure the following Common Services are enabled prior to installing the chart.

```
# ## Common service definition
# management_services:
#   # Default base services
#   tiller: enabled
#   monitoring-crd: enabled
#   mongodb: enabled
#   platform-api: enabled
#   icp-management-ingress: enabled
#   internal-management-ingress: enabled
#   helm-api: enabled
#   helm-repo: enabled
#   mgmt-repo: enabled
#   oidcclient-watcher: enabled
#   secret-watcher: enabled
#   security-onboarding: enabled

#   # Default core services
#   cert-manager: enabled
#   cert-manager-webhook: enabled
#   configmap-watcher: enabled
#   auth-idp: enabled
#   auth-apikeys: enabled
#   auth-pap: enabled
#   auth-pdp: enabled
#   iam-policy-controller: enabled
#   metering: enabled
#   licensing: disabled
#   monitoring: enabled
#   nginx-ingress: enabled
#   catalog-ui: enabled
#   mcm-kui: enabled
#   logging: enabled
#   common-web-ui: enabled

#   # mcm services
#   multicluster-hub: enabled
#   search: enabled
#   key-management: enabled
#   image-security-enforcement: enabled

#   # Deprecated services
#   calico: enabled
#   kube-dns: enabled
```

Note : If IBM Cloud Platform Common Services is not installed, see the following topic on the IBM Knowledge Center: Installing Common Services.

Red Hat OpenShift security context constraints

On Red Hat OpenShift Container Platform, this chart requires a SecurityContextConstraints to be bound to the target namespace prior to installation. To meet this requirement there may be cluster scoped as well as namespace scoped pre and post actions that need to occur.

The predefined PodSecurityPolicy name `ibm-restricted-scc` has been verified for this chart. If your target namespace is bound to this SecurityContextConstraints resource, you can proceed to install the chart.

This chart also defines a custom SecurityContextConstraints which can be used to finely control the permissions/capabilities needed to deploy this chart. You can enable this custom SecurityContextConstraints resource using the the supplied instructions/scripts in the `pak_extension pre-install` directory.

The OpenShift Container Platform (OCP) provides pod security policies using SecurityContextConstraints (SCC) resources rather than the PodSecurityPolicies (PSP) like all other Kubernetes platforms. SCCs control the actions that a pod can perform and what it has the ability to access. IBM Cloud Private on OCP installations uses SCCs instead of PSPs.

- From the user interface, you can copy and paste the following snippets to enable the custom PodSecurityPolicy:
  - Custom SecurityContextConstraints definition:

```
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: "This policy is the most restrictive,
      requiring pods to run with a non-root UID, and preventing pods from accessing the
host.
      The UID and GID will be bound by ranges specified at the Namespace level."
    cloudpak.ibm.com/version: "1.1.0"
  name: ibm-netcool-gateway-cem-prod-scc
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegedContainer: false
allowPrivilegeEscalation: false
allowedCapabilities: null
allowedFlexVolumes: null
allowedUnsafeSysctls: null
defaultAddCapabilities: null
defaultAllowPrivilegeEscalation: false
forbiddenSysctls:
  - "*"
fsGroup:
  type: MustRunAs
  ranges:
  - max: 65535
    min: 1
readOnlyRootFilesystem: false
requiredDropCapabilities:
- ALL
runAsUser:
  type: MustRunAsNonRoot
seccompProfiles:
- docker/default
seLinuxContext:
  type: RunAsAny
supplementalGroups:
  type: MustRunAs
  ranges:
  - max: 65535
    min: 1
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret
```

- From the command line, you can run the setup scripts included under `pak_extensions`.

  As a cluster administrator, the pre-install scripts and instructions are in the following location:

  `pre-install/clusterAdministration/createSecurityClusterPrereqs.sh`

As team admin/operator the namespace scoped scripts and instructions are in the following location:

`pre-install/namespaceAdministration/createSecurityNamespacePrereqs.sh`

Securing gateway and ObjectServer communication requirements

There are several mechanisms to secure Netcool/OMNIbus. Authentication can be used to restrict user access while Secure Sockets Layer (SSL) protocol can be used for different levels of encryption.

The gateway connection mode is dependent on the server component configuration. Check with your Netcool/OMNIbus Administrator whether the server is configured with either secured mode enabled without SSL, SSL enabled with secured mode disabled, or secured mode enabled with SSL protected communications

The chart must be configured according to the server components setup in order to establish a secured connection with or without SSL. This can be configured by setting the `netcool.connectionMode` chart parameter with one of the following options:

- `AuthOnly` - Use this mode when the ObjectServer is configured to run in secured mode without SSL.This is the default mode.
- `SSLAndAuth` - Use this mode the ObjectServer is configured with SSL and secure mode.

To secure the communications between gateway clients and the ObjectServer, you must perform the following tasks that must be completed before installing the chart. The steps are outlined in the **Pre-installation tasks** section.

Note : There are several known limitations when securing communications. See the **Limitations** section.

Role-based access control

Role-Based Access Control (RBAC) is applied to the chart by using a custom service account having a specific role binding. RBAC provides greater security by ensuring that the chart operates within the specified scope.

For details about creating the RBAC resources, see Role-Based Access Control page.

Pre-installation tasks

Before installing the helm chart, you must perform the following tasks:

1. Gathering ObjectServer details
2. Preparing ObjectServer communication secret
3. Preparing the Cloud Event Management (CEM) incoming integration
4. Preparing the Cloud Event Management (CEM) gateway secret
5. Preparing the Cloud Event Management (CEM) TLS secret
6. Pre-creating a Persistent Volume (PV)

1. Gathering ObjectServer details

For the gateway to successfully connect to ObjectServer, the gateway must be configured with:

1. ObjectServer host or service details
2. ObjectServer TLS certificate if SSL protected communication is enabled.
3. Credentials to authenticate with the ObjectServer.

Note : In production environments, it is recommended to use TLS/SSL enabled communications with the ObjectServer.

Follow these steps to obtain the required details for the ObjectServer:

1. Contact your administrator to find out the ObjectServer that the CEM Gateway should connect to. Note the NOI release name and namespace as `NOI_RELEASE_NAME` and `NOI_NAMESPACE` respectively. This information will be used when preparing the ObjectServer communication Kubernetes secret later.
2. Determine the ObjectServer NodePort service name and port number. Refer to Connecting with the ObjectServer NodePort for more info on identifying the ObjectServer NodePort and note down the service names and the NodePort number as `NOI_OBJECT_SERVER_PRIMARY_SERVICE` and `NOI_OBJECT_SERVER_PRIMARY_PORT` respectively. You may also note the backup ObjectServer service if you wish to connect to the backup ObjectServer too as `NOI_OBJECT_SERVER_BACKUP_SERVICE` and `NOI_OBJECT_SERVER_BACKUP_PORT` respectively . This information will be used when configuring the chart.
3. Determine the TLS Proxy listening port by following the steps in Identifying the proxy listening port page and note down the TLS proxy Common Name (CN) and port number as `NOI_TLS_PROXY_CN` and `NOI_TLS_PROXY_PORT` respectively. This information will be used when configuring the chart.
4. Determine the TLS Proxy certificate Kubernetes secret that should be used by the CEM Gateway. This is usually set in `{{ Release.Name}}-proxy-tls-secret` secret, where `{{ Release.Name }}` is the NOI release name. Note down the secret name and namespace. This information will be used when preparing the ObjectServer communication Kubernetes secret later.
5. Get the ObjectServer user password which is required by the Gateway when creating and Insert, Delete, Update, or Control (IDUC) communication protocol connection. The credential is provided in the `{{ Release.Name }}-omni-secret`, where `{{ Release.Name }}` is the NOI release name.
6. Obtain the cluster proxy IP address.

The "Gathered Facts" table below lists the details that is gathered from the above steps. These items will be referenced in the following sections.

Item	**Description and sample value**
CLUSTER_MASTER_IP	The cluster master node IP address. This should be set as the `netcool.primaryIP` and optionally `netcool.backupIP` to also connect to the backup ObjectServer.
CLUSTER_MASTER_HOST	The cluster master node hostname. This should be set as the `netcool.primaryHost` and optionally `netcool.backupHost` to also connect to the backup ObjectServer.
NOI_RELEASE_NAME	The NOI release name. For example, `noi-m76`
NOI_NAMESPACE	Namespace where NOI is installed. For example, `default`
NOI_OBJECT_SERVER_PRIMARY_SERVICE	The primary ObjectServer Nodeport service. For example, `noi-m76-objserv-agg-primary-nodeport`. This should be set as the `netcool.primaryIDUCHost` parameter.
NOI_OBJECT_SERVER_PRIMARY_PORT	The primary ObjectServer Nodeport number. This should be set as the `netcool.primaryPort` parameter.
NOI_OBJECT_SERVER_BACKUP_SERVICE	(Optional) The backup ObjectServer Nodeport service. For example `noi-m76-objserv-agg-backup-nodeport`. This should be set as the `netcool.backupIDUCHost` parameter.
NOI_OBJECT_SERVER_BACKUP_PORT	(Optional) The backup ObjectServer Nodeport number

Item	Description and sample value
NOI_TLS_PROXY_CN	The NOI TLS certificate subject Common Name. For example `proxy.noi-m76.mycluster.icp`. This should be set as the `netcool.primaryHost` (and optionally `netcool.backupHost`). For more details on how to obtain the Subject Common Name, see Configuring TLS encryption with the default certificate or Configuring TLS encryption with a custom certificate pages.
NOI_TLS_PROXY_PORT_1	The NOI TLS Proxy service port number (first port).
NOI_TLS_PROXY_PORT_2	(Optional) The NOI TLS Proxy service port number (second port), required to connect to backup ObjectServer.
NOI_TLS_SECRET_NAME	Secret name containing the TLS certificate of the TLS Proxy. For example `noi-m76-proxy-tls-secret`
NOI_OMNI_USER	Username for IDUC connection.
NOI_OMNI_PASSWORD	Password for IDUC connection.

2. Preparing ObjectServer communication secret

The secret can be created using the utility script ("create-noi-secret.sh") provided in the `pak_extensions/pre-install` directory. Before running this script, several pieces of information must be gathered and then configured in the script's configuration file ("create-noi-secret.config") which should be obtained following the steps in "Gathering ObjectServer Details" section above.

For this task, you will need the CEM Gateway image in your local file system because the script requires several utility commands such as `nco_gskcmd` and `nco_aes_crypt` to add the ObjectServer certificate into the key database.

Note : If you are using a root CA signer and want to import it into the Key Database file as a trusted CA signer, see the following Technote for instructions: https://www.ibm.com/support/pages/node/1274896

1. Follow the steps in Pushing and pulling images to pull the CEM Gateway image `netcool-gateway-cem` into your local file system. The following steps uses `netcool-gateway-cem:latest` as the image name and image tag for simplicity.
2. From the command line, review and update the create-noi-secret.sh utility script configuration file (`create-noi-secret.config`) file provided in the `pak_extensions/pre-install` directory. Several items from the "Gathered Facts" table above is required when configuring the script.
3. Run the "create-noi-secret.sh" to create the Gateway-ObjectServer communication secret. The script should be run as an administrator or a user with read permissions to the NOI TLS secret so that the script can retrieve the TLS certificate file.
4. Optionally, verify that the secret is successfully created using the `kubectl describe secret <secret name> --namespace <namespace>` command.

```
kubectl describe secret cem-gw-noi-secret --namespace cemgw-ns
Name:         cem-gw-noi-secret
Namespace:    cemgw-ns
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
omni.sth:            193 bytes
AuthPassword:        70 bytes
AuthUserName:        4 bytes
```

```
encryption.keyfile:  36 bytes
omni.kdb:            10088 bytes
```

3. Preparing the Cloud Event Management (CEM) incoming integration

To forward events from IBM Netcool Operations Insight (NOI) to IBM CEM, the `ibm-cem` chart can be installed in the same cluster as NOI or in another cluster. NOI and the CEM Gateway need to be installed in the same cluster. For detailed steps on installing and configuring IBM CEM, see Installing with Netcool Operations Insight.

The CEM Gateway requires the CEM webhook URL from the Netcool/OMNIbus Incoming Integration in CEM. Follow the steps below to create the incoming integration.

1. Login to the CEM User Interface as an administrator.
2. Click **Integrations** on the CEM **Administration** page.
3. Click **New integration**.
4. Go to the **Netcool/OMNIbus** tile and click **Configure**.
5. Enter a name for the integration and click **Copy** to add the generated webhook URL to the clipboard. Ensure you save the generated webhook to a file. The CEM webhook URL is required when creating CEM Gateway secret.
6. Enable this integration.
7. Click the **Save** button.

4. Preparing the Cloud Event Management (CEM) gateway secret

The chart requires CEM Gateway Secret to send the events to CEM. This secret contains sensitive information such as `CEMWebhookURL`, `NewKeystorePassword` and `HttpAuthenticationPassword`. `CEMWebhookURL` is the CEM webhook URL and it is the required key in the secret. `NewKeystorePassword` is the new password to access cacerts in gateway and it is an optional key in the secret. `HttpAuthenticationPassword` is the HTTP basic authentication password string which is required by the test pod (`helm test`). This is only used by an internal API to check the liveness of the CEM Gateway and it is an optional key in the secret.

You can refer to the example below to create the secret. In the example, `cem-gateway-secret` is the secret name. `cem-gateway-namespace` is the namespace where the CEM gateway will be installed. `cem-webhook-url` is CEM webhook URL. `http-basic-authentication-password` is HTTP basic authentication password. `new-keystore-password` is the new password to access cacerts in gateway. The secret and chart must reside in the same namespace.

```
kubectl create secret generic cem-gateway-secret --namespace cemgw-ns \
--from-literal=CEMWebhookURL=cem-webhook-url \
--from-literal=NewKeystorePassword=new-keystore-password \
--from-literal=HttpAuthenticationPassword=http-basic-authentication-password
```

Optionally, verify that the secret is successfully created using the `kubectl describe secret cem-gateway-secret --namespace cem-gateway-namespace` command.

```
kubectl describe secret cem-gateway-secret --namespace cemgw-ns
Name:         cem-gateway-secret
Namespace:    cemgw-ns
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
CEMWebhookURL:               60 bytes
NewKeystorePassword:         10 bytes
HttpAuthenticationPassword:  10 bytes
```

5. Preparing the Cloud Event Management (CEM) TLS secret

To allow the CEM Gateway to send events to CEM on OCP, a Transport Layer Security (TLS) certificate for Fully Qualified Domain Name (FQDN) of the CEM must be obtained to establish a secure trusted connection between the CEM Gateway and CEM. You must create the secret, if the TLS certificate is not signed by a well known certificate authority. Follow these steps to obtain the CEM TLS certificate from the CEM Ingress TLS secret.

1. Login to the cluster where CEM installed.
2. Obtain the CEM TLS certificate from the CEM Ingress TLS secret. The sample command below can be used to obtain the CEM TLS certificate from the CEM Ingress TLS secret. In the sample command below, `cem-tls-secret-name` is the CEM Ingress TLS secret name, `cem-tls-secret-namespace` is the namespace where the secret created, and `tls.crt` is the file contains the command output.

   ```
   kubectl get secret cem-tls-secret-name \
   --namespace cem-tls-secret-namespace \
   -o json | grep tls.crt | cut -d : -f2 | cut -d '"' -f2 | base64 --decode > tls.crt
   ```

3. Login to the cluster where the CEM Gateway will be installed and create the CEM TLS secret. The sample command below can be used to create CEM TLS secret. In the sample command that follows, `cem-tls-secret` is the secret name, `cemgw-ns` is the namespace where the CEM gateway will be installed and `tls.crt` is a CEM TLS certificate file. The CEM TLS certificate filename must be `tls.crt`.

   ```
   kubectl create secret generic cem-tls-secret \
   --namespace cemgw-ns \
   --from-file=tls.crt
   ```

6. Pre-creating a Persistent Volume (PV)

The chart requires a Persistent Volume (PV) to store Store and Forward (SAF) files. You can opt to use dynamic provisioning and skip this step, if your cluster supports dynamic provisioning. Otherwise, to pre-create a PV or if you want the chart Persistent Volume Claim (PVC) to bind to a pre-created PV. You should only perform one of the following steps to pre-create a PV.

- Refer to Understanding persistent storage to pre-create a PV of your choice.
- If your cluster supports Network File System (NFS) PV. Refer to the sample YAML file of creating NFS PV in `pak_extensions/pre-install/clusterAdministration/ibm-netcool-gateway-cem-prod-nfs-pv.yaml` to pre-create a NFS PV. Refer to comments in `ibm-netcool-gateway-cem-prod-nfs-pv.yaml` and update the YAML file accordingly. Then, run this command `kubectl create -f ibm-netcool-gateway-cem-prod-nfs-pv.yaml` as a cluster administrator to provision a NFS PV. The details of creating NFS PV are available in Persistent storage using NFS.

Installing the chart

To install the chart, use one of the following methods:

Install the chart using the IBM Cloud Pak dashboard console

1. From the IBM Cloud Pak dashboard console, open the **Catalog**.
2. Locate and select the `ibm-netcool-gateway-cem-prod` chart.
3. Review the provided instructions and click **Configure**.
4. Provide a release name and select the namespace and cluster.
5. Review and accept the license(s).
6. The Configuration table provides the required configuration based on the requirements specific to your installation. See “Configurable parameters for the CEM Gateway” on page 11.

7. Click the **Install** button to complete the helm installation.

Install the chart using the command line

1. Extract the helm chart archive and customize `values.yaml`. The configuration section lists the parameters that can be configured during installation, see "Configurable parameters for the CEM Gateway" on page 11.
2. Refer to "Configuring ObjectServer connection parameters" on page 15.
3. Install the chart with the release name `my-gateway` using the configuration specified in the customized `values.yaml` using following command:

   ```
   helm install --tls --namespace cemgw-ns --name my-gateway -f values.yaml
   stable/ibm-netcool-gateway-cem-prod
   ```

   Where: *`my-gateway`* is the release name for the chart.

   Helm searches for the `ibm-netcool-gateway` chart in the helm repository called `stable`. This assumes that the chart exists in the stable repository.

   Tip : You can list all releases using `helm list --tls` or you can search for a chart using **helm search**.

Verifying the chart

See the instructions at the end of the helm installation for chart verification. The instructions can also be displayed by viewing the installed helm release under **Menu -> Workloads -> Helm Releases** or by running the following command:

```
helm status <release> --tls
```

Improving the performance of the Cloud Event Management (CEM) Gateway

You can increase the number of simultaneous connections between CEM Gateway and CEM to improve the performance. In general, the more connections specified, the faster the gateway can create requests in CEM from alerts sent by the ObjectServer. You can change this parameter before or after installing the chart by referring to the following steps.

Edit the parameter before installing the chart

1. The Configuration table provides the required configuration based on the requirements specific to your installation. See Update cemgateway.connections in values.yaml. Then, save value.yaml.
2. Proceed to chart installation.

Edit the parameter after installing the chart

1. Obtain the ConfigMap name for the current CEM Gateway Helm release. Use the sample command below to query the ConfigMap and write the output into a file. In the sample command below, <release>-gateway-cem-config is the CEM Gateway ConfigMap name, cemgw-ns is namespace where the chart is installed and cemgw-configmap.yaml is YAML file contains output of the command.

   ```
   kubectl get configmap <release>-gateway-cem-config -n cemgw-ns -o yaml > cemgw-configmap.yaml
   ```

2. Update the related parameter in the ConfigMap to increase the number of connections. Open cemgw-configmap.yaml and update the Gate.CEM.Connections value. Then, save cemgw-configmap.yaml.
3. Replace the CEM Gateway ConfigMap with the modified ConfigMap. Use the sample command below to replace the CEM Gateway ConfigMap with the modified ConfigMap. In the sample command below,

<release>-gateway-cem-config is the CEM Gateway ConfigMap name, cemgw-ns is namespace where the chart is installed and cemgw-configmap.yaml is the modified YAML file in the step above.

```
kubectl replace configmap <release>-gateway-cem-config -n cemgw-ns -f cemgw-configmap.yaml
```

4. Restart the CEM Gateway StatefulSet with the modified ConfigMap in the step above. Use the sample commands below to scale down CEM Gateway StatefulSet to 0 then scale up CEM Gateway StatefulSet to 1. In the sample commands below, <cem-gateway-statefulset> is CEM Gateway StatefulSet name and cemgw-ns is namespace where the chart is installed.

```
kubectl scale statefulset/<cem-gateway-statefulset> -n cemgw-ns --replicas=0
kubectl scale statefulset/<cem-gateway-statefulset> -n cemgw-ns --replicas=1
```

Uninstalling the chart

To uninstall the chart, use the following steps:

1. Run the following command:

   ```
   $ helm delete my-gateway --purge --tls
   ```

   Where: *my-gateway* is the release name for the chart.

   The command removes all the Kubernetes components associated with the chart and deletes the release.

2. Access the storage using the UI, select **Menu -> Platform -> Storage**. Review any orphaned Persistent Volume Claims (PVC) and delete if necessary. Otherwise, run pak_extensions/post-delete/clusterAdministration/deletePersistentVolume.sh as a cluster administrator to delete the PVC and Persistent Volume (PV) for the release. The script takes two arguments which are the namespace and the release name that has been uninstalled. Example usage: bash deletePersistentVolume.sh namespace chartReleaseName

Clean up any prerequisites that were created

As a Cluster Administrator, run the cluster administration cleanup script included under `pak_extensions` to clean up cluster scoped resources when appropriate:

`post-delete/clusterAdministration/deleteSecurityClusterPrereqs.sh`

As a Cluster Administrator, run the namespace administration cleanup script included under `pak_extensions` to clean up namespace scoped resources when appropriate:

`post-delete/namespaceAdministration/deleteSecurityNamespacePrereqs.sh`

As a Cluster Administrator, run the cluster administration cleanup script included under `pak_extensions` to delete Persistent Volume Claim (PVC) and Persistent Volume (PV) :

`post-delete/clusterAdministration/deletePersistentVolume.sh`

Configuring the chart

The following topics describe how to configure the helm chart.

Configurable parameters for the CEM Gateway

You use parameters to specify how the gateway interacts with the device. You can override the chart's default parameter settings during installation.

The following table describes the configurable parameters for this chart and lists their default values.

Configurable parameters

Parameter name	Description
license	The license state of the image being deployed. Enter `accept` to install and use the image. The default value is `not accepted`.
image.repository	Gateway image repository. Update this repository name to pull from a private image repository. For details see Specifying the image repository. The default value is `netcool-gateway-cem`, and must not be changed.
image.tag	The image tag. The default value is `2.1.0-amd64`.
image.pullPolicy	The image pull policy. The default value is `Always`.
global.image.secretName	The name of the secret containing the docker config to pull the image from a private repository. Leave this parameter blank if the gateway image already exists in the local image repository or the Service Account has a been assigned with an Image Pull Secret. The default value is `nil`.
global.serviceAccountName	Description: Name of the service account to be used by the helm chart. If the Cluster Administrator has already created a service account in the namespace, specify the name of the service account here. If left blank, the chart will automatically create a new service account in the namespace when it is deployed. This new service account will be removed from the namespace when the chart is removed. The default is `nil`.
global.persistence.useDynamicProvisioning	Use Storage Class to dynamically create Persistent Volume and Persistent Volume Claim. The default is `true`.
global.persistence.storageClassName	Storage Class for dynamic provisioning. The default is `""`.
global.persistence.selector.label	The Persistent Volume Claim Selector label key to refine the binding process when dynamic provisioning is not used. The default is `""`.
global.persistence.value	The Persistent Volume Claim Selector label value related to the Persistent Volume Claim Selector label key. The default is `""`.
global.persistence.storageSize	Storage size to store CEM Gateway Store and Forward Files. The default is `3Gi`.
global.persistence.supplementalGroups	Provide the gid of the volumes as list (required for NFS). The default is `[]`.

Parameter name	Description
netcool.connectionMode	The connection mode to use when connecting to the Netcool/OMNIbus ObjectServer. Refer to Securing probe and ObjectServer communications for more details. **Note:** Refer to the limitations section for more details on available connection modes for your environment. The default is `AuthOnly`.
netcool.primaryServer	The primary Netcool/OMNIbus server to connect to. This is usually set to `NCOMS` or `AGG_P`. The default value is `AGG_P`.
netcool.primaryHost	The host of the primary Netcool/OMNIbus server. Specify the ObjectServer hostname or IP address. The default value is `nil`
netcool.PrimaryIP	The primary Netcool/OMNIbus ObjectServer IP address. If specified along with primaryHost, a host alias entry will be added. The default is `nil`.
netcool.primaryPort	The port of the primary Netcool/OMNIbus server. You may need to update the port number if a different port number is used, for example, if you are connecting to the NOI TLS Proxy Service. The default value is `4100`.
netcool.primaryIDUCHost	The primary Netcool/OMNIbus ObjectServer IDUC Host or Service name. Should be set if the primary IDUC host is different from the primary ObjectServer hostname. When connecting to NOI on OCP, this should be set to the primary ObjectServer NodePort service name. The default is `nil`.
netcool.backupServer	The backup Netcool/OMNIbus server to connect to. If the **backupServer**, **backupHost** and **backupPort** parameters are defined in addition to the **primaryServer**, **primaryHost**, and **primaryPort** parameters, the gateway will be configured to connect to a virtual object server pair called `` `AGG_V` ``. If no backup ObjectServer is configured, only the primary server parameters will be used. The default value is `nil`.
netcool.backupHost	The host of the backup Netcool/OMNIbus server. Specify the ObjectServer hostname or IP address. The default value is `nil`
netcool.backupIP	The backup Netcool/OMNIbus ObjectServer IP address. If specified along with primaryHost, a host alias entry will be added. The default is `nil`.
netcool.backupPort	The port of the backup Netcool/OMNIbus server. The default value is `nil`.

Parameter name	Description
netcool.backupIDUCHost	The backup Netcool/OMNIbus ObjectServer IDUC Host or Service name. Should be set if the primary IDUC host is different from the primary ObjectServer hostname. When connecting to NOI on OCP, this should be set to the primary ObjectServer NodePort service name. The default is `nil`.
netcool.secretName	This is a pre-created secret for `AuthOnly`, `SSLOnly` or `SSLAndAuth` connection mode. Certain fields are required depending on the connection modes. The default is `nil`.
cemgateway.messageLevel	The gateway log message level. The default value is `warn`.
cemgateway.connections	The number of simultaneous connections the gateway makes with the Cloud Event Management. The default value is 3
cemgateway.connectionTimeout	The interval (in seconds) that the gateway allows for HTTP connections and responses to HTTP requests. The default value is `15`
cemgateway.retryLimit	The maximum number of retries the gateway should make on an operation (for example, forwarding an event to the Event Source instance) that failed. The default value of 0 (zero) means there is no limit to the number of retries that the gateway makes on a failed operation. The default value is `0`
cemgateway.retryWait	The number of seconds the gateway should wait before retrying an operation (for example, forwarding an event to the Event Source instance) that failed. The default value is 7
cemgateway.reconnectTimeout	The time (in seconds) between each reconnection poll attempt that the gateway makes if the connection to the ObjectServer is lost. The default value is `30`
cemgateway.locale	Environment locale setting. Used as the LC_ALL environment variable. The default is `en_US.utf8`.
cemgateway.setUIDandGID	When set to true, the helm chart will specify the UID and GID values for the netcool user else the netcool user will not be assigned any UID or GID by the helm chart.
cemgateway.cemTlsSecretName	A pre-created secret name to store CEM certificate. In the secret, the key must be tls.crt which holds the certificate file in PEM format.

Parameter name	Description
cemgateway.cemSecretName	A pre-created secret to store CEMWebhookURL, NewKeystorePassword and HttpAuthenticationPassword. CEMWebhookURL is the CEM webhook URL to send notification from gateway to CEM and the required key in the secret. NewKeystorePassword is the new password to access cacerts in gateway and the optional key in the secret. HttpAuthenticationPassword is the HTTP basic authentication password string which is required by the test pod (helm test) and only used by an internal API to check the liveness of the CEM Gateway. HttpAuthenticationPassword is the optional key in the secret
resources.requests.cpu	The minimum required CPU core. Specify integers, fractions (for example 0.5), or millicore values (for example 100m, where 100m is equivalent to .1 core). The default value is `100m`.
resources.requests.memory	The minimum memory in bytes. Specify integers with one of these suffixes: E, P, T, G, M, K, or power-of-two equivalents: Ei, Pi, Ti, Gi, Mi, Ki. The default value is `128Mi`.
resources.limits.cpu	The upper limit of the CPU core. Specify integers, fractions (for example 0.5), or millicore values (for example 100m, where 100m is equivalent to .1 core). The default value is `500m`.
resources.limits.memory	The memory upper limit in bytes. Specify integers with one of these suffixes: E, P, T, G, M, K, or power-of-two equivalents: Ei, Pi, Ti, Gi, Mi, Ki. The default value is `512Mi`.
arch	The worker node architecture. This is set to `amd64`, and cannot be changed.

Configuring ObjectServer connection parameters

After completing the pre-installation tasks, you can configure the Netcool/OMNIbus parameters. The following parameters show a sample configuration YAML to connect to a Netcool/OMNIbus ObjectServer in OCP using the TLS proxy:

```
netcool:
  # (Required) The connection mode to use.
  connectionMode: "SSLAndAuth"

  # (Required) ObjectServer Name that the gateway should connect to. (Usually set to NCOMS or
AGG_P)
  primaryServer: "AGG_P"

  # (Required) Hostname of the primary ObjectServer. Set to "<NOI_TLS_PROXY_CN>"
  primaryHost: "noi-m76-proxy-tls-secret"
```

```
# IP address of the primary ObjectServer host. This should be the set to the
# OCP Master node IP address for NOI on OCP.
# If both host and IP parameters are specified,
# and entry will be added as host alias.
primaryIP: "9.30.117.58"

# (Required) ObjectServer Port. Set to "<NOI_TLS_PROXY_PORT_1>", 3XXXX.
primaryPort: 30135

# For NOI on OCP, set to the primary ObjectServer
# nodeport service name "<NOI_OBJECT_SERVER_PRIMARY_SERVICE>"
primaryIDUCHost: "noi-m76-objserv-agg-primary-nodeport"

# (Optional) Backup ObjectServer Name that the gateway should connect to.
backupServer: "AGG_B"

# (Optional) Hostname of the backup ObjectServer. Set to <NOI_TLS_PROXY_CN>
backupHost: "noi-m76-proxy-tls-secret"

# (Optional) IP address of the backup ObjectServer host. This should be the set to the
# OCP Master node IP address for NOI on OCP.
# If both host and IP parameters are specified,
# and entry will be added as host alias. Set to "<CLUSTER_MASTER_IP>", same as 'primaryIP'
backupIP: "9.30.117.58"

# (Optional) Backup ObjectServer Port, set to "<NOI_TLS_PROXY_PORT_2>", 3XXXX
backupPort: 30456

# (Optional) For NOI on OCP, set to the backup ObjectServer nodeport
# service name "<NOI_OBJECT_SERVER_BACKUP_SERVICE>"
backupIDUCHost: "noi-m76-objserv-agg-backup-nodeport"

# (Required) A pre-created secret for AuthOnly or SSLAndAuth connection mode.
# This is the secret name configured in the create-noi-secret.sh script configuration file.
# It should contain several fields required for securing the connection between the gateway
# and ObjectServer.
secretName: "cemgw-noi-secret"
```

Storage

The CEM Gateway uses Persistent Volume (PV) to store Store and Forward files (SAF). The SAF files are used to keep track of the events already sent to CEM so that the CEM Gateway will not resend the same events to CEM to avoid event duplications in CEM. When the CEM Gateway Pod is restarted, it will access the SAF to check the last event sent to CEM and start forwarding the next events to CEM. You can opt to use Kubernetes dynamic provisioning to create both PV and Persistent Volume Claim (PVC) or pre-create a PV with label for the chart to perform the binding process. The PV must have access mode of `ReadWriteOnce` and minimum of `3 GiB` storage capacity. Kubernetes dynamic provisioning is the default option to provision the PV and PVC dynamically.

To provision PV and PVC dynamically, set the following global values:

- **`persistence.useDynamicProvisioning`** to `true`. The default is `true`.
- **`persistence.storageClassName`** to custom Storage Class name or leave the value empty to use default Storage Class. The default is "".

To bind to a pre-created PV,

- Set the following global values:
  - **`persistence.useDynamicProvisioning`** to `false`. The default is `true`.
  - **`persistence.storageClassName`** to custom Storage Class name or leave the value empty to use default Storage Class. The default is "".
  - **`persistence.selector.label`** to target PV label for PV and PVC binding process. The default is "".
  - **`persistence.selector.value`** to the value related to **`persistence.selector.label`**. The default is "".
- The chart deployment will unintentionally attempt to do dynamic provisioning, if **`global.persistence.storageClassName`** is set to a valid Storage Class name without specifying

global.persistence.selector.label and **global.persistence.selector.value**. To avoid this case, **global.persistence.selector.label** and **global.persistence.selector.value** must be specified with non-empty values.

Limitations

This solution has the following limitations:

- Only the AMD64 / x86_64 architecture is supported.
- It is verified to run on Red Hat OpenShift Container Platform 4.3.
- There are several known limitations when enabling a secure connection between gateway clients and the server:
  - The required files in the secret must be created using the `nc_gskcmd` utility.
  - If your ObjectServer is configured with FIPS 140-2, the password for the key database file (`omni.kdb`) must meet the requirements stated in the following IBM Knowledge Center page: Creating a key database using nc_gskcmd.
  - When encrypting a string value using the encryption config key file (`encryption.keyfile`), you must use AES_FIPS as the cipher algorithm. AES is not supported.
  - When connecting to an ObjectServer in the same IBM Cloud Private cluster, you may connect the gateway to the secure connection proxy which is deployed with the IBM Netcool Operations Insight chart to encrypt the communication using TLS, but the TLS termination is done at the proxy. You should enable IPSec on IBM Cloud Private to secure cluster data network communications.
  - Only one active CEM Gateway pod can use the Store-and-Forward (SAF) directory which is stored in the Persistent Volume. You should scale down the **StatefulSet** to 0 and then back to 1 in order to start the CEM Gateway properly during a node maintenance, and it should be able to resume processing the last known SAF file.

Troubleshooting

The following table describes how to troubleshoot issues when deploying the chart and how to resolve them.

Table 2. Problems

Problem	Cause	Resolution
Pod failed to mount to the Config Maps or some object name appear to be some what random.	If a long release name is used, the chart will generate a random suffix for objects that exceeds the character limit. This may cause mapping issues between the Kubernetes objects.	Use a shorter release name, below 20 characters.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

© Copyright IBM Corp. 2019, 2020

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

SC28-3108-00