



IBM Cloud

IBM Cloud Security and Compliance Center Data Security Broker

Product guide

Edition notices

This PDF was created on 2023-09-29 as a supplement to *IBM Cloud Security and Compliance Center Data Security Broker* in the IBM Cloud docs. It might not be a complete set of information or the latest version. For the latest information, see the IBM Cloud documentation at <https://cloud.ibm.com/docs/security-broker>.

© IBM Corp. 2023

Getting started with IBM Cloud Security and Compliance Center Data Security Broker

Overview

Protect your data in the cloud with the IBM Cloud Data Security Broker, which is a complete data encryption solution that secures sensitive data in enterprise databases by integrating with key management and databases to provide application-level encryption.

Data Security Broker is a software that makes data breaches irrelevant by ensuring data remains encrypted, not only when it is stored but also when it is being processed by databases and applications.

Data Security Broker offers Data encryption services which consists of two main components, namely:

IBM Cloud Security and Compliance Center Data Security Broker Manager is the administrative console for the solution that integrates with enterprise key managers and databases and manages the Data Security Broker solution components.


IBM Cloud Security and Compliance Center Data Security Broker Shield is the SQL proxy that functions to encrypt and decrypt data at the field or record level.

Data Security Broker Manager enforces encryption policies and configurations by:

- Communicating with key management solutions, the Data Security Broker Shield, and databases.
- Orchestrating configuration and deployment.

Data Security Broker Shield is a stateless proxy that intercepts and encrypts application data sent to the database and decrypts encrypted data.

Data Security Broker provide a range of data encryption services such as data encryption, data tokenization, record level encryption, and data masking.

 **Note:** Data Security Broker supports only PostgreSQL database.

Setting up your environment

Before you begin

Before you begin installing and configuring the Data Security Broker, ensure that you have met the following requirements:

- Create an IBM Cloud account.
- Set up your environment.
- Set up the minimum permissions required in the IBM Cloud account.

Ensure that your environment meets the following minimum system level and resource level requirements:

Cluster	Operating System	Number of Worker nodes required
Red Hat® OpenShift® cluster	RHEL7/RHEL8 and CoreOS	2
IBM Cloud Kubernetes cluster	Ubuntu 18	2

Table 1. Resource level requirements for Data Security Broker

Minimum permissions required to install, set up, and access Data Security Broker

As an IBM Cloud user, you need to set the following minimum permissions to install, set up and access Data Security Broker. By using the following steps and the information in the table, assign the required permissions:

1. Log into your IBM Cloud account and click **Manage -> Access (IAM)**.
2. In the Manage access and users dashboard, click **View all** in the **My user details** section.
3. In the **Access** tab, click **Assign access +**. From the Table 2, select a service and click **Next**.
4. In the **Roles and actions** section, select the specified permissions that are required.
5. Click **Add** and **Assign** to assign the permissions required.

Service Name	Permission level
Key Protect	Writer
IBM Cloud® Object Storage	Manager

Kubernetes Service	Manager, Editor
IBM Red Hat OpenShift Kubernetes Service	Editor
Schematics	Manager, Administrator

Table 3. Permissions required for Data Security Broker

You can find details about platform roles and the actions mapped to each of the role in the table below:

Platform Roles	Description
Administrator	As an administrator, you can perform all platform actions based on the resource this role is being assigned, including assigning access policies to other users.
Editor	As an editor, you can perform all platform actions except for managing the account and assigning access policies.
Operator	As an operator, you can perform platform actions required to configure and operate service instances, such as viewing a service's dashboard.
Viewer	As a viewer, you can view service instances, but you can't modify them.

Table 4. Platform roles and their actions

Sizing Guidelines

The factors that affect the sizing of the Data Security Broker deployments consist of the Data Security Broker Manager management console and one or more Data Security Broker Shield proxies. Each component has its own resource needs depending on the anticipated workloads.

Data Security Broker Manager

In general, resources allocated to a Data Security Broker Manager deployment needs to be scaled with the number of managed Data Security Broker Shields and the number of concurrent users using the Data Security Broker Manager.

Data Security Broker Shield

The general rule for Data Security Broker Shield sizing, to handle peak utilization scenarios, is to match the sum of all Data Security Broker Shield's memory and CPU allocations to that of the database instance. The initial vCPU and memory requests for the pod installation can start low and can be scaled up based on utilization, based on pod scaling policies, and depending on the workload in a particular installation. Resource allocation to Data Security Broker Shield deployments typically scales with the expected maximum number of concurrent connections.

Data Security Broker Shield consists of a single container that runs in its own pod. The Data Security Broker Shield pod can be in the same or different cluster but must have network connectivity to Data Security Broker Manager.

Minimum system requirements for deploying in IBM Cloud Kubernetes cluster (IKS) or Red Hat® OpenShift® Kubernetes (ROKS) cluster:

Product	Container/service	IKS/ROKSVersion	vCPU	Memory	Disk Space
Data Security Broker (DSB) Manager	DSB-manager	IKS v 1.17+, ROKS v 4.8.54+	4	8 GB	5 GB
Data Security Broker (DSB) Shield	DSB-shield	IKS v 1.17+, ROKS v 4.8.54+	2	8 GB	(No persistent volume necessary)

Table 2. Sizing guidelines

After the environment is set up and you have the required roles and permissions, and sizing guidelines defined, you can start setting installing the Data Security Broker by following the instructions in the [Installing Data Security Broker](#) section.

Next Steps

Now that you have an understanding of the various entities that exist within Data Security Broker, and that you have setup the environment to install the

Data Security Broker, the following diagram details the user flows that you might be helpful when you are working with Data Security Broker.

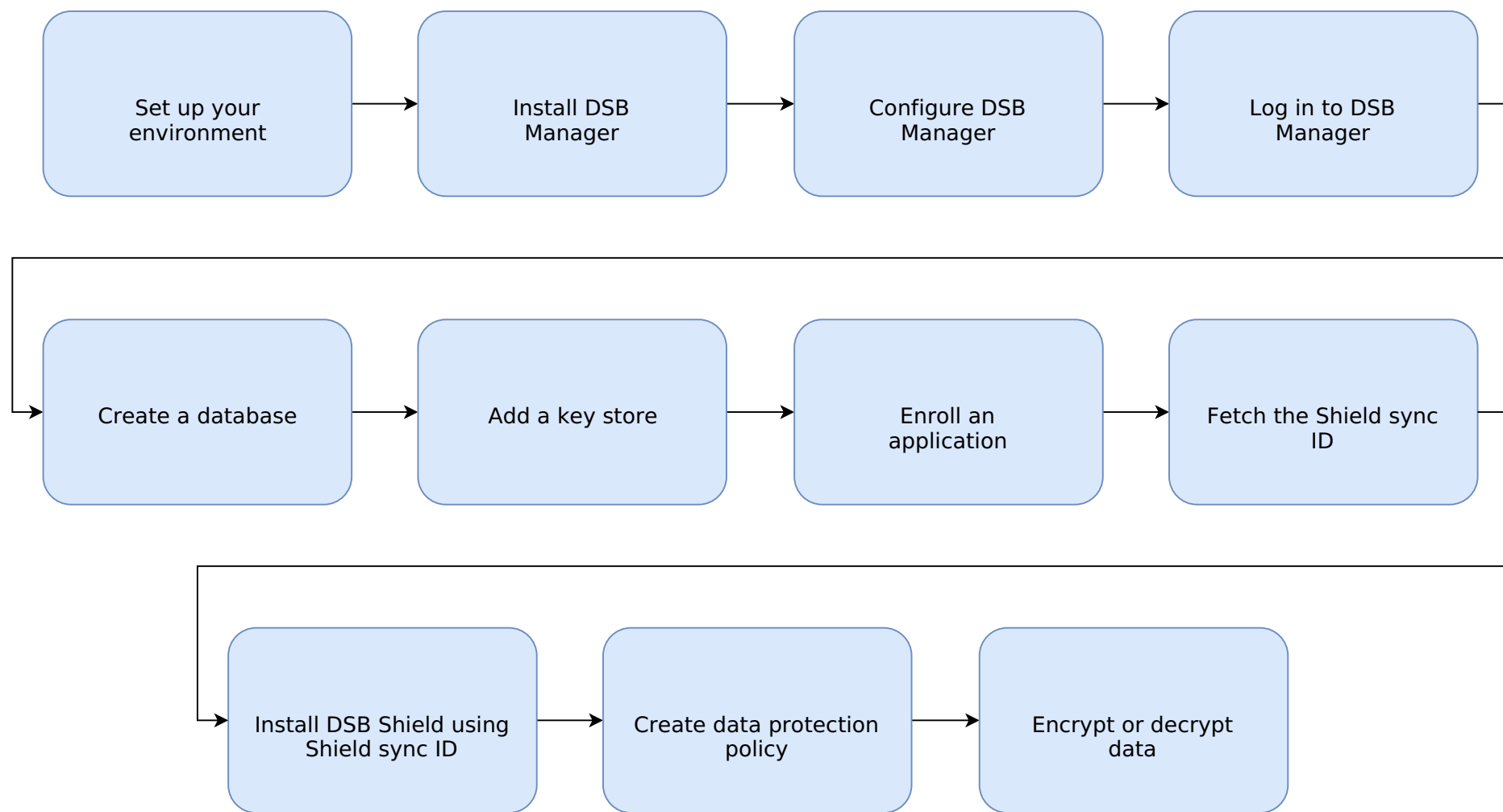


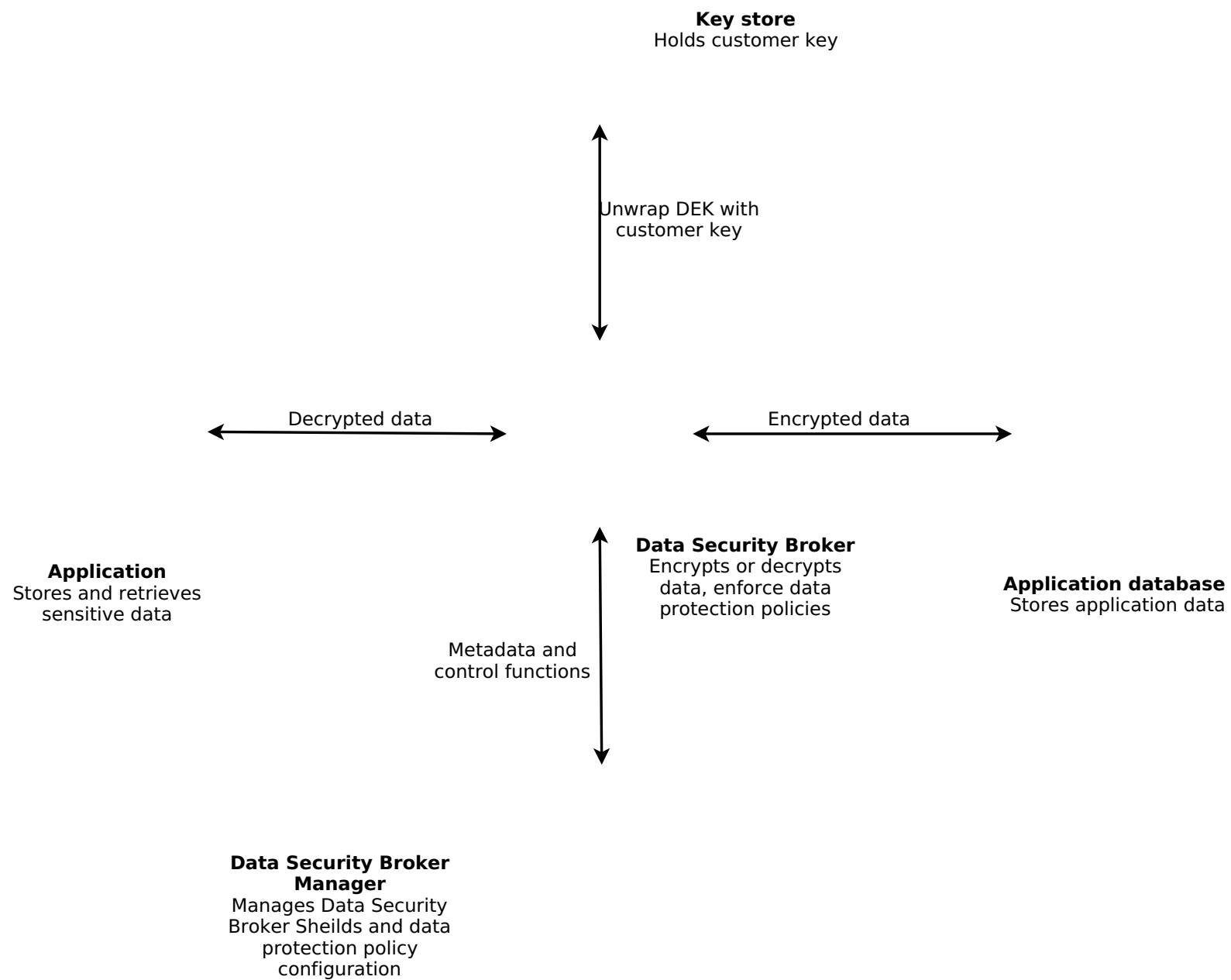
Figure 1. User Workflow

How does IBM Cloud Security and Compliance Center Data Security Broker work?

Data Security Broker delivers an enterprise-level transparent data security platform that secures databases through a "no code" model at the field or file level. The software supports tokenization, format-preserving encryption (FPE), and role-based access control.

The encryption is made simpler, faster, and seamless.

High level architecture of Data Security Broker Manager



The Data Security Broker supports data encryption by migrating the data from an existing SQL database into a secure database with the ability to encrypt fields at a column granularity, thereby enabling selective privacy.

This is accomplished by inserting the Data Security Broker Shield driver layer proxy between the application and the database. The Shield is used to intercept SQL equivalent generated by the application to access data records in the database by encrypting on writes and decrypting on reads.

The encryption process manages keys that are generated by a commercially available key management service such as IBM® Key Protect and Hyper Protect Crypto Services.

Application performance is minimally impacted allowing for enterprise workflows to continue to operate in a secure environment.

How does Data Security Broker calculate pricing?

Pricing for Data Security Broker depends on the number of Data Security Broker Shield instances that you have installed.



Important: For the most up-to-date pricing information, you can refer to the provisioning page for Data Security Broker Shield in the IBM Cloud Catalog. The displayed prices do not include tax.

Plan Type

The service offers Hourly Pay-as-you-go pricing plan.

Every Data Security Broker Shield instance that you install, is charged on an hourly basis.



Note: Data Security Broker Manager instances are not charged.

When am I charged?

You are charged right after the Data Security Broker Shield instance is installed in your cluster.

How do I stop getting charged for Data Security Broker?

If you no longer want to be charged for a specific shield instance, uninstall the specific shield instance that you do not want to be charged for, by following the instructions in the [Uninstalling Data Security Broker Shield](#) section.

Release notes for Data Security Broker

Use this release notes to learn about the latest changes to the IBM Cloud Data Security Broker documentation.

29 September 2023

IBM Cloud Data Security Broker 1.70.1650 version is now generally available

Fixes to some of the Common Vulnerabilities and Exposures (CVEs) are provided in this version along with the bug fixes.

Security enhancements

The following security enhancements were incorporated: - Added support for second-level domain name configuration for Data Security Broker Manager user accounts. As part of this enhancement, Data Security Broker users with an email on a sub-domain of the primary domain name can now enroll successfully. - Additional security enhancement has been enforced wherein, if you try to log in with an invalid OAuth token for more than 5 times, the Data Security Broker Manager login gets locked and you must attempt to log in again after 30 minutes.

Pricing plans

Pricing plans have been added to Data Security Broker. For more information, see [Pricing plans for Data Security Broker](#).

13 February 2023

Introducing IBM Cloud Data Security Broker Beta version 1.70.855

IBM Cloud Data Security Broker is a complete data protection solution that secures sensitive data in enterprise databases by integrating with key management and databases to provide application-level encryption. The following features are supported in the Beta version:

- Data encryption and decryption
- Data masking

Installation and Setup

Install Data Security Broker Manager

Overview

Access the IBM Cloud catalog to install the Data Security Broker Manager. You need to install Data Security Broker Manager first and then proceed with Data Security Broker Shield installation.



Note: Ensure that the Data Security Broker Manager must not be exposed to the public network. Only authorized personnel must have access to the Data Security Broker Manager.

Pre-requisites:

- User must be able to access the IBM Cloud Kubernetes cluster (IKS) or Red Hat® OpenShift® Kubernetes (ROKS) cluster.

Procedure

1. Log into the IBM Cloud Account (<https://cloud.ibm.com>) with a valid username and password.
2. Click **Catalog** and enter **Data Security Broker** in the **Catalog** search text box. The Data Security Broker components appear in the Catalog.
3. To sort the catalog products using the type, click **Software** in the **Type** option, which is present in the left-hand navigation.
4. The Data Security Broker software comprises of two components, which is displayed in the Catalog list as **Data Security Broker Manager** and **Data Security Broker Shield**.
5. You must install the **Data Security Broker Manager** first and then get the Shield Sync ID from the application in the **Data Security Broker Manager** to install the **Data Security Broker Shield**.

Install Data Security Broker Manager

Complete the following steps to install the **Data Security Broker Manager** from the IBM Cloud catalog:

1. Click **Data Security Broker Manager** catalog item.
2. The Data Security Broker Manager catalog item opens in a separate window. In the **Select your deployment target** drop down, select **IBM Cloud® Kubernetes Service (IKS)** or **Red Hat Openshift (ROKS)** to install the Data Security Broker Manager in the IKS or ROKS cluster.
3. The delivery method is selected as **HELM chart** by default under the **Select a delivery method** drop down.
4. Select the version of the software to install, in the **Select Version** drop-down.
5. From the list of available clusters, select the cluster on which you wish to perform the installation.
6. Select an existing namespace to deploy the Data Security Broker Manager or click **Add namespace** to add a new namespace. Specify the name for the namespace and click **Add** to create a new namespace within the selected cluster.
7. You must run the preinstallation script by clicking on the **Run Script** button after you select the cluster and the namespace. If you do not run the preinstallation script, you cannot proceed with the installation of the Data Security Broker Manager.
8. Configure your workspace by specifying the following details:
 - a. Specify the **Name** for the workspace. The workspace name must be unique and using the name of the workspace, you can manage, update or uninstall Data Security Broker Manager from the [IBM Schematicss Workspace](#).
 - b. Select the **Resource group**, **Location**, and specify the **Tags** required for configuring the workspace.
9. Set the input variables, as mentioned below:
 - a. Specify the password, which is used to encrypt the MangoDB password in the **secrets.credstorePass** parameter.
 - b. Specify the password, which is used to unlock the initial registration page for Data Security Broker Manager in the **secrets.initPass** parameter.

It is recommended to save the passwords for later use.
10. Click **Install** in the **Summary** pane on the right to complete the installation process.
11. You will be navigated to the Schematics Workspace to track the installation progress. Once the installation is successful, a message **Workspace creation successfull** is displayed.



Note: If you get an error message saying, **Workspace creation failed**, refer to the Logs available in the Terraform output.

Next Steps

After installing Data Security Broker Manager, you can start configuring the After installing Data Security Broker Manager by following the steps mentioned in the [Configure Data Security Broker Manager](#) section.

Configure Data Security Broker Manager

Configuring the Data Security Broker Manager console is the first step to log in and implement the data encryption services.

Prerequisites

Before you begin configuring Data Security Broker Manager, ensure you meet the following requirements:

- Data Security Broker Manager must have been installed.
- Load Balancer URL is required to access the Data Security Broker Manager.

To obtain the **Load_balancer_url** of the **dsb-nginx** service, perform the following steps.

- Log into your IBM Cloud account.
- Select **Resource List** from the left navigation menu. Click **Containers** to view the list of clusters.
- Select the cluster where you have installed the Data Security Broker Manager instance.

IBM Cloud® Kubernetes Service

- If you have installed the Data Security Broker Manager in a public IBM Cloud Kubernetes Service cluster, follow the below steps:
- Click **Kubernetes Dashboard** from the cluster, where you have installed Data Security Broker Manager.
- Select the namespace from the drop-down, on which you have installed the Data Security Broker Manager.
- Navigate to **Services** to view the list of Data Security Broker Manager services running in the namespace.
- Fetch the **LoadBalancer IP** from the **External Endpoints** column for the **dsb-nginx** service.

Red Hat® OpenShift®

- If you have installed the Data Security Broker Manager in a public Red Hat OpenShift cluster, click **Openshift web console** from the cluster.
- Click **Projects** in the left navigation menu and select the project from the drop-down, on which you have installed the Data Security Broker Manager.
- Navigate to **Networking -> Routes** to view the list of Data Security Broker Manager services running in the project.
- Fetch the **Data Security Broker Manager URL** from the **Locations** column for the **dsb-nginx** service.

If you have installed the Data Security Broker in a private VPC cluster, follow the instructions in the [Deployment models for Data Security Broker](#) section to fetch the Data Security Broker Manager URL.

To configure Data Security Broker Manager, perform the following steps

1. Copy the **load_balancer_url** from the Kubernetes dashboard for the IKS cluster or from the Openshift web console for the ROKS cluster.
2. Open a browser window and paste the **load_balancer_url** in the following format:

```
$ https://<load balancer url without the port number>
```

Example: If the LoadBalancer IP is <http://150.238.243.117:443/>, specify the IP in the format <https://150.238.243.117>

The warning **Your connection is not private** is displayed.

3. Click **Advanced**, and click the **Proceed to link** at the bottom of the page.
4. The **Getting Started** dialog to proceed with the configuration of the Data Security Broker Manager appears.
5. Configure the basic System Settings by entering the Init Password, Organization name, Domain name, and Proxy access, and click **Continue**.

- The Init Password field must contain the same password that you specified for the **secrets.initPass** parameter during the Data Security Broker Manager installation.
 - The domain name is part of the email, followed after the "@" character. For example, if the email specified is test@xyz.example.com, the domain name must be specified as **example.com**. The domain name that you enter must match in step 1 and step 2 or the domain name in step 2 can be a subset of the domain name specified in the step 1.
 - The proxy access name is the name that is responsible for the connectivity from Proxy to Data Security Broker Manager (the IP, or DNS, or the Service Name). For example, specify **dsb-nginx**, if you do not have any Proxy configured.
6. Create an Admin Account for the initial Data Security Broker Manager administrator by specifying the email address in the **Configure Super Admin User** page. This account is used to configure the subsequent components such as the keystore, data store connections, and Data Security Broker Shields. Click **Continue**.
 7. Once you complete the configuration process for the Data Security Broker Manager, the next step is to log in to the Data Security Broker Manager using the steps mentioned in the [Login to Data Security Broker Manager](#) section.

Log in to Data Security Broker Manager

Once you have configured the Data Security Broker Manager, the next step is log in to the Data Security Broker Manager using the OAuth token.

Complete the following steps to log in to the Data Security Broker Manager:

1. Generate an IBM OAuth token. To obtain the OAuth token, log in to your IBM Cloud® account from the Command Line Interface using the following steps:
 - a. Open the Command terminal from your system.
 - b. Execute the following command to log in to your IBM Cloud account:

```
$ ibmcloud login -sso
```

When prompted to open the one-time passcode in a browser window, specify **Y** for "Yes". You will be re-directed to the browser window, from where you can copy and paste the one-time passcode.

Navigate back to the CLI and paste the one-time passcode and select the account in which you wish to log in using your IBM Cloud credentials.



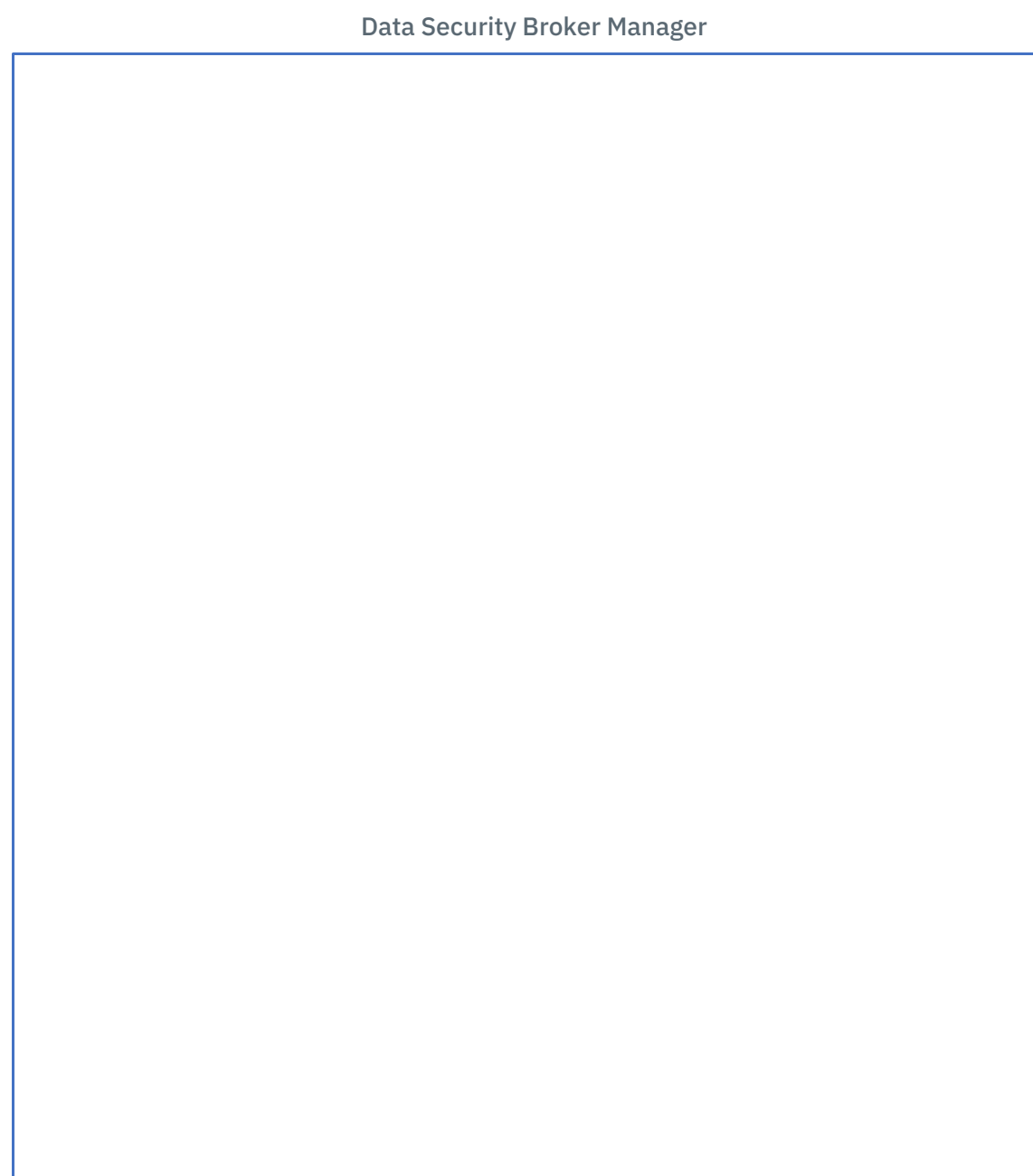
Note: Ensure that the IBM Cloud account, which is used during the configuration of the Data Security Broker Manager as the Super Admin user, and the IBM Cloud account that you are using to log in using the CLI to generate the OAuth token must be the same. Otherwise, you might get an error which says that the OAuth token is invalid.

2. Run the following command to generate the OAuth token:

```
$ ibmcloud iam oauth-tokens
```

⚠ Important: The **Admin** users that are added to Data Security Broker Manager must use the tokens generated in the same IBM Cloud® account, where the **Super Admin** user resides. Data Security Broker Manager does not support cross account access, considering the security risks involved.

3. Open the Data Security Broker Manager console which has already been configured in a browser window. For example, <https://150.238.243.117>.
4. Copy the OAuth token without any space or additional characters and paste it in the **Enter token** text box in the Login page and click **Sign In** to log in to the Data Security Broker Manager.



5. Once you have logged into the Data Security Broker Manager, the next step is add a database, connect to a keystore, and enroll an application to proceed with the data encryption. Refer to the [Data Encryption Services Overview](#) section to proceed with the data encryption services.

Setting up Data Security Broker Manager

Adding a Keystore, connecting to a Database and enrolling an Application

After you have completed configuring the Data Security Broker Manager, the next steps is to connect to a database, add a keystore and enroll an application to perform the data encryption.

1. Refer to the [Add a Database in Data Security Broker Manager](#) section to add a database in the Data Security Broker Manager.
2. Refer to the [Add a Keystore in Data Security Broker Manager](#) section to connect to a keystore in the Data Security Broker Manager.
3. Refer to the [Enrolling an Application in Data Security Broker Manager](#) section to enroll an application in the Data Security Broker Manager.

Connecting to a Datastore

To add and connect to a data store, complete the following steps in the Data Security Broker Manager:

1. Login to Data Security Broker Manager.
2. Select **Databases** from the left navigation and click **Add Database +** to add a data store.
3. In the **Add Database** dialog, enter a name and description for the database in the **Database Name** and **Database Description** fields.
4. Select the database type as **Postgres** from the **Database Type** drop-down list.


5. Enter the IP address of the database in the **Hostname or IP Address** field.
6. Specify the **Port** for the database and enter the user **Database Username** and **Database Credential**.
7. Create a new user on your database for use with IBM Cloud Security Broker. For more information, see [Database Privileges](#) section.
8. You can use only a **Postgres** database and enter your database name in the **Postgres Database Name** field.
9. **Optional:** Select **Use SSL**, click **Add file**, and upload an SSL Certificate.
10. Click **Add Database** to complete enrolment. The new database appears in the list of configured databases.
11. Create the data that is required for encryption or decryption as tables in the new database that is created.
12. Once you have completed adding a database, the next step is to connect to a keystore. Refer to the [Add a Keystore in Data Security Broker Manager](#) section to connect to a keystore in the Data Security Broker Manager.

Adding Keystore in Data Security Broker Manager

Before you can enroll your applications, add databases, and enable encryption, you must enroll your Keystore, so that the Data Security Broker Manager can access and create data encryption keys (DEKs) that is used to protect your data.

Follow the steps below to enroll a Keystore that you can use with Data Security Broker Shields and databases.

1. Login to Data Security Broker Manager.
2. Select **Keystores** from the left navigation and click **Add Keystore +**.
3. Specify a **Keystore name** and provide a valid **Description**.
4. Select the Keystore Type from the **Keystore Type** dropdown menu. Enter values for the **Instance ID**, **App Namespace**, **IBM Key Protect Alias**, and **IAM API Key**. Select the region in the **IBM Region** drop down and choose the Data Encryption Key (DEK) Storage type from the **DEK Storage Type** drop down. For information on how to create encryption keys, see [Creating and importing encryption keys](#).

 **Note:** Keystore parameters are specific to each Keystore type or vendor.

For more information on configuring the IBM Key Protect and configuring the IBM Cloud Hyper Protect Crypto Services (HPCS), refer to the [Configure IBM Key Protect and HPCS](#) section.

5. Click **Add Keystore** to create a Keystore.
6. Once you have completed adding a keystore, the next step is to enroll an application. Refer to the [Enrolling an Application in Data Security Broker Manager](#) section to enroll an application in the Data Security Broker Manager.

Enrolling an Application in Data Security Broker Manager

An application is the framework that links Data Security Broker Manager, databases, and Data Security Broker Shield and instructs the Data Security Broker Shield to encrypt and decrypt data.

A Data Security Broker Shield is limited to be enrolled with one application. But you can associate one application with multiple Data Security Broker Shields.

To enroll an application in Security Broker Manager, complete the following steps:


1. Log in to Data Security Broker Manager.
2. Click the **Applications** icon in the left navigation panel.
3. Click **Enroll Application +** in the upper right corner of the window. The **Enroll Application** dialog appears.
4. Enter an **Application Name** and **Application Description** in the respective fields.
5. Perform the following tasks:
 - Choose the Data Security Broker Shield from the drop-down list.
 - Select a **Data Store** for encryption.
 - Select the **Keystore** to be used as a source for data encryption keys.
 - Specify an **Encryption Method** as Column Level or Row Level.
6. Click **Enroll Application**. After the application is enrolled, it is displayed under the Applications in the Data Security Broker Manager.

The Shield Sync ID is used when you install Data Security Broker Shield. Ensure that you have the Shield Sync ID handy during the Data Security Broker Shield installation.

7. Once you have completed enrolling an application, you can proceed with the data encryption. Refer to the [Data Encryption using IBM Cloud](#)


[PostgreSQL Database](#) section to start protecting your data.

Install Data Security Broker Shield

 **Important:** Once you have completed setting up the Data Security Broker Manager, the next step is to install the Data Security Broker Shield. Copy the **Shield Sync ID** from the Data Security Broker Manager application, to use during the Data Security Broker Shield installation.

Complete the following steps to install the Data Security Broker Shield from the IBM Cloud® Catalog:

1. Click **Data Security Broker Shield** catalog item.
2. In the **Select your deployment target** drop down, select **IBM Cloud Kubernetes Service** or **Red Hat Openshift** to install the Data Security Broker Shield in the IKS or ROKS cluster.
3. The delivery method is selected as **HELM chart** by default under the **Select a delivery method** drop down.
4. Select the version of the software to install, in the **Select Version** drop-down.
5. From the list of available clusters, select the cluster on which you wish to perform the installation.
6. Select an existing namespace to deploy the Data Security Broker Shield or click **Add namespace** to add a new namespace. Specify the name for the namespace and click **Add** to create a new namespace within the selected cluster.
7. You must run the preinstallation script by clicking on the **Run Script** button after you select the cluster and the namespace. If you do not run the preinstallation script, you cannot proceed with the installation of the Data Security Broker Shield.
8. Configure your workspace by specifying the following details:
 - a. Specify the **Name** for the workspace. The workspace name must be unique and using the name of the workspace, you can manage, update or uninstall Data Security Broker Shield from the IBM Schematics Workspace (<https://cloud.ibm.com/schematics/workspaces>).

 **Note:** If you are installing more than one instance of Data Security Broker Manager in the same namespace, ensure that you provide unique workspace names to each of the Data Security Broker Manager installation. Otherwise, you might get error which describes that the same workspace already exists in the namespace.

- b. Select the **Resource group**, **Location**, and specify the **Tags** required for configuring the workspace.
9. Copy the Shield Sync ID from the Data Security Broker Manager application.
 10. Set the input variables. Provide the Shield Sync ID and Shield name and specify the values for the other required parameters as per the description.

If you are installing Data Security Broker Shield in a different namespace apart from the namespace, where Data Security Broker Manager is installed, specify values for the following input variables:

dsbShield.configMap.data.BM_IP: Enter a value in the format: **dsb-nginx.namespace name**.

dsbShield.secret.credstorePass: Enter the same value that you specified during the Data Security Broker Manager installation for the **secrets.credstorePass** input variable.

If you are installing Data Security Broker Shield in a different cluster apart from the cluster where Data Security Broker Manager is installed, specify values for the following input variables:

dsbShield.configMap.data.BM_IP: Replace **dsb-nginx** with the IP address of the Data Security Broker Manager.

11. Click **Install** in the **Summary** pane on the right to complete the installation process.
12. You will be navigated to the IBM Schematics Workspaces to track the installation progress. Once the installation is successful, a message **Workspace creation successful** is displayed.

 **Note:** If you get an error message like **Workspace creation failed**, refer to the Logs available in the Terraform output.

Next steps

Once you have successfully installed Data Security Broker Shield, you can refer to the [Data Encryption Services](#) section to encrypt data.

Deployment models

Deployment models for Data Security Broker

You can deploy Data Security Broker Manager and Data Security Broker Shield on a private Virtual Private Cloud (VPC) cluster.

Private VPC ROKS cluster

If you are planning your deployment on a private Red Hat® OpenShift® cluster, follow the steps below after installing Data Security Broker Manager and Data Security Broker Shield to get the public Data Security Broker Manager URL and a private Data Security Broker Shield URL.

Data Security Broker Manager on a private Red Hat OpenShift cluster

1. If you do not have access to the Red Hat OpenShift console to access the cluster, you can follow the two steps mentioned below to fetch the Data Security Broker Manager URL:

- a. Using the Virtual Machine (VM) user interface deployed within the VPC.

- Login to the Red Hat OpenShift cluster using ibmcloud shell, and execute the following command:

```
oc get routes dsb-manager -n <data_security_broker_manager_deployment_projectname>
```

where **data_security_broker_manager_deployment_projectname** is the project name that you created or selected during the Data Security Broker Manager installation.

- Login to the virtual machine and open the Data Security Broker Manager URL, that is obtained from the previous step.

- b. Create a public Load Balancer (LB).

- Login to the Red Hat OpenShift cluster using ibmcloud shell, and execute the following command:

```
$ ``sh {: codeblock}
oc get routes dsb-manager -n <data_security_broker_manager_deployment_projectname>
```

where **data_security_broker_manager_deployment_projectname** is the project name that you created or selected during the Data Security Broker Manager installation.

- Create a YAML file with the below format for the load balancer resource.

```
$ apiVersion: v1
kind: Service
metadata:
  labels:
    app: dsb-nginx
    name: dsb-nginx-public
spec:
  ports:
    - port: 443
      protocol: TCP
      targetPort: 8443
  selector:
    app: dsb-nginx
  type: LoadBalancer
```

- Execute the command to apply the YAML file:

```
oc apply -f <YAML> -n <data_security_broker_manager_deployment_projectname>
```

- Wait for the load balancer to get into **Active** state. This process might take from five to ten minutes.

- Fetch the load balancer URL by executing the command:

```
oc get svc dsb-nginx-public -n <data_security_broker_manager_deployment_projectname>
```

Data Security Broker Shield on a private Red Hat OpenShift cluster

1. After you install Data Security Broker Shield in a private Red Hat OpenShift cluster, by default, a public Load Balancer IP is provisioned.
2. If you require a private Load Balancer, you can use create a YAML file for the load balancer with the format mentioned below.

```
$ apiVersion: v1
kind: Service
metadata:
  annotations:
    service.kubernetes.io/ibm-load-balancer-cloud-provider-ip-type: "private"
  labels:
    app: <dsb-deployment-name>
    name: dsb-shield-private
```

```
spec:
  ports:
    - port: 8444
      protocol: TCP
      targetPort: 8444
  selector:
    app: <dsb-deployment-name>
  type: LoadBalancer
```

where **dsb-deployment-name** is the name of the Data Security Broker Shield deployment in your project. To get your **dsb-deployment-name**, execute the following command from your project, where you have installed Data Security Broker Shield.

```
helm list -n <project_name> | grep shield
```

3. Execute the command to apply the YAML file:

```
oc apply -f <YAML> -n <data_security_broker_manager_deployment_projectname>
```

4. Wait for the load balancer to get into **Active** state. This process might usually take from five to ten minutes.

5. Fetch the load balancer URL by executing the command:

```
oc get svc dsb-nginx-public -n <data_security_broker_manager_deployment_projectname>
```

Private VPC IKS cluster

Data Security Broker Manager on a private IBM Cloud® Kubernetes Service cluster

1. After you install Data Security Broker Manager in a private IBM Cloud Kubernetes Service cluster, by default, a public Load Balancer IP is provisioned.
2. If you require a private Load Balancer, you can use create a YAML file for the load balancer with the format mentioned below.

```
$ apiVersion: v1
kind: Service
metadata:
  annotations:
    service.kubernetes.io/ibm-load-balancer-cloud-provider-ip-type: "private"
  labels:
    app: dsb-nginx
    name: dsb-nginx-private
spec:
  ports:
    - port: 443
      protocol: TCP
      targetPort: 8443
  selector:
    app: dsb-nginx
  type: LoadBalancer
```

3. Execute the command to apply the YAML file:

```
kubectl apply -f <YAML> -n <data_security_broker_manager_deployment_name_of_the_namespace>
```

where **data_security_broker_manager_deployment_name_of_the_namespace** is the namespace name that you created or selected during the Data Security Broker Manager installation.

4. Wait for the load balancer to get into **Active** state. This process might usually take from five to ten minutes.

5. Fetch the load balancer URL by executing the command:

```
kubectl get svc dsb-nginx-private -n <data_security_broker_manager_deployment_name_of_the_namespace>
```

Data Security Broker Shield on a private IBM Cloud Kubernetes Service cluster

1. After you install Data Security Broker Shield in a private IBM Cloud Kubernetes Service cluster, by default, a public Load Balancer IP is provisioned.
2. If you require a private Load Balancer, you can use create a YAML file for the load balancer with the format mentioned below.

```
$ apiVersion: v1
kind: Service
metadata:
  annotations:
```



```
service.kubernetes.io/ibm-load-balancer-cloud-provider-ip-type: "private"
labels:
  app: <dsb-deployment-name>
  name: dsb-shield-private
spec:
  ports:
  - port: 8444
    protocol: TCP
    targetPort: 8444
  selector:
    app: <dsb-deployment-name>
    type: LoadBalancer
```

where **dsb-deployment-name** is the name of the Data Security Broker Shield deployment in your namespace. To get your **dsb-deployment-name**, execute the following command from your namespace, where you have installed Data Security Broker Shield.

```
helm list -n <namespace_name> | grep shield
```

3. Execute the command to apply the YAML file:

```
kubectl apply -f <YAML> -n <data_security_broker_shield_deployment_name_of_the_namespace>
```

where **data_security_broker_shield_deployment_name_of_the_namespace** is the namespace name that you created or selected during the Data Security Broker Shield installation.

4. Wait for the load balancer to get into **Active** state. This process might usually take from five to ten minutes.
5. Copy the load balancer by executing the command:

```
kubectl get svc dsb-nginx-private -n <data_security_broker_shield_deployment_name_of_the_namespace>
```

Data Security Broker Manager and Data Security Broker Shield in different or multiple private VPC clusters

If you are planning to install Data Security Broker Manager and Data Security Broker Shield in different or multiple private VPC clusters, you have two options:

1. Using VPC VPN connectivity: Refer to [Setting up VPC VPN connectivity](#) for more details on how to setup the VPC VPN connectivity.
2. Using Transit Gateway: Refer to [IBM Cloud Transit Gateway](#) for more details on how to use the transit gateway method.

Data Encryption Services

Data Encryption Services

Data Security Broker offers Data Encryption services which enables the provisioning of Data Security Broker Manager and Data Security Broker Shield. It also helps in configuring encryption or decryption rules against the IBM Cloud Databases such as PostgreSQL to encrypt and decrypt the database records or columns on the fly. It also helps in migrating the existing database records, apply record or column level encryption rules.

Data Security Broker supports two types of Data Encryption services. They are:

1. [Data Encryption](#)
2. [Data Masking](#)

Deployment Plans in IBM Cloud Data Security Broker

When you assign and customize default Data Protection Policies with Data Security Broker Manager, there are three options that you can choose to implement your data encryption policy:

Save Policy

Save Policy option is selected by default. This option saves your selected data, but does not execute encryption or data protection on your database. Your policy remains saved with the application until a new policy is saved to overwrite it. You can use the saved policy to deploy or migrate it later.

Deploy Policy

Deploy Policy option saves your policy and deploys your configured Data Security Broker Shield as a proxy for the configured database. You can connect to your Data Security Broker Shield endpoint, as if it was your database endpoint, to access your database. Any data that passes through your Data Security Broker Shield proxy is encrypted in the database, if that data is defined in your policy.

Deploy Policy and Migrate Data

Deploy Policy & Migrate Data option saves your policy, deploys it, and migrates your existing selected tables and columns through the specified Data Security Broker Shield to encrypt the data. Data Security Broker Shield acts as a proxy for the configured database, and the data is encrypted as well.



Note: This option is disabled for applications, which does not have a Data Security Broker Shield associated with it.

Encryption Models supported by Data Security Broker

Data Security Broker supports data encryption services that can be configured in four main modes.

Data Encryption

Data Security Broker functions as an application-level encryption (ALE) software in this mode for encrypting data on a field-level basis. This is performed using Data Security Broker Manager to enumerate the data schema and enable an encryption key mapping.

Data Tokenization

Data Security Broker supports length preserving and data type preserving tokenization method to anonymize data at the field level databases or in semi-structured data files.

Record Level Encryption

Data Security Broker can be configured for record level encryption to support multiple keys within a single column that are mapped to respective data owners or entities. This encryption mode can be used effectively in multi-tenant or shared data environments where segmenting of the data can be challenging. In this mode, data shredding can be enabled by deleting public keys and private keys for a respective entity.

Data Masking:

Data Security Broker can enable simplified data masking to prevent decryption of data and sensitive file information based on configuration or deleted keys. This mode can minimize data exposure in public cloud environment and provides a better control of data exfiltration to external parties.

Procedure

After you have completed setting up and configuring the Data Security Broker Manager, you can perform standard encryption or data masking by defining a Data Protection policy. Ensure that you complete the steps below before you can use the data encryption services offered by Data Security Broker.

1. You must add a Keystore, so that the Data Security Broker Manager can access and create data encryption keys (DEKs) that is used to protect your data. For more information, see [Adding a Keystore in Data Security Broker Manager](#).

2. Connect to a database in the Data Security Broker Manager. For more information, see [Connect to a Datastore in Data Security Broker Manager](#).
3. Enroll an Application in Data Security Broker Manager. For more information, see [Enrolling an application in Data Security Broker Manager](#).
4. Assign and customize a Default Data Protection Policy. For more information, see [Create, assign, and Customize Data Protection Policy in Data Security Broker Manager](#).
5. Encrypt and Decrypt Data. For more information, see [Encrypting the data with Data Security Broker on an IBM Cloud PostgreSQL Database](#).

Data Encryption using IBM Cloud Databases for PostgreSQL

Overview

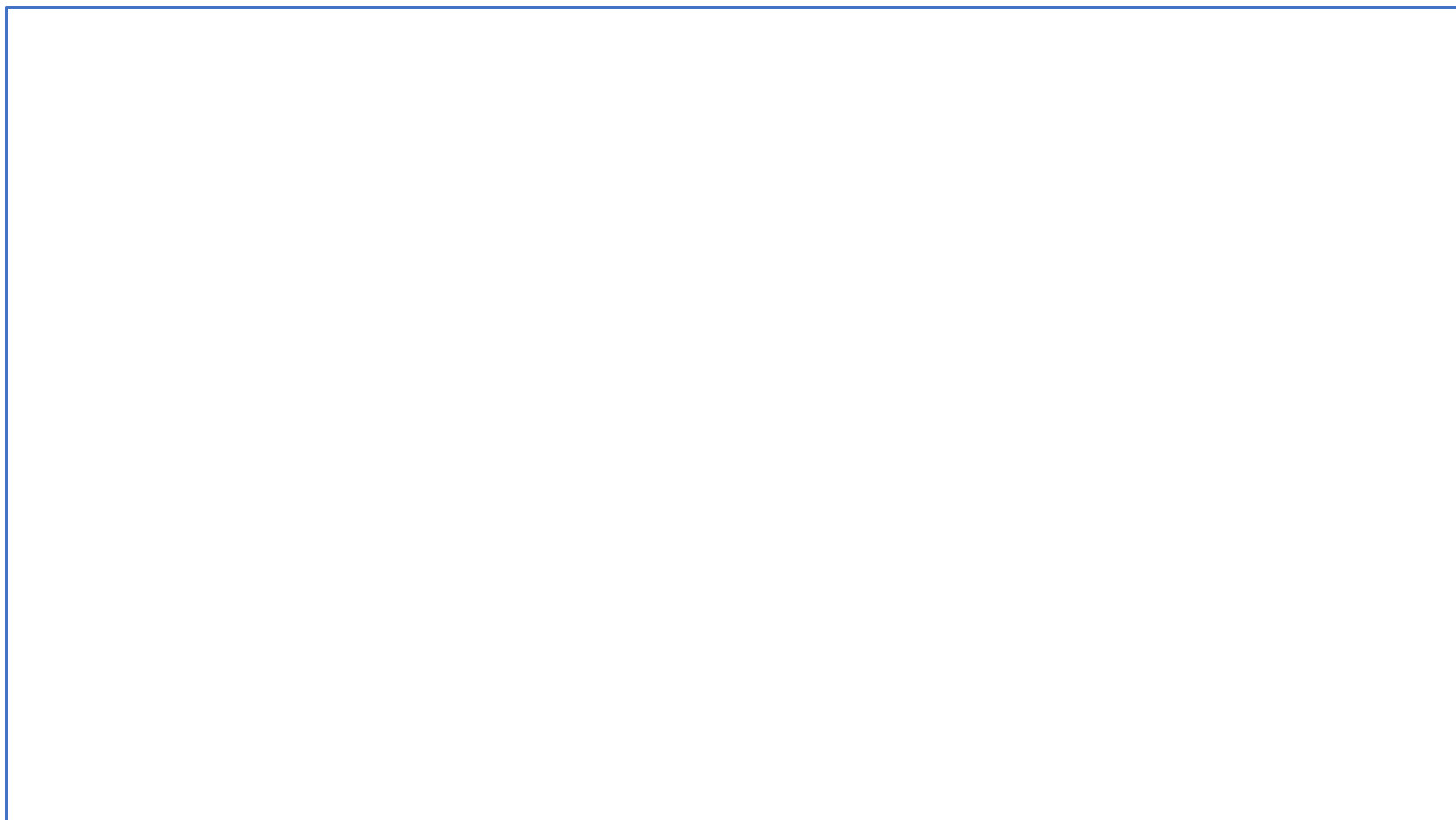
Data Security Broker functions as an application-level encryption (ALE) equivalent in this mode encrypting data on a field-level basis. This is performed using Data Security Broker Manager to enumerate the data schema and enable an encryption key mapping.

Procedure

Complete the following steps to encrypt the data with Data Security Broker Manager on an IBM Cloud PostgreSQL Database:

1. Login to Data Security Broker Manager.
2. Click on an application and select the drop down which is present in the **Migration Details** field in the right side and click **Encrypt**.
3. Select the Database and the table where you have the data created and select the **Column** which needs to be encrypted. Choose the **Data Protection** policy, **Encryption mode**, and **masking mode** for the encryption process and click **Review**.

Data Encryption



4. Choose **Deploy Policy & Migrate Data** under the **Deployment Plan** option. There are three options that you can choose to implement your data encryption policy. For more information on Deployment plans, see [Deployment Plans in Data Security Broker Manager](#). Select the Security Broker Shield service IP address in the **Migration Shield** field and click **Save** to start the encryption process.
5. The status of the application shows **Migrating** when the encryption process starts.
6. Once the encryption is complete, the status is changed to **Protected**. You can view more information by clicking **Migration Details** in the **Applications** sidebar.



Note: If there is new data which gets inserted in the database, by default, the data is encrypted by using the default data encryption policy that is being selected by the user.

Reference

Format Preserving Encryption (FPE) Supported Data Types

The following tables lists the FPE supported data types for the data encryption in Data Security Broker Manager:

PostgreSQL

Original Data Type	FPE Data Type
smallint	fpe-int
int	fpe-int
integer	fpe-int
bigint	fpe-int
bytea	fpe-int
numeric	fpe-decimal
decimal	fpe-decimal
numeric (s,p)	fpe-decimal
decimal (s,p)	fpe-decimal
money	fpe-decimal
var	<ul style="list-style-type: none"> fpe-decimal - fpe-alphanum - fpe-latin1
char	<ul style="list-style-type: none"> fpe-win1252 - fpe-cc
text	<ul style="list-style-type: none"> fpe-email1 - fpe-email2
date	fpe-datetime
time	fpe-datetime
timestamp	fpe-datetime
uuid	fpe-hexadecimal

Table 1. FPE Supported Data Types caption-side=bottom"

Counter-Mode (CTR) Supported Data Types



Note: Data Security Broker Shield only supports one word for a data type name. **BYTEA** is a PostgreSQL data type that has the capability to store hexadecimal data which is used to store encrypted data. **BYTEA** is an equivalent of **VARBINARY** in **MySQL** or **RAW** datatype in **Oracle** database.

PostgreSQL The following table lists PostgreSQL supported data types for M_CTR mode in Data Security Broker Manager.

Original Data Type	Encrypted Data Type
SMALLINT	BYTEA
INT, INTEGER	BYTEA
BIGINT	BYTEA

REAL, FLOAT4	BYTEA
FLOAT8 - Used in Data Security Broker Shield for "double precision"	BYTEA
DECIMAL, NUMERIC	BYTEA
VARCHAR - Used in Data Security Broker Shield for "character verification"	BYTEA
CHAR, CHARACTER, BPCHAR	BYTEA
TEXT	BYTEA
JSON, JSONB	BYTEA
BYTEA	BYTEA
MONEY	BYTEA
DATE	BYTEA
TIMESTAMP - Used in Data Security Broker Shield for "timestamp without time zone"	BYTEA
TIMESTAMPZ - Used in Data Security Broker Shield for "timestamp with time zone"	BYTEA
UUID	BYTEA
BIT	BYTEA
VARBIT - Used in Data Security Broker Shield for "bit verification"	BYTEA

Table 2. CTR Supported Data Types caption-side=bottom"

Data Migration using Data Security Broker Manager

Overview

Data Security Broker provides a data migration utility that assists with the offline encryption (and optionally, decryption) of existing data stored in supported databases. The utility accomplishes this by querying cleartext from the database and inserting the same data back into the database through Data Security Broker Shield to perform encryption.

You can add or select specific Data Security Broker Shield that is used for migration and set migration properties in Data Security Broker Manager.



Note: This option is only available when you select **Deploy Policy & Migrate Data** as the Data Protection Policy.

1. Log in to Data Security Broker Manager.
2. Select an application and click **Encrypt** to open the Encryption Schema Builder window.
3. Under the **Migration Shield** dropdown, select the Data Security Broker Shield you need to use for migration.



Note: Only the Shields enrolled with the application are available for selection.

4. Click **+Migration Shield**. The Add Data Security Broker Shield dialog appears. Make a note in the Shield description that this shield was added for migration.
5. Specify the necessary options and click Add Data Security Broker Shield.
6. Click Save to initiate your data protection policy. The migration process begins, and it takes place on your selected migration Shield. The new migration Shield is now visible on the Shields dashboard.

You can also modify the below Migration Properties for the Migration Plan:

- Batch Size - 2000, 50000, 100000
- Failure Scope - Server, Database, Table
- Parallel Processing - On or Off
- Clear Temp Tables - On or Off
- Migration Shield - only appears if there are multiple Data Security Broker Shields

Data Masking

About Data Masking

Overview

Data Security Broker Shield can mask the encrypted data and return the data in a user-friendly format without decryption or with partial decryption.

Data Masking converts data values into a predefined format that is typically irreversible. A Masking Mode defines the format used to mask selected data. You can create new Masking Modes and use them in a policy or map them to a column.

Creating a new masking mode

This section demonstrates how to create a new masking mode that is added to the Data Protection Library.

To create a masking mode, complete the following steps:

1. Go to the Data Protection Library and in the header, click the Plus (+) sign to begin creating a new mode.
2. Select **Masking Mode** from the drop-down list. The Masking Mode Builder dialog appears.
3. Enter a unique **Mask Mode Name** (30 characters or less) and optionally, a **Description** of up to 100 characters.
4. Select one of the following options from the Mask Mode drop-down menu: FIXED, PATTERN, NUMERIC, CHARACTER.
5. In the **Mask Format** field, enter the string to be used to mask the data.
6. Click **Save**. The new masking mode appears in the Data Protection Library, as well as a selection in the **Encryption Mode** drop-down.



Note: The compatible data types are automatically associated with the mode when you click Save. In this way, only the modes that are compatible with a column's data type are shown.

Viewing and editing a masking mode

This section demonstrates how to view and edit Masking Modes. However, editing is limited to modifying the name and description of the selected mode.

To view the details of a masking mode, complete the following steps:

1. Go to the Data Protection Library and click the carrot icon (>) next to **Masking Modes** to view the list of modes.
2. Select a mode from the list. The details for the mode appear in the window on the right.

To edit a masking mode, perform the following steps:

- a. Go to the Data Protection Library and click the carrot icon (>) next to **Masking Modes** to view the list of modes. The details for the mode appear in the window on the right.
 - b. Click the **Pencil** icon in the upper right corner to edit the mode.
 - c. Modify the **Mask Mode Name**.
 - d. Modify the **Description** for the mode.
3. Click **Save**.

To perform Data Masking, follow the steps below:

1. Refer to the [Configure RBAC Policy](#) section to configure a Role-Based Access Control (RBAC) policy.
2. Refer to the [Create RBAC Policy](#) section to create a RBAC policy.
3. Refer to the [Configure User groups](#) section to configure or add user groups.
4. Refer to the [Apply RBAC policy](#) section to a RBAC policy to user groups.
5. Perform a simple masking by following the instructions in the [Apply simple masking](#) section to a RBAC policy to user groups.

Configure RBAC for an application

Role-Based Access Control (RBAC) allows you to create policies that specify permissions for groups of users who access a Data Security Broker Shield. To configure RBAC, you need to add and manage user's client addresses, user IDs to defined groups. You need to map each user group to a permission (such as READ, READ_WRITE, or MASK) and a column in a table. In other words, RBAC policies determine **who can see what** data.

This RBAC configuration process is divided among four tasks:

1. Specify RBAC Configuration
2. Add User Groups
3. Create an RBAC policy
4. Apply an RBAC policy to columns

Specify RBAC configuration

To specify RBAC configuration for an application, complete the following steps:

1. In Data Security Broker Manager, navigate to **Applications**, select an application from the list, and click **Encrypt** to access the Encryption Schema Builder.
2. In the Data Protection Policies section, expand **RBAC** and select **RBAC Configuration**.
3. In the RBAC Configuration window on the right, click **Complete Configuration**.
4. In the window on the right, select a Mode from the drop-down list:
 - **Disable** switches off RBAC for the whole application. Definitions of policies and groups do not change but the shield does not use these policies. You can select this mode to start and define users and policies.
 - **Supported** switches on RBAC but does not require it for all connections to the Shield.
 - **Required** switches on RBAC and requires it for all sessions or transactions on the Shield. If the session or transaction is not found in a defined RBAC Policy, then that session is dropped by the Shield.
5. Select an option for User Determination from the drop-down list:
 - **SESSION** signifies that the users connecting to the Shield are determined by the session user ID, as defined in the specified User Groups.
 - **SQL_COMMENT_RAW** signifies that users connecting to the Shield are determined by user IDs specified in SQL comments. An SQL Comment Prefix field appears. This field defines the prefix string for SQL Comments containing the User ID. Here is an example SQL query with the appropriate comment:

```
$ select * from public.table1 *+ User:user_name *
```



Note: You can specify an arbitrary string as the SQL Comment Prefix, but the user ID must be defined in User Groups.

- **SQL_COMMENT_JWT** signifies users connecting to the Shield are determined by a JWT specified in SQL comments. An SQL Comment Prefix field appears. This field defines the refix string for SQL comments containing the JWT.

The JWT serves as an authenticator for the User Group. Individual User IDs are ignored in this RBAC mode; instead, the Role argument in the JWT must match a User Group Name, in order to determine group membership. The following options are available:

- a. **SQL Comment Prefix** -- for the prefix string for SQL Comments containing the JWT.
- b. **JWT TID** (optional) for consuming the tenant identifier.
- c. **JWT AUD** (optional) for consuming the audience identifier.
- d. **Key Value Pairs** (optional) to consume key-value pairs in a line-by-line format. Each key must be specified on a single line, in key:value format.
- e. **JWT secret key** is required field for an HS256 key.



Note: f. **JWKS Provider URL** is required field for a URL, for an RS256 key. At least one of the previous two fields is required. If you submit either a JWT secret key or JWKS Provider URL, you can proceed without submitting the other field. The Data Security Broker Shield determines which algorithm to use.

- g. **JWKS Cache Capacity** (optional) integer value - default is 10000. This field is only used with RS256 keys specified by the JWKS Provider URL.

6. Click **Save**. A confirmation dialog appears.



Note: When you update your RBAC configuration, all Data Security Broker Shields enrolled with the application must be restarted. This will cause a temporary outage of connectivity between the Data Security Broker Manager application and the database.

7. Click the **Yes, restart Data Security Broker Shields** checkbox and then select **Restart Shield**. The configuration is saved for the selected application.
8. Continue to configure user groups.



Note: You can edit your RBAC Configuration at any time, including the Mode and User settings.

Create an RBAC policy

The RBAC Policy Builder allows you to define Rules, which is, the permissions for user groups. One or more groups can be given READ, READ_WRITE, or MASK permissions. The group-to-permission mapping constitutes a single Rule, which in turn is added to the policy. Once a policy is established, you can apply it to the selected columns in a table, thereby determining **who** has access to **what** data.

To create an RBAC policy, complete the following steps:

1. In the Data Protection Policies section, expand **RBAC** and select **Policies**. On the RBAC Policy builder panel, click **Create RBAC Policy**. You also have the option for creating another user group before creating a policy.
2. In the Policy Builder panel, enter a **Policy Name** of 30 characters or less, and a unique **Description** of 100 characters or less.
3. From the Default Permissions drop-down, select one of the following:
 - a. **READ** assigns read-only permission as the default. The user sees clear data.
 - b. **READ_WRITE** assigns read and write permissions as the default. The user sees clear data and writes via Shield.
 - c. **MASK** assigns a data masking format as the default. You can see the masked data, according to the **Mask** mode specification that is selected.



Note: d. All Mask Modes are available for selection. The Default Permission is applied to any user or client connection who does not belong to a user group in the given RBAC policy.

4. Add a Rule to the Policy. Each Rule consists of two pieces:



Note: User Groups and Permission. The individual Rule maps a Permission to one or more User Groups. A policy can have an unlimited number of rules, or no rules. An RBAC Policy with no rules, containing only the Default Permission, can be used to apply simple data masking for all Users which connect to the Shield.

- a. Enter a **Rule Name** of 30 characters or less.
- b. Select relevant User Groups from the drop-down list. All User Groups are available, but only one Rule will be considered per User. See step 5.
- c. Select a **Permission** option from the drop-down: **READ**, **READ_WRITE**, or **MASK**.



Note: MASK permission allows you to specify an existing Masking mode for a given user group. In other words, upon accessing the data, members of that group view the data in the mask format that you specify.

5. RBAC policy rules must be arranged in a hierarchy. Reorder the rules by clicking the Up and Down arrows to update the Rank number in the policy.



Note: Technically, an individual database user can have membership to more than one user group. However, this exposes a potential contradiction where multiple rules could apply to an individual database user. To avoid this, Rules must be arranged in a hierarchy, where only the highest-ranked rule is considered for each user group, and each user.

6. **Optional:** Create another rule by clicking **+ Rule**. Rules can be collapsed, expanded and deleted. When collapsed, only the Rule Name and Rank are visible.
7. When you are done with the policy, click **Save**. Role-Based Access Control is fully configured, and ready to apply to columns.
8. **Optional:** After saving the policy, create another User Group or create another RBAC Policy.
9. Continue with applying an RBAC policy to the database columns.



Note: Once you edit the RBAC configuration, the Data Security Broker Shield is restarted and this might result in a brief outage of connectivity between the database and the application.

Configure user groups

User Groups allow you to define groups of database-level users and permitted IP addresses, based on the desired role and permission level. User groups are mapped to permissions to create an RBAC policy. User groups are also shared across all applications on your Data Security Broker Manager. If a group's membership is modified, the change is persisted in every policy and application where the group is selected.

Note:

- The Users list can be changed at any time. You are simply changing the membership of a given User Group. There is no impact on the rest of the RBAC configuration, other user groups, or policies.
- Allowed Subnet(s) can be changed at any time. You are changing the allowed subnets for a given User Group. There is no impact on the rest of the RBAC configuration, other user groups, or policies.

To configure RBAC user groups, perform the following:

1. In the Data Protection Policies section, expand **RBAC** and select **User Groups**. Click **Create User Groups**.
2. In the User Groups panel, enter a **User Group Name** of 30 characters or less, and a unique **Description** of 100 characters or less.
3. **Optional:** Select the **Global** checkbox to create a group where individual Users are ignored.

Anyone connecting to the Shield has membership to this Global group, provided their IP address is within the **Allowed Subnets** range. If you select Global, proceed with step 5.

4. Enter usernames as a CSV-formatted list in the **Users** text box.

This is the list of client IDs which Data Security Broker Shield will consider when the User Determination is set to either **SESSION** or **SQL_COMMENT_RAW**.

5. **Optional:** Enter a range of permitted IP addresses as **Allowed Subnets**. When subnets are configured, a user's membership depends on the following conditions:
 - a. User is specified in the Users box (User Determination is either SESSION or SQL_COMMENT_RAW) or User is specified by the JWT (User Determination is SQL_COMMENT_JWT) and User has an IP address in the permitted range.

Multiple IP address ranges can be specified for one group, separated by commas. Spaces between IP addresses are fine.

Note: If **Allowed Subnets** is empty or 0.0.0.0/0 is specified, then **ALL IPs** are permitted for the user group.

6. Click **Save**.
7. Perform one of the following actions:
 - Continue to create an RBAC Policy.
 - Create another user group.

Apply an RBAC policy to columns

This task demonstrates how to select an RBAC policy for an application and apply it to selected columns in a table. You may optionally select an encryption mode on the same columns, to transform the underlying data.

Note:

- Only one RBAC policy can be applied to a column.
- RBAC policies are application-specific and cannot be shared with other applications.
- RBAC policies have a dependency on data types, because only certain Mask Modes can be applied for certain data types. In the Column selector, only compatible RBAC Policies are available to select for each column.
- Editing an RBAC policy affects all columns to which the policy is already applied.

To apply an RBAC policy to columns, do the following :

1. In Data Security Broker Manager, navigate to the **Applications** dashboard, select an application from the listing, and click **Encrypt** to access the Schema Builder.
2. In the Tree Menu, select a database, schema, and table. This populates the column selector with available columns.
3. Select columns to define with an RBAC policy.
4. In the Data Protection dropdown menu, select an RBAC policy from the list. RBAC policies have a dependency on data types, because only certain Mask Modes can be applied for certain data types. Therefore, in the Column selector, only compatible RBAC Policies are available to select for each column.

5. By default, the standard Encryption mode 'DEFAULT_CTR_DET' is selected.

When an RBAC Policy and encryption mode is selected on the same column, then the underlying database is encrypted as well. To apply an RBAC policy without encrypting, use the **Clear Selections** option, then select the RBAC policy.

6. **Optional:** Specify a Key ID for each column from the drop-down menu or accept the default.
7. Click **REVIEW** at the bottom left panel to review your selections.
8. Review your policy and select a Migration Plan:
9. **Save:** Saves the data schema for future use.
10. **Deploy:** Establishes a data schema for an environment that was not processed through Data Security Broker migration and data type conversion but does not migrate the data.
11. **Deploy and Migrate:** Establishes a data schema *and* transforms the existing data in the data store.

NOTE: If you have only configured RBAC Policies, without specifying encryption, then the **Deploy and Migrate** option is not available. This is because RBAC does not transform the underlying data.

Apply simple masking on clear data

This section describes a simple RBAC policy to apply basic data masking for all users. Follow the steps below to apply masking on the data using Data Security Broker Manager.

Step 1: RBAC configuration:

Configure RBAC for all connections to Shield, by setting the Mode as Required. Also, set the User Determination as Session, to reference the DB session user ID for User Group membership. However, when configuring a global RBAC policy, the User Determination is not relevant.

Step 2: User group - All-Users:

You can create a Global user group called "All-Users". Select the Global checkbox and do not specify any Allowed Subnets. (0.0.0.0/0 is automatically filled when this text box is left blank.) This User Group applies to all possible connections to Shield.

Step 3: RBAC policy - Default Masking:

You can create two Masking policies which are nearly identical. The only difference is that one policy applies for VARCHAR columns, and the other applies for INT columns. The parameters are as follows:

The first policy, "Default Varchar Masking", has one Rule to use **MASK_VARCHAR** for the **All-Users** user group. The second policy, "Default Int Masking", has one Rule to use **MASK_INT** for the **All-Users** user group.

Because both rules contain the global **All-Users** group, the default permission is irrelevant in this case.

Step 4: Application of Masking policies on data columns:

Finally, select a number of columns to mask. For each column, remove the DEFAULT_CTR_DET selection and add one or the other RBAC policy based on the Data Type.

On the Confirmation page, review the data protection schema. Because there is no Encryption defined, and no data transformation taking place, the option for **Deploy & Migrate** is unavailable.

Deploy the policy. If you connect to the Shield proxy, you can see that the Data Masking has been applied to the selected columns.

Configurations in Data Security Broker Manager

Assigning or changing a Default Data Protection Policy

Overview

A Data Protection Policy consists of a combination of encryption modes and masking rules. Default Data Protection Policies allow you to apply default settings for encryption, Format Preserving Encryption (FPE), or Data Masking. A policy is applied to the columns of a data store associated with an application enrolled in Data Security Broker Manager and linked to a Data Security Broker Shield.

A list of Default Data Protection Policies is available in the Data Protection Library at the bottom of the left navigation bar. Click the carrot icon (<) to expand and view the list of **Default Data Protection Policies, Encryption Modes, and Masking Modes**.

To assign or change a Default Data Protection Policy, do the following

1. In Data Security Broker Manager, click the **Application** icon in the left navigation menu bar.
2. Select an application from the list and click **Encrypt**.
3. In the left navigation menu, navigate to a database and schema and select the table for which you want to assign or change a policy. The table appears on the right.
4. Click the checkbox for a column and then expand the **Data Protection** drop-down list. By default, CTR Policy is selected.
5. Perform one of the following operations:
 - Accept the default policy or select a different default policy.
 - Specify a new default policy and add an encryption or masking mode.

Default Policy rules are **overwritten** by individual mode selections. However, because the Default FPE policy contains only encryption modes, and the Default Masking Policy contains only Masking modes, selecting the opposite mode will not affect the default policy. For example:

- If you select Default FPE and any Mask Mode, both are applied to that column.
 - If you select Default FPE and any other FPE mode, (like FPE-cc), the FPE-cc option **overwrites the** default FPE option for that column.
6. Select a **Key ID** from the drop-down menu or accept the default.
 7. Click **REVIEW** at the bottom of the left panel to review your selections.
 8. Review your policy and select a **Migration Plan**. The migration plans are listed below:
 - **Save Policy**: Defines a policy for future migration.
 - **Deploy Policy**: Establishes a data schema for an environment that was not processed through Data Security Broker migration and the data type conversion but does not migrate the data.
 - **Deploy Policy & Migrate Data**: Defines a policy and migrates the existing data in the data store.

Adding users

Overview

Data Security Broker Manager supports multiple users who can access the Manager console. All users on Data Security Broker Manager automatically have administrative privileges, so that they can enroll databases and applications, modify them, and deploy policies. However, only a user with Super Admin privileges can add, invite, and delete new users. A Data Security Broker Manager instance is limited to have only one Super Admin.

Adding a user to the Data Security Broker Manager

Complete the following steps to add a new user to the Data Security Broker Manager:

1. As a Super Admin, navigate to the **Users** tab in the left navigation menu. Click **Add Users +** in the top right corner. Enter an email address to add the user to the account.



Note: The user that you are trying to invite to the Data Security Broker Manager instance must be part of the same cloud account where the Data Security Broker Manager is installed.

Settings

Overview

You can use the Settings page to change the basic system settings in the Data Security Broker Manager.

1. Log into the Data Security Broker Manager.
2. Navigate to the **Settings** icon in the left navigation
3. Click the **Edit** icon to change the system settings. Modify the **Organization name**, and **domain settings**, and click **Save**.

HTTPS Certificate

HTTPS Certificate allows you to enable SSL for the Data Security Broker Manager console. This ensures that you have secure access to the Data Security Broker Manager web interface.

When you try to log into the Data Security Broker Manager after the set up, you get a warning: **Untrusted certificate browser warning**.



Note: The warning appears when you have not uploaded an HTTPS certificate. HTTPS certificate is a digital certificate that authenticates a website's identity and enables an encrypted connection.

Upload an HTTPS certificate by following the instructions in [Adding HTTPS certificate](#) section.

Logger

Complete the following steps to configure the Logger settings for the Data Security Broker Manager:

1. In the **Settings** page, click **Logger +** to configure a new logger.
2. Specify a name for the logger according to Java logger naming convention. A dot is used to separate names, that enforces a hierarchy.
3. Select a Log Level from the dropdown. These levels define the severity and granularity of the events which is captured by this logger. The options include: a. TRACE (Fine-tuned debugging messages) b. DEBUG (General debugging messages) c. INFO (Informational events) d. WARN (Events that might lead to an error) e. ERROR (Error messages)
4. Click Save to complete the **Logger** configuraion.

Database Privileges

In this section, you can find the details about minimum required database privileges for encryption using Data Security Broker.

Database privileges for encryption and migration

To carry out encryption and migration, Data Security Broker Shield requires certain user permissions on the database. It is recommended that you create a new user on your database for Data Security Broker Shield to use.

Database privileges required for PostgreSQL 12

Operation	Details	Queries used by Shield	Minimum required grants
Proxy normal operation: Support implicit inserts, and Obtain column information	Access information_schema to get information about columns. This is needed to support queries such as implicit inserts (inserts that don't have column names specified explicitly). [Per database or per schema or per table that is defined in Data Security Broker] select ordinal_position, column_name, data_type from information_schema .COLUMNS where table_catalog=Database Name and table_schema=Schema Name, and table_name=TableName order by ordinal_position	Select grant is required for all tables that are defined in Data Security Broker.	If a new database, schema or column is added to your protection plan, ensure the grant is applied
CheckProxyPort	Check if the port specified for the Shield is responsive	Select 1	Select grant


Table 1. Database privileges required for PostgreSQL 12 in Data Security Broker Manager caption-side=bottom"

Update Database Certificate

You need to update the database certificate in Data Security Broker Manager before it expires. Ensure that you keep a track on the database certificate expiry date. If the database certificate expires, the connection between the application and the database is lost.

To update the database certificate, complete the following steps in the Data Security Broker Manager:

1. Click **Databases** from the left navigation and select a database.
2. Click **Update** in the Database Summary pane on the right to update the Database Certificate.

 **Note:** Once you update the database certificate, the Data Security Broker Shield is restarted and this might result in a brief outage of connectivity between the database and the application.

Adding HTTPs certificate

To add or update an HTTPS certificate, follow the steps below:

1. Copy the certificate files (certificate and key) into a specific directory.
2. Find the pod name of the Data Security Broker Manager using the command in the Kubernetes cluster:

```
kubectl get pods
```

3. Log in to the pod using the command:

```
kubectl exec -it <pod_name> -- sh
```

4. Navigate to the directory, where you have copied the certificate file, for example, /opt/baffle/dsb-manager and check if the SSL directory is present. If the directory is not present, then create a directory with the certificate files.
5. Exit the pod.
6. Copy certificate and key file to the pod using command:

```
kubectl cp <source_file> <pod_name>:/opt/baffle/baffle-manager/ssl
```

7. Once the certificate files are copied, select the **Refresh** button in the **HTTPS Certificate** section in the **Settings** page of Data Security Broker Manager.

Key Management Services

Configure IBM Key Protect or IBM Cloud Hyper Protect Crypto Services (HPCS) or add a Local Keystore

In this section, you can find details on how to configure IBM Key Protect and IBM Cloud Hyper Protect Crypto Services (HPCS) and add it as a keystore in Data Security Broker Manager.

Refer to the [Configuring and adding IBM Key Protect as a Keystore](#) section to configure IBM Key Protect and add it as a Keystore in Data Security Broker Manager.

Refer to the [Configuring and adding HPCS as a Keystore](#) section to configure IBM Key Protect and add it as a Keystore in Data Security Broker Manager.

Refer to the [Adding a local Keystore](#) section to add a Local Keystore in Data Security Broker Manager.

IBM Key Protect, HPCS, and Local Keystore

Configure IBM Key Protect

In this section, you can find details on how to configure IBM Key Protect and add it as a keystore in Data Security Broker Manager.

Overview

The [IBM® Key Protect](#) service enables you to provision and store encrypted keys across your IBM Cloud environment. IBM Key Protect provides full encryption visibility and control, allowing you to see and manage data encryption and the entire key lifecycle from a single location.

Procedure

Complete the following steps to configure IBM Key Protect. After completing this procedure, you can add IBM Key Protect as a keystore in Data Security Broker Manager.

To configure IBM Key Protect, do the following:

1. Get an IBM Instance ID in one of the following ways:
 - Using the IBM CLI -- enter the following command in a shell window:
ibmcloud resource service-instance 'Key Protect-dsb-1'
 - Using the IBM Cloud web console -- navigate to **Services and software** and select the Key Protect instance. The GUID is displayed in the sidebar, and is what you use for the instanceID.
2. Create and retrieve API Key, in the following way:
 - Open a web browser and navigate to: <https://cloud.ibm.com/iam/apikeys>.
 - Select **Create an IBM Cloud API Key** and name the key.
 - Copy or download the key after it's created.
3. Create Cloud Object Storage in the resource group.
4. Create a Bucket in the COS instance, for example: bm-ibm-bucket
5. Generate Service Credentials for the COS Bucket.
6. In Object Storage, click on ****Service Credentials**** in the side panel. Then click **New Credential**.
7. Enable HMAC and click **Add**.

Add IBM Key Protect as a keystore in Data Security Broker Manager

Complete the following steps to complete the process for using IBM Key Protect as a keystore in Data Security Broker Manager.

To add IBM Key Protect as a keystore, perform the following steps:

1. In the Data Security Broker Manager console, click the key icon in the left navigation bar. The Keystore window appears.** **
2. Click **+Keystore**. The Add Keystore dialog appears.
3. Enter a keystore Name and select ****IBM Key Protect**** in the Keystore Type drop-down menu.
4. Enter the Instance ID, that you have created for the **KeyProtect Instance**.

5. For **App Namespace**, enter a string to identify the application.
6. For **IBM Key Protect Alias**, enter a unique string value.
7. For the **IAM API Key**, use the key that you have created.
8. For **IBM region**, specify the region for the IBM Key Protect instance.
9. For the IBM Cloud Object Storage URL, specify the IBM endpoint URL for COS, for example: <https://s3.us-south.cloud-object-storage.appdomain.cloud>
10. Enter the **cos_hmac_keys** for the Access Key ID and Secret Key.
11. Click **Add Keystore**. Note: The key will not appear in IBM Key Protect until Data Security Broker Shield has been connected and data encryption migration has occurred.

Configure IBM Cloud Hyper Protect Crypto Services (HPCS)

In this section, you can find details on how to configure HPCS and add [IBM Cloud Hyper Protect Crypto Services \(HPCS\)](#) as a keystore in Data Security Broker Manager.

Overview

[IBM Cloud Hyper Protect Crypto Services](#) offers a cloud hardware security module (HSM) and key management service. It aims to give you control over your cloud hardware security models and cloud data encryption keys as it is the only service in the market built on FIPS 140-2 Level 4-certified hardware.

The HPCS service, which is based on IBM LinuxONE technology, helps to guarantee that only you have access to your keys. Using a dedicated customer-controlled HSM that provides single-tenant key management and key vaulting makes it simple to create encryption keys. You can also bring your own encryption keys to manage instead.

Configure HPCS instance:

Configure IBM Cloud Hyper Protect Crypto Services by following the steps below.

1. Get an IBM HPCS Instance ID in one of the following ways:
 - Using the IBM CLI – Login to your IBM CLI and execute the following command:

```
ibmcloud resource service-instance 'HPCS-DSB-1'
```

Note: If you are using the IBM Cloud web console, navigate to **Services and Software** and select the HPCS instance. The **GUID** is displayed in the sidebar and the **instanceID** is the same as the **GUID**. 2. Create and retrieve API Key, in the following way: a) Open a web browser and navigate to <https://cloud.ibm.com/iam/apikeys>. b) Select **Create an IBM Cloud API Key** and provide a name for the key. c) Copy or download the key after it is created. 3. Create Cloud Object Storage in the resource group. 4. Create a Bucket in the COS instance, for example: **dsb-ibm-bucket** 5. Generate Service Credentials for the COS Bucket. 6. In Object Storage, click on **Service Credentials** in the side panel. Click **New Credential**. 7. Enable **HMAC** and click **Add**. 8. The next step is to add the IBM Cloud Hyper Protect Crypto Services as a keystore in Data Security Broker Manager.

Add IBM Cloud Hyper Protect Crypto Services as a keystore in Data Security Broker Manager

You can add the IBM Cloud Hyper Protect Crypto Services instance as a keystore in Data Security Broker Manager by completing the following steps.

1. Log into Data Security Broker Manager.
2. Select **Keystores** from the left navigation and click **Add Keystore +**.
3. Specify a name for the Keystore in the **Keystore name** field and provide a valid description in the **Description** field and select **IBM Cloud Hyper Protect Crypto Services** in the **Keystore Type** drop-down list.
4. Enter the Instance ID for the HPCS KeyProtect Instance, which is obtained from the Step 1 in configuring the HPCS instance.
5. For App Namespace, enter a string to identify the application.
6. For IBM Cloud Hyper Protect Crypto Services Alias, get the Alias name of the Key which has been created in the HPCS instance.
7. For the IAM API Key, use the key you created using Step 2 in Configuring the HPCS section.
8. For IBM region, specify the region for the IBM Cloud Hyper Protect Crypto Services instance.
9. For the IBM Cloud Object Storage URL, specify the IBM endpoint URL for COS, for example: <https://s3.us-south.cloud-object-storage.appdomain.cloud>
10. Enter the **cos_hmac_keys** for the Access Key ID and Secret Key.
11. Click **Add Keystore**.

Local Keystore

In this section, you can find details on how to configure local keystore in Data Security Broker Manager.

Procedure

Complete the following steps to add a local keystore in Data Security Broker Manager.

1. In the Data Security Broker Manager console, select **Keystores** from the left navigation and click **Add Keystore +**.
2. Enter a Keystore Name of up to 30 characters and select **LOCAL** in the **Keystore Type** drop-down menu.
3. Enter the Data Security Broker Secret Key in the text field. The Data Security Broker secret Key must contain at least 10 characters, a mixture of upper and lower case, including at least 1 number.
4. Click **Add Keystore**.

Key rotation using IBM Key Protect and HPCS

Rotate Keys for IBM Key Protect or IBM Cloud Hyper Protect Crypto Services (HPCS) in Data Security Broker Manager

In this section, you can find details on how to rotate the Master and Data Encryption Keys for IBM Key Protect or HPCS Keystore in Data Security Broker Manager.

1. Log in to Data Security Broker Manager.
2. Select **Keystores** from the left navigation and click on one of the Keystore.
3. In the right navigation, select the Master Key and click on the three dots. Click **Rotate Key**
4. Perform the same operation for the Data Encryption Key (DEK) for the same keystore.
5. Once the Key rotation operation is successful, you will get a message saying "Master Key was successfully rotated" for the Master Key. For the Data Encryption Key, you will get a message saying "KeyID (ID value) was successfully rotated to KeyID (ID value), if the key rotation is successful.



Note: If key rotation is performed on an encrypted column, all other columns that have a referential relationship with the encrypted column must have their keys rotated to the same new key as well.

Upgrade and Uninstall

Upgrading using IBM Cloud Catalog

After Data Security Broker is installed on your cluster, you can upgrade it at any time using the IBM Cloud Schematics workspace (<https://cloud.ibm.com/schematics/workspaces>)

Pre-requisite:

The user must be aware of the workspace name which is provided during the Data Security Broker Manager and Data Security Broker Shield installation process.

Upgrading Data Security Broker Manager:

Log into IBM Cloud Schematics workspace and follow the steps below to upgrade the Data Security Broker Manager version:

1. Search for the workspace name that you provided during the Data Security Broker Manager install and click on the workspace to open it.
2. Click **Settings** option in the left navigation.
3. If the new version is available for the Data Security Broker Manager, then the **Update** button is enabled and you can click the **Update** button to proceed with upgrading the Data Security Broker Manager.
4. In the **Update Workspace resources** window, select the version of the Data Security Broker Manager, which you wish to upgrade to, and click **Update**.



Note: Once you click **Update**, IBM Cloud Schematics runs the terraform code in the backend to execute the newer version of the Data Security Broker Manager from the IBM Cloud Helm Catalog. After the upgrade is complete, the pods are refreshed in the namespace where you have upgraded the Data Security Broker Manager.

5. Follow the same process to proceed with upgrading the Data Security Broker Shield. Remember to work with the correct workspace name, which is provided during the Data Security Broker Shield install.

Note: Once the Data Security Broker Shield upgrade completes, based on the prior Shield Sync ID, new Data Security Broker Shield instances is automatically picked up by the enrolled application in Data Security Broker Manager.

Once the upgrade operation is successful, the pods will come to **Running** state in the target cluster.

Uninstalling through IBM Cloud Catalog

If you no longer need to use Data Security Broker, you can uninstall all the workloads that are associated with the Data Security Broker using the [IBM Cloud Schematics workspace](#).

Pre-requisite

The user must be aware of the workspace name which is provided during the Data Security Broker Manager and Data Security Broker Shield installation process.

Uninstalling Data Security Broker Manager

Log into IBM Cloud Schematics workspace and follow the steps below to uninstall the Data Security Broker Manager:

1. Search for the workspace name that you provided during the Data Security Broker Manager install and click on the workspace to open it.
2. Select **Actions** -> **Destroy Resources** to destroy the workloads associated with the Data Security Broker Manager.
3. Follow the same process for the Data Security Broker Shield to uninstall the workloads associated with the Data Security Broker Shield. Remember to work with the correct workspace name, which is provided during the Data Security Broker Shield install.

Once you uninstall the Data Security Broker Manager and Data Security Broker Shield workloads, you can see that the pods are being terminated and you can monitor the successful uninstall operation through the logs.

Data Security Broker Shield Performance

Data Security Broker Shield is an SQL proxy that is placed between the application and the database.

The network latency between Data Security Broker Shield and the application and between Data Security Broker Shield and the database should be as low as possible, when compared with the original latency between the application and database. The network bandwidth between Data Security Broker Shield and the application and database should also match the original network bandwidth available between the application and database.

One way to minimize latency between the application and Data Security Broker Shield is to directly deploy Data Security Broker Shield on the application host, in cases where the number of application servers are small and where sufficient CPU and memory capacity are available on each application server, and this deployment model might be a desirable approach.

In cases where there is a high concurrent workload, a single Data Security Broker Shield instance might become a bottleneck. In these situations, Data Security Broker recommends deployment of multiple Data Security Broker Shields behind a load balancer. Each Data Security Broker Shield instance is designed to handle hundreds of concurrent connections, but the actual number of connections that each Data Security Broker Shield can reasonably support depends on the resource availability of the machine.

Goal	Deployment	Use case
Minimize latency between the application and Baffle Shield	DSB Shield on the application host	Small number of application servers and Sufficient CPU and memory capacity per application server
Prevent Baffle Shield from becoming bottleneck	Multiple DSB Shields behind a load balancer	High concurrent workload and Resourced machines

Table 1. Data Security Broker Shield Performance

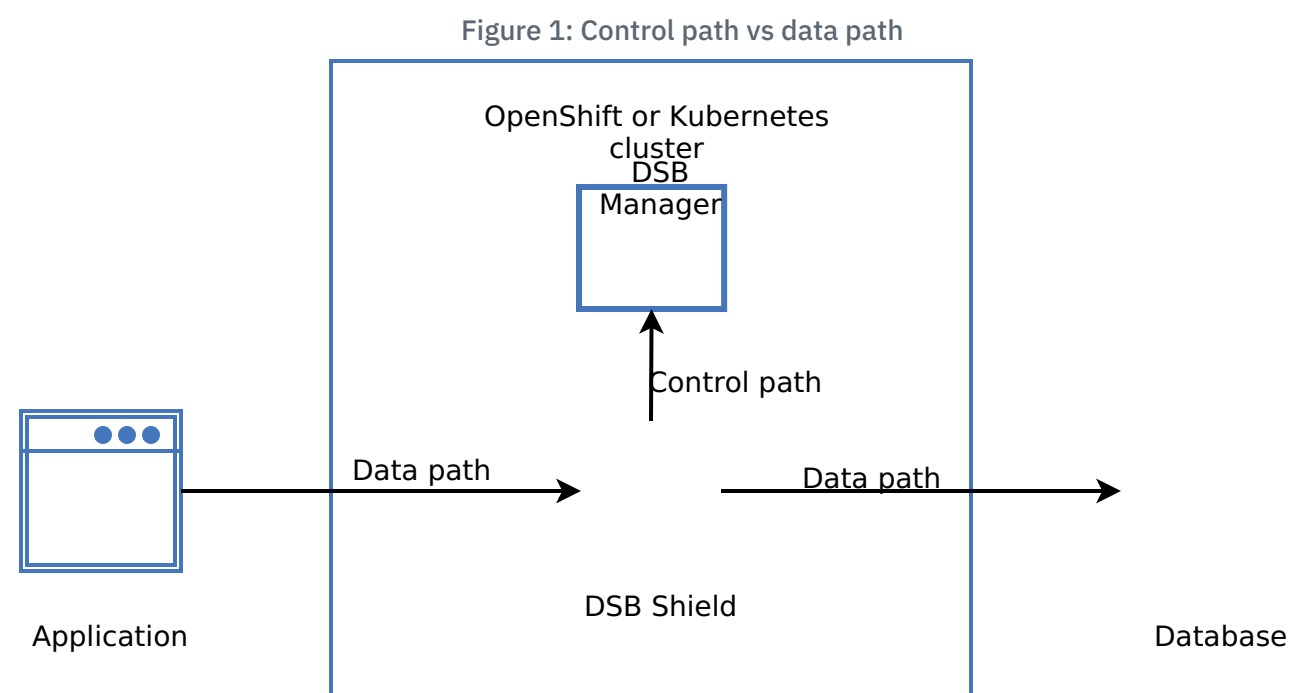
High availability and disaster recovery

Understanding High Availability for IBM Cloud Data Security Broker

Data Security Broker components can be deployed in an environment with highly available configuration to minimize or eliminate the downtime caused by momentary failures of Data Security Broker components or the infrastructure they depend on, ensuring the continuous availability of data protected by Data Security Broker Data encryption services. The high volume of data access during peak load periods that can result from deployment in a highly available configuration can also help mitigate service degradations.

Overview of High Availability Architecture

A management server connects to the proxies and manages their configuration and settings, and proxies are placed in the data paths between applications and data stores as part of the Data Security Broker Data Encryption service. Figure 1 shows the control path, which is made up of the management servers and the connections between the proxies and the management servers.



The data path's proxies can function without maintaining a constant connection to the management server. Furthermore, deployments rarely involve changing configuration and policy settings. With a much stricter availability requirement for the proxies than the management server, this means that the availability requirement for data path components and control path components can differ.

Data Security Broker typically advises using Data Security Broker Shields to achieve consistent availability with the database it is protecting. Shields must typically be installed in as many regions and availability zones as the database. To take advantage of the dynamic routing and scaling features provided by the framework, Data Security Broker also advises customers to deploy Data Security Broker components in a Kubernetes or Red Hat OpenShift environment.

Data Security Broker suggests maintaining a standby Data Security Broker Manager instance with regular metadata synchronization for organizations that needs to shorten the time it takes to restore access to a Data Security Broker Manager instance.

Deploying Data Security Broker Shields for High Availability

Data Security Broker recommends deploying in Kubernetes or OpenShift clusters for use cases that demand high availability. You can find details on the Data Security Broker Shields deployments in Kubernetes or Red Hat OpenShift environment.

Deploy across regions

Every Data Security Broker Shields is typically deployed in a cluster as its own pod. It is not necessary for the Shield to have a constant connection to Data Security Broker Manager. To retrieve the configuration and data security policy information and to update configuration or policies, the Shields are needed to communicate with Data Security Broker Manager during initial registration. This helps to deploy the Data Security Broker Shields with little to no network proximity requirements.

As a result, Data Security Broker Shields must be installed in Kubernetes or Red Hat OpenShift clusters across a number of different geographical areas for a deployment that aims to achieve maximum availability. One active Data Security Broker Manager can still control every Shield deployed across all of the different regions as shown in Figure 2.

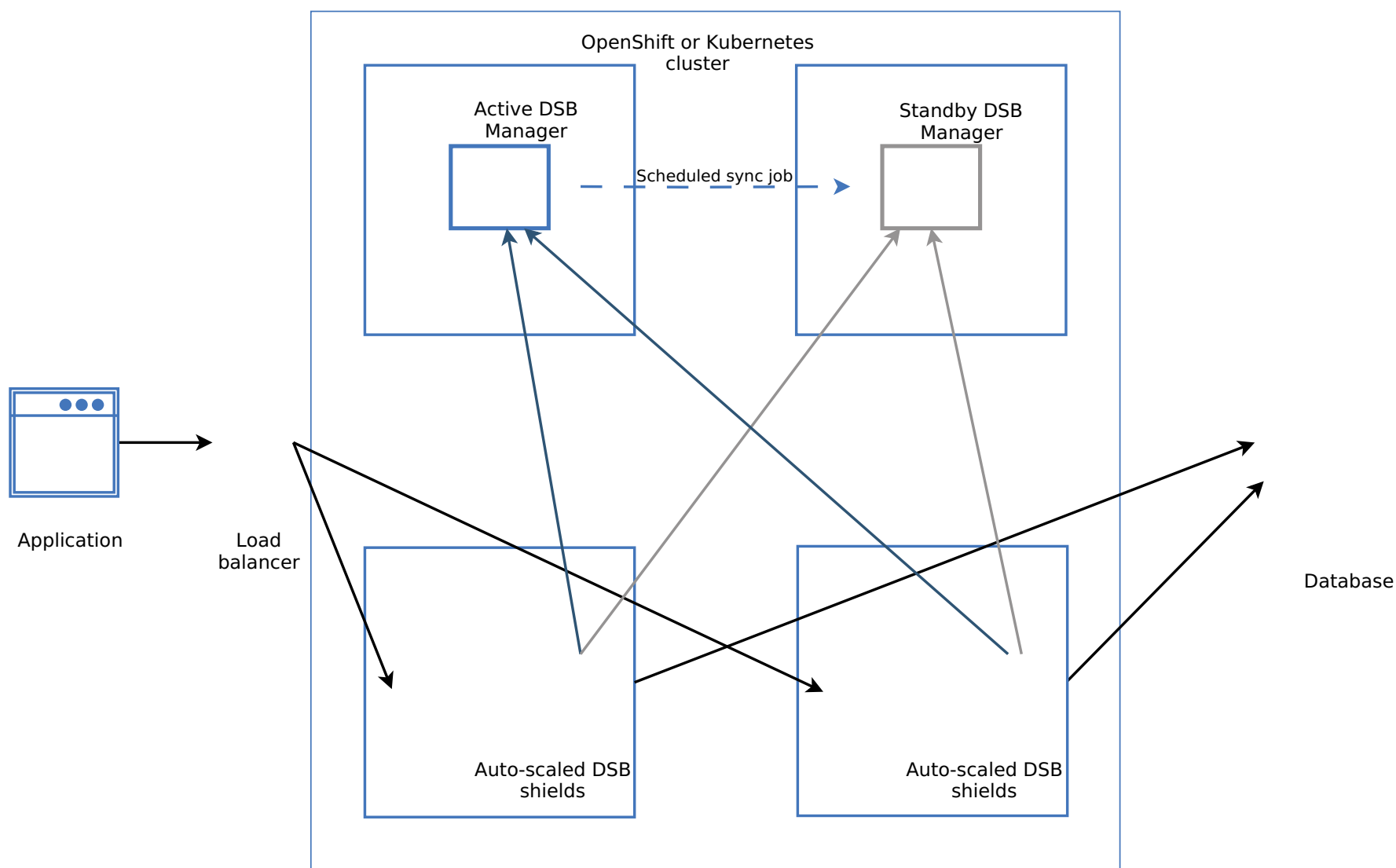


Figure 2: High availability deployment architecture of Data Security Broker

For applications in each region, one or more load balancers must be set up with routing rules that direct traffic to the shield connection with the lowest latency.

Horizontal Pod Autoscaling (HPA) Configuration

Data Security Broker advises using horizontal pod scaling for each cluster to increase service availability within the cluster. Data Security Broker Shield's deployment cluster's HorizontalAutoscaler (HPA) feature can be used to configure horizontal scaling. Based on the proper resource utilization, this HPA can be configured. The HPA configuration described in the following YAML file supports scaling for increased cluster availability. Data Security Broker recommends boosting the **maxReplica** setting to a value appropriate for the anticipated peak load in order to scale and handle the large peak loads. It is recommended to allocate the same number of CPUs to the database server's CPUs as you would do for the maximum number of replicas.

```

apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: data_security_broker-pg-shield-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: data_security_broker-pg-shield
  minReplicas: 1
  maxReplicas: 2
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 75
  - type: Resource
    resource:
      name: memory
      target:
        type: Utilization
        averageUtilization: 75

```

Deploying Data Security Broker Manager for High Availability

Currently, Data Security Broker Manager supports a high availability deployment that is active-passive. To configure and manage Data Security Broker Shields across all regions for a specific application, administrators can only use a single instance of Data Security Broker Manager.

The metadata kept by the Data Security Broker Manager instance must regularly be backed up from the primary instance and restored to the standby instance in order to guarantee that the metadata for the primary and standby Data Security Broker Managers remain consistent. To perform the backup and recovery procedures required to synchronize the Data Security Broker Manager instances, see [Disaster recovery for IBM Cloud Data Security Broker](#).

Synchronization Frequency

In order to configure data encryption policies or audit current policies, service availability requirements determine how frequently the active and standby Data Security Broker Managers synchronize. Typically, these are rare occurrences, and a job that automatically synchronizes metadata between the primary and standby must be adequate.

The synchronization job between the primary and standby Data Security Broker Managers must be run after each configuration change if you do not wish for configuration loss.

Deploying Across Regions

The standby Data Security Broker Manager instance must be deployed in a different geographic region than the primary Data Security Broker Manager instance in order to maximize availability of Data Security Broker Manager in the event of any outage. It is not necessary for the two Data Security Broker Manager instances to be directly connected to one another, but they must both have access to a common backup repository where the backup files used to synchronize the two instances are kept.

To ensure that Data Security Broker Shields are aware of both the primary and standby Data Security Broker Manager instance, follow the steps below:

1. The IP address of the deployment of the standby Data Security Broker Manager must be configured in the primary Data Security Broker Manager's **Proxy_Access** parameter.
2. The primary Data Security Broker Manager deployment's IP address must be entered in the **BM_IP** parameter for the Helm Chart that is used to deploy Data Security Broker Shield.



Note: Ensure that you add and configure the Application in Data Security Broker Manager after completing High availability configuration.

Understanding disaster recovery for IBM Cloud Data Security Broker

Components of the Data Security Broker data encryption services do not require any protected data to be stored or cached to operate. This guarantees that the data is not lost if any Data Security Broker component fails or goes down. In Data Security Broker components, only policy configurations and metadata is stored, and all this data is recoverable.

It is possible to back up the configurations and metadata stored in a Data Security Broker deployment and recover them after a significant outage, facilitating quick service restoration in a disaster recovery scenario. This section outlines the procedures for backing up and restoring the Data Security Broker deployment.

Backup considerations

The following services are required for a complete Data Security Broker deployment:

- Key management services
- Database
- IBM Cloud® Object Storage

Ensure that the backup schedule for these services align with the Data Security Broker backups. Additionally, administrators must keep records which are used for linking the Data Security Broker backup and the backups of the other services, as well as all backups for the Data Security Broker deployment and the above mentioned services, in a secure location.

Backup procedure

In Data Security Broker Manager, all configuration and metadata data needed for a Data Security Broker deployment is stored. Data Security Broker Manager's most recent configuration is always accessible to Data Security Broker Shields. As a result, Data Security Broker Manager is the only part of Data Security Broker that requires backup.

Backing up Data Security Broker Manager deployed on IBM Cloud cluster or (Red Hat OpenShift) cluster

Data Security Broker Manager instances running in a Kubernetes cluster or Red Hat OpenShift cluster require network access to the cluster as well as access to the workstation with permission to execute Kubernetes or Red Hat OpenShift command line tools. Additionally, the workstation must have a

directory with write permissions and sufficient storage to store the backup. 125GB of storage space is sufficient for all deployments.

To take a backup, follow these steps:

1. Access the cluster where Data Security Broker Manager is deployed by logging into a Kubernetes or Red Hat OpenShift workstation.
2. To back up the MongoDB collections and Data Security Broker Manager configuration files, create the script provided below and execute it. The location that is specified after the **-b** option is where the backup file is kept by the script. Check whether the script is being used to back up a Data Security Broker Manager deployment on a Red Hat OpenShift cluster or a Kubernetes cluster, and uncomment the relevant command alias for the specified type of cluster where Data Security Broker Manager is installed.

```
#!/bin/bash
if [ $# -eq 0 ]
then
    echo "No arguments supplied"
    echo "Usage: $0 -b <backup location> -n <k8s namespace>"
fi

while getopts b:n: flag
do
    case "${flag}" in
        b) backup=${OPTARG};;
        n) namespace=${OPTARG};;
    esac
done

if [ -z "${backup}" ];
then
    echo "Please provide backup location with -b option"
    exit
fi

if [ ! -d "$backup" ];
then
    echo "Please provide a valid backup location with -b option"
    exit
fi

if [ -z "${namespace}" ];
then
    echo "Please provide a valid namespace with -n option"
    exit
fi

    ### NOTE: SET THE kb ALIAS TO THE CORRECT ONE FOR THE CLUSTER TYPE
#
## For Kubernetes
#kb='kubectl --namespace $namespace'
## For OpenShift
kb='kubectl --namespace $namespace'

# Retrieving container details
#
#
echo $kb

BM_container="$(($kb get pods -o=jsonpath='{range .items[?(@.metadata.labels.app=="dsb-manager")]}{.metadata.name}')"
Mongo_container="$(($kb get pods -o=jsonpath='{range .items[?(@.metadata.labels.app=="dsb-mongoddb")]}{.metadata.name}')"
echo "BM_container ==>" $BM_container
echo "Mongo_container ==>" $Mongo_container

version="$(($kb exec -it $BM_container -- bash -c 'cat /opt/baffle/BAFFLEVERSIONS | grep BMVERSION' | awk -F "=" '{print $2}')"
version="${version//'\r/'}"
echo "version ==>" $version

BM_BACKUP_DIR=$backup

MONGO_USER="$(($kb exec -it $Mongo_container -- bash -c 'cat /run/secrets/mongodb_user' )"
MONGO_PASS="$(($kb exec -it $Mongo_container -- bash -c 'cat /run/secrets/mongodb_pass' )"
echo $MONGO_USER
echo $MONGO_PASS

backup_directory=$BM_BACKUP_DIR/$version
mkdir -p $backup_directory
echo "backup_directory ==>" $backup_directory

#cp -r $BM_DEPLOY_DIR/config $backup_directory
```

```

# MongoDB backup
BackupMongo() {
  echo "MongoDB Backup.."
  readarray -t backup_dbs <<($kb exec -it $Mongo_container -- sh -c 'mongo m_db --quiet -u '$MONGO_USER' -p '$MONGO_PASS' --authenticationDatabase admin --eval "db.tenant.find({}, {"_id\":false,\"tid\":true})"' | awk -F'"' '{print $4}')
  $kb exec -it $Mongo_container sh -c 'mongodump -u '$MONGO_USER' -p '$MONGO_PASS' --authenticationDatabase admin -d m_db -o /tmp/dump'
  for backup_db in "${backup_dbs[@]}"; do
    $kb exec -it $Mongo_container -- sh -c 'mongodump -u '$MONGO_USER' -p '$MONGO_PASS' --authenticationDatabase admin -d '$backup_db' --excludeCollection=audit -o /tmp/dump'
  done
  backupDumpMongoFile="/tmp/"$version"MONGO.tar.gz"
  $kb exec -it $Mongo_container -- sh -c 'tar -czvf '$backupDumpMongoFile' /tmp/dump'
  $kb cp $Mongo_container:$backupDumpMongoFile $backup_directory/"$version"MONGO.tar.gz
}

# DSB Manger backup
BMbackup() {
  echo "BM backup.."
  BMbackupDumpFile="/tmp/"$version"BM.tar.gz"
  $kb exec -it $BM_container -- sh -c 'tar -czvf '$BMbackupDumpFile' /opt/dsb/dsb-manager'
  $kb cp $BM_container:$BMbackupDumpFile $backup_directory/"$version"BM.tar.gz
}

BackupMongo
BMbackup

```

- The following files are added to the specified backup location after you have finished executing the script:

```

Release-DSB.<release>MONGO.tar.gz
Release-DSB.<release>BM.tar.gz

```

- Transfer the backups of the key management, Object Storage, and database services related to the Data Security Broker deployment, as well as the Data Security Broker Manager backup files, to a reliable backup storage location.

Restore procedure

The Data Security Broker deployment can be recovered in a disaster recovery situation to a state that was previously backed up, provided that all the dependent services are also recovered in the same manner. This applies to any services that has suffered a irrecoverable failure. If the dependent services were not affected, all that is required is to ensure that the network connectivity between the Data Security Broker deployment and the dependent services is restored.

Since Data Security Broker Manager contains all the configuration and metadata needed for a Data Security Broker deployment, the goal of the process is to restore Data Security Broker Manager to a previous state so that a new shield can be deployed to enforce the previously configured security policies.

Restoring Data Security Broker Manager deployed on a Kubernetes or Red Hat OpenShift cluster to a previous state

A new Data Security Broker Manager deployment needs to be set up in a Kubernetes or Red Hat OpenShift cluster, in order to return Data Security Broker Manager to its previous state. The administrator carrying out the restore operation requires network connectivity to the cluster as well as access to a workstation with permission to use the command-line tools for Kubernetes or Red Hat OpenShift cluster.

To restore a Data Security Broker deployment, follow the steps:

- Log in to the workstation using command-line tools, and use the recently installed Data Security Broker Manager to access the Kubernetes or Red Hat OpenShift cluster.
- To create a temporary storage area on the workstation, copy the Data Security Broker Manager backup files there. The names of the backup files includes the following:

```

Release-DSB.<release>MONGO.tar.gz
Release-DSB.<release>BM.tar.gz

```

- To restore the Data Security Broker Manager configuration files and MongoDB collections, create and execute the script provided below. As an input to the script, specify the location of the temporary storage location for backup files.

```

#!/bin/bash
if [ $# -eq 0 ]
then
  echo "No arguments supplied"
  echo "Usage: $0 -m <mongodb backup tar.gz> -b <BM backup tar.gz file> -u
<MongoDB user name used for original dsb Manager deployment> -p <MongoDB
password used in original dsb Manager deployment> -n <k8s namespace>"

```

```

fi
while getopts m:b:u:p:n: flag
do
case "${flag}" in
m) MONGO_DUMP_FILE=${OPTARG};;
b) BM_DUMP_FILE=${OPTARG};;
u) MONGO_USER=${OPTARG};;
p) MONGO_PASS=${OPTARG};;
n) NAMESPACE=${OPTARG};;
esac
done
if [ -z "${MONGO_DUMP_FILE}" ];
then
echo "Please provide backup location with -m option"
exit
fi
if [ -z "${BM_DUMP_FILE}" ];
then
echo "Please provide backup location with -b option"
exit
fi
if [ ! -f "$MONGO_DUMP_FILE" ];
then
echo "Please provide a valid backup location with -b option"
exit
fi
if [ ! -f "$BM_DUMP_FILE" ];
then
echo "Please provide a valid backup location with -b option"
exit
fi
if [ -z "${MONGO_USER}" ];
then
echo "Please provide Mongoddb user via -u option. It should be same user used
for backup"
exit
fi
if [ -z "${MONGO_PASS}" ];
then
echo "Please provide Mongoddb password via -p option. It should be same password
specified in the original dsb Manager deployment"
exit
fi
if [ -z "${NAMESPACE}" ];
then

echo "Please provide kubernetes NAMESPACE via -n option."
exit
fi
### NOTE: SET THE kb ALIAS TO THE CORRECT ONE FOR THE CLUSTER TYPE
## For Kubernetes
#kb='kubectl --namespace '$NAMESPACE
#echo "KB = "$kb
## For OpenShift
#kb='kubectl --namespace '$NAMESPACE
echo "OC = "$kb

# Retrieving container details
BM_container="$($kb get pods -o=jsonpath='{range .items[?(@.metadata.labels.app=="dsb-manager")]}{.metadata.name}')"
Mongo_container="$($kb get pods -o=jsonpath='{range .items[?(@.metadata.labels.app=="dsb-mongoddb")]}{.metadata.name}')"
echo "BM_container ==>" $BM_container
echo "Mongo_container ==>" $Mongo_container

version="$($kb exec -it $BM_container -- bash -c 'cat /opt/baffle/BAFFLEVERSIONS | grep BMVERSION' | awk -F "=" '{print $2}')"
version="${version//'\r'}"
echo "version ==>" $version

# Restore DSB MongoDB
RestoreMongo()
{
echo "Mongoddb Restore..."
MONGO_FILE=$(basename $MONGO_DUMP_FILE)
echo "MONGO_FILE ==>" $MONGO_FILE
$kb cp $MONGO_DUMP_FILE $Mongo_container:/tmp/$MONGO_FILE
$kb exec -it $Mongo_container -- sh -c 'tar -xvf /tmp/$MONGO_FILE
$kb exec -it $Mongo_container -- sh -c 'mongorestore --username='${MONGO_USER}' --password='${MONGO_PASS}' --
nsInclude=ibm.* --verbose --authenticationDatabase=admin /tmp/dump'
}

```



```

# Restore DSB Manager
RestoreBM()
{
echo "BM Restore..."
BM_FILE=$(basename $BM_DUMP_FILE)
echo "BM_FILE ==> "$BM_FILE
$kb cp $BM_DUMP_FILE $NAMESPACE/$BM_container:/tmp/$BM_FILE
$kb exec -it $BM_container -- sh -c 'tar -xmvf /tmp/'$BM_FILE

# $kb exec -it $BM_container -- sh -c 'cp -pr /tmp/opt/dsb/dsb-manager/config /opt/dsb/dsb-manager/'
}

RestoreMongo
RestoreBM

```

4. Open a new browser tab and log into Data Security Broker Manager after you have finished executing the script. Verify that the original configurations of the applications, databases, and key stores have been restored.
5. For all of the applications of Data Security Broker Manager, with data security policies, deploy new Data Security Broker Shields by adhering to the deployment procedures. Ensure that the Helm chart is updated with the appropriate Sync ID for each application.
6. Log into Data Security Broker Manager after the Data Security Broker Shield have been deployed to ensure that the newly installed Data Security Broker Shields are in the **RUNNING** status.

Best Practices for backups and restore:

Setting Up Configuration backup in IBM Cloud Object Storage

- It is recommended to store the database configuration and the backup objects in a Object Storage instance. The backup is stored in a separate cloud account with Object Storage buckets created.
- Ensure that the backup Object Storage bucket is configured with immutable settings.
- Setup an Automation jobs to back up the database, preferably twice a day and upload it to the Object Storage bucket. It is recommended to setup an automation job to restore the database whenever necessary by picking up the latest changes from the backup Object Storage bucket.

Logging and monitoring

Logging and Debugging

Overview:

This section demonstrates how to enable logging for Data Security Broker. Use IBM® Log Analysis to add log management capabilities to Data Security Broker.

The logging agent collects and forwards logs to your IBM Log Analysis instance. After you provision an IBM® Log Analysis instance, you must configure a logging agent for each log source that you want to monitor in Data Security Broker Manager. For more information on provisioning a log instance, see [Logging with IBM Cloud Services](#).

Pre-requisites

- You need a **logdna** instance, which is configured to receive platform logs.

Procedure

To configure a logging instance from the **Observability** dashboard in the IBM Cloud, complete the following steps:

1. Log in to your IBM Cloud account. After you log in, the IBM Cloud UI opens.
2. Go to the Menu icon in the left navigation and click **Observability** to access the **Observability** dashboard.
3. Click **Logging**, then click **Options > Edit Platform**.
4. Select a region.
5. Choose which logging instance will receive logs from Data Security Broker on that location.
6. Click **Select** to create the log instance.

After you provision an instance of the IBM Log Analysis service in the IBM Cloud, and configure a logging agent for a log source, you can view, monitor, manage log data, and debug the logs through the IBM Log Analysis Web UI. To debug the logs, follow the steps below:

- a. Log in to your IBM Cloud account. After you log in, the IBM Cloud UI opens.
- b. Go to the Menu icon in the left navigation and click **Observability** to access the **Observability** dashboard.
- c. Click **Logging** and select the log instance which you have created for the Data Security Broker.
- d. Click **Open Dashboard** for the log instance that you have created. You can view the detailed log summary in the dashboard.
- e. You can start debugging the logs, which are displayed with a predefined format. You can also filter logs and modify search settings, then bookmark the result as a view.

Data Security Broker Monitoring

You can monitor Data Security Broker with IBM Cloud® Monitoring dashboard. For more information about IBM Cloud Monitoring, see the IBM Cloud Monitoring [Getting started tutorial](#).

You can setup the following monitoring for Data Security Broker:

- Configuring the alert tool for the IKS/ROKS cluster deployment.
- Configuring alert rules when there is no pod availability for deployment.
- Adding customized alerts from the Alerts Library.

Configuring the alert tool for IKS/ROKS cluster deployment

You can configure the alert tool through which the alerts will be sent, for any issues in the clusters.

1. Log in to the IBM Cloud account.
2. Select **Kubernetes** -> **Clusters** from the left navigation menu to open the [Kubernetes Clusters Dashboard](#), to view the the list of clusters.
3. Select the cluster where you have installed Data Security Broker.
4. Click **Launch** under **Monitoring** in the **Integrations** section to open the IBM Cloud Monitor Dashboard.
5. Select Integrations -> **Notification Channels** in the left navigation menu.
6. Click **Add Notification Channel** and select the alert tool from the list of tools available, which will be used to send the alerts for any issues in the cluster.

Configuring alerts for pod unavailability during deployment

You can configure alerts when the pods are not available in a cluster for the Data Security Broker by following the steps below:

1. Navigate to the **IBM Cloud Monitor** dashboard.
2. In the left navigation menu, select **Alerts** and click **New Alert** to configure an alert.
3. Select **PromQL** as the **Alert type** and enter the condition mentioned below in the **Condition** text box and click **Run Query**.

```
kube_deployment_status_replicas_available{namespace="<your_dsb_manager_namespace>", deployment="dsb-manager"} < 1
```



Note: Note: The above condition is used to configure an alert when there is no pod available for the Data Security Broker Manager pod deployment. You can specify the different condition for different types of pods for the Data Security Broker deployment and configure alerts for the corresponding pods.

Use the following conditions to configure alerts for different pod deployment:

dsb-mangodb alert condition:

```
kube_deployment_status_replicas_available{namespace="<your_dsb_manager_namespace>", deployment="dsb-mongodb"} < 1
```

dsb-nginx alert condition:

```
kube_deployment_status_replicas_available{namespace="<your_dsb_manager_namespace>", deployment="dsb-nginx"} < 1
```

dsb-web alert condition:

```
kube_deployment_status_replicas_available{namespace="<your_dsb_manager_namespace>", deployment="dsb-web"} < 1
```

dsb-shield alert condition:

```
kube_deployment_status_replicas_available{namespace="<your_dsb_manager_namespace>", deployment="dsb-shield-app1"} < 1 dsb-nginx:
```

4. Under **Notifications**, select an alert tool, which is already configured, in the **Notification Channel** drop-down list.
5. Specify the alert name, alert severity, and description for the alert under **Settings** for the alert.
6. Click Save to configure the Prometheus Alert.

Adding customized alerts from the Alerts Library

In this section, you can find information on how to add pre-defined alerts, which are available in the Alerts Library. For example, you can configure an alert when the Container Status remains in the Waiting status for a long time using the template available in the Alerts Library.

Configuring Alerts to check the Container status

1. Navigate to the **IBM Cloud Monitor** dashboard.
2. In the left navigation menu, select **Alerts** and click **New Alert** to configure an alert.
3. Select **Browse Alert Library** in the **Select Alert Type** window.
4. Click **Kubernetes** in the Alerts Library.
5. Select **[Kubernetes] Container Waiting** alert.
6. Choose the cluster, namespace, workload where the alert must be configured to monitor the container waiting status and select the notification channel, from which the alerts will be sent.
7. Click **Enable Alert**.



Note: Note: You can try all other alert templates available in the **Alerts Library** according to your requirement.

Getting help and support

If you have problems or questions when you use Data Security Broker, you can get help by opening an IBM Cloud support case.

To learn about opening an IBM support case, or about support levels and case severities, see [Contacting support](#).

FAQs for Data Security Broker

This section provides answers to common questions about the Data Security Broker service.

What is Data Security Broker Manager?

Data Security Broker Manager enables encryption policies and configurations by communicating with the Data Security Broker Shield and the desired databases. Data Security Broker Manager constructs a privacy schema that maps key IDs to data columns, thus enabling encryption in a simplified manner. Data Security Broker Shield carries out de-identification, masking, and encryption tasks for cloud databases.

Data Security Broker integrates with key management stores through a key virtualization layer. It also provides for a local key store, so you can use your own keys for data protection in the cloud.

Is the instance hosted by Data Security Broker?

Data Security Broker software is hosted entirely in the customer's environment. Data Security Broker Manager and Data Security Broker Shield may be hosted on-premises or on cloud platforms such as IBM Cloud.

What size instance is good for getting set up?

30 gigabytes are sufficient for Data Security Broker Manager because it controls the Shields and the memory needed stays static. Sizing for Data Security Broker Shield depends on the number of shields and the number of concurrent users.

What are the prerequisites for installing Data Security Broker?

Make sure you meet the following requirements before configuring Data Security Broker Manager and Data Security Broker Shield:

- Admin privileges for your platform
- The user account used to log in to the Data Security Broker Shield host machine must have a home directory on that system
- SSH client
- Private key pair
- Database privileges for encryption and migration

Do Data Security Broker Manager and Shield run on different instances?

Technically, it is possible for the Data Security Broker Manager and Data Security Broker Shield to both run on the same host. However, it is recommended that separate host instances are provisioned for the Data Security Broker Manager and Data Security Broker Shield, to accommodate different workloads.

Which OS is necessary to set up the Data Security Broker Shield?

Data Security Broker Shield can be installed on instances running with Ubuntu 18 for **IBM Cloud Kubernetes cluster** (IKS) or RedHat Enterprise Linux (RHEL) 7 for **IBM Red Hat OpenShift Kubernetes cluster** (ROKS).

Why does an invalid certificate browser warning occasionally show up?

Most web browsers will flag this connection as unsafe if Data Security Broker Manager is not initialized with an HTTPS certificate. To avoid the warning, configure HTTPS in the settings page in Data Security Broker Manager.

Where are the secrets submitted to Data Security Broker kept?

A secure credential store is created in MongoDB by Data Security Broker Manager. This is encrypted with a key that is configured during set up.

Why is the Deploy Policy and Migrate Data option disabled in the Data Security Broker Manager UI?

The application in which you are trying to perform the encryption or decryption does not have a Data Security Broker Shield attached to it. Associate a Data Security Broker Shield to the application before performing the encryption or decryption process.

Where do I get the information to connect the application to Data Security Broker Shield?

If you have self-registered Shields, they are added using the Shield Sync ID, after the application is set up. The Shield Sync ID is found in the Application side panel.

Are there any analytics reports?

For this service, analytics reports are not applicable.

How do I debug if there are any problems?

Refer to the [Logging and Debugging in Data Security Broker section](#).

How do I add multiple Data Security Broker Shields?

Yes, you can install multiple instances of the Data Security Broker Shields using the IBM Cloud Catalog. Ensure that you have specified different Shield names during the installation for each Shield.

How is the load balanced between Shield instances?

Load balancing between Shield instances is managed by OpenShift load balancing policies.

What kind of delays should I expect with the Shield acting as a proxy?

The environment, network latency, and network architecture that manage the database and CDSB components all play a role in this. The delay is only a few milliseconds.

How do I add more users to the account?

Refer to the [Adding users in Data Security Broker Manager section](#).

What is the impact of encryption on indexing?

If the index column is encrypted, in standard encryption mode, sorting on that encrypted column will not work, by design, as the encrypted value has no relationship to the cleartext data. All operations will work once the data is extracted from the encrypted database as the proxy will decrypt it.

What is the impact of encryption on search?

For exact match searches, where a query is looking for a specific value, it will work as long as the user issuing the query is authorized since they will be able to obtain the key to decrypt the value. For partial match searches, like wildcard searches where only a part of the ciphertext is to be identified, the search will fail as the query will run on encrypted data that has no relationship to the plaintext.

These limitations are identical to any application encryption approach where the application makes API calls to encrypt and decrypt the data. Both of these queries, sorting on encrypted indexes and wildcard searches, will work with the Baffle advanced encryption mode where the decryption happens in a secure way on the server side itself using Postgres extensions.

What are some known issues with respect to Key Management services?

You must have a place to store your secret key or key file. You can use IBM Key Protect. IBM Key Protect provides full encryption visibility and control, allowing you to see and manage data encryption and the entire key lifecycle from a single location. Alternatively, you can secure the information with a password, but this reduces security, depending on the level of your password's strength.



Note: During Keystore enrollment, Data Security Broker Manager will accept a Key Protect alias that refers to a destroyed or disabled Master Key. Destroyed Keys are unusable, and Data Security Broker migration will fail in this case. As a workaround, ensure that the Keystore Alias does not refer to a destroyed Master Key.

What if there is a performance hit?

Data encryption consumes more time and resources than data decryption. This overhead is normally negligible. If the proxy (Shield) has an appropriately sized CPU and Memory, there should not be any noticeable performance penalties. Most users experience a 10--20% reduction in the overall application performance. This can be mitigated by horizontal scaling. In the event of failure, Shield can be horizontally scaled by adding more of them behind a load balancer.

What if I lose my key file?

If your key file is lost, your data is also lost. So, make a backup of your data and your key.

What are the limitations with respect to the database operations carried out on encrypted versions of the data?

The information is kept in encrypted form in the database and the database engine and, consequently, the database administrator is never permitted to see the information in form of the plaintext.

Many of today's server and network technologies allow for easier configuration and implementation to minimize the impact on utilization. Implementing encryption of data in transit from endpoint to endpoint both remotely and internally is mandatory in today's cyber risk environment.

The following are considered equality check operators and are supported:

- =
- <>
- IS NULL
- IS NOT NULL
- IN
- NOT
- JOIN (all types)
- GROUP BY
- DISTINCT

Indexes can be created on encrypted columns, but the ciphertext is used to create the index and not the underlying cleartext values.

How do I upgrade my Data Security Broker Manager?

Upgrades from previous versions of Data Security Broker Manager is no longer available. Any credentials that were previously stored in MongoDB, in earlier versions, cannot be upgraded to new versions.

What are the unsupported data types for encryption?

The **timestampz** and **timetz** data types are not supported for the PostgreSQL Database in Data Security Broker.

Troubleshooting

What if the HTTPS certificate is not added?

The HTTPS certificate is not added and thus, initialization and setup process is failing.

What's happening

You get error messages or warnings during the initialization or the setup of Data Security Broker with the following message:

Message: Untrusted certificate browser warning

Why it's happening

HTTPS certificate has not been updated. This warning occurs when the Data Security Broker Manager's pre-scanned certificate does not match the hostname of the Manager server.

How to fix it

Upload an HTTPS certificate by following the instructions in [Adding HTTPS certificate](#) section.

How to resolve the database related issues?

You face database-specific issues.

What's happening

You get errors or warnings during database enrolment.

How to fix it

If you get error messages or warnings during the database enrolment, check for the following troubleshooting tips:

1. Message: Database credentials are incorrect

Ensure the Database Username and Password are correct.

2. Message: Hostname or IP address is not submitted

Copy and paste the database endpoint precisely in the appropriate field, with no extra characters or spaces.

3. Message: SSL certificate is not accepted

Ensure you have uploaded a valid SSL certificate.

4. Message: Max connections reached on the database

Some database instances contain a limit on the maximum allowed connections to the server. For example, if the Postgres database is not connecting, check for the following troubleshooting tips:

- Specify the Postgres Database name.
- For PostgreSQL database servers, you must also pass the name of the database itself, which you intend to connect.
- The default name is postgres; however, if there is not a database named "postgres" on your server, then the connection fails.

How to resolve issues that occur during Shield enrollment?

You face issues during Data Security Broker Shield enrollment.

What's happening

Error messages or warnings displayed during Shield enrollment.

If you get error messages or warnings during the Shield enrollment, check for the below scenarios:

1. Hostname or IP address is not submitted: Copy and paste the Shield host endpoint precisely in the appropriate field, with no extra characters or spaces.

2. Shield host is already in use by Data Security Broker: Multiple Shields may be hosted on the same endpoint, but only if the port number is unique. If the same port number is reused, the connection fails.

3. Public IPv4 address is configured: If all Data Security Broker resources are running in the same environment, then it is recommended to use the

private IP address. Alternatively, submit the DNS endpoint for the instance.

4. Insufficient permissions to write to the temporary directory: The Host User must have write permissions for a temporary directory on the Shield host. The default directory is **/tmp**. If the Host User does not have permissions for **/tmp**, submit an alternative temporary path in the appropriate field.

How to resolve issues with respect to Keystore enrollment?

Adding a Keystore is a mandatory process in configuring the Data Security Broker Manager.

What's happening

You get errors or warnings during Keystore enrollment.

Why it's happening

User Permissions are insufficient.

How to fix it

Check for user permissions.

Also, ensure the user credentials which is connected to the configured Keystore have adequate permissions.

How to resolve the Application workflow issues?

You get error messages or warnings during the application encryption workflow.

How to fix it

If you get error messages or warnings during the application encryption workflow, check for the following troubleshooting tips:

1. Encrypt button is disabled: Shield is initializing, restarting, or stopped. The Shields enrolled in an application must all be in a **running** state to enable the Encryption operations.
2. Shield connection to the database fails: Database credentials are changed. The Database Username or Database Password might have been updated; from the time the connection was created in Manager.
3. Database grants are insufficient. The database credentials which the Shield uses to create a session must have a minimum set of permissions.
4. Shield does not use SSL. If the database connection uses SSL, the Shield connection must also use SSL.
5. Custom Data Protection mode is not visible in the Data Protection dropdown for selection. Custom mode is invalid for the selected datatype. Certain custom encryption modes and masking modes may only be applied on data types.
6. Multiple Data Protection modes are selected on one column. Default Data Protection is overridden. By design, default data protection policies may be applied on any column. If a custom or column-specific Encryption Mode or Masking Mode is simultaneously selected for a column, then the default policy is ignored. The custom selection is applied.
7. Option to Deploy Policy & Migrate Data is disabled. Shield is initializing, restarting, or stopped. The Shields enrolled in an application must all be in a **running** state to enable the Encryption operations. Or, if no Shields are enrolled in the application, the migration option is disabled.



Note: If you are experiencing any other issues with the Data Security Broker, go to the IBM Cloud [Support Center](#) and navigate to creating a case. Use the All products option to search for IBM Cloud Data Security Broker to continue creating the case or to find more information about getting support.

What if the Persistent Volume Claim (PVC) goes into pending state after installing Data Security Broker Manager?

After you install Data Security Broker Manager, sometimes the PVC goes into pending state.

How to fix it

Create a YAML file with the below format and deploy the YAML file in the Kubernetes cluster, where you have installed the Data Security Broker Manager.

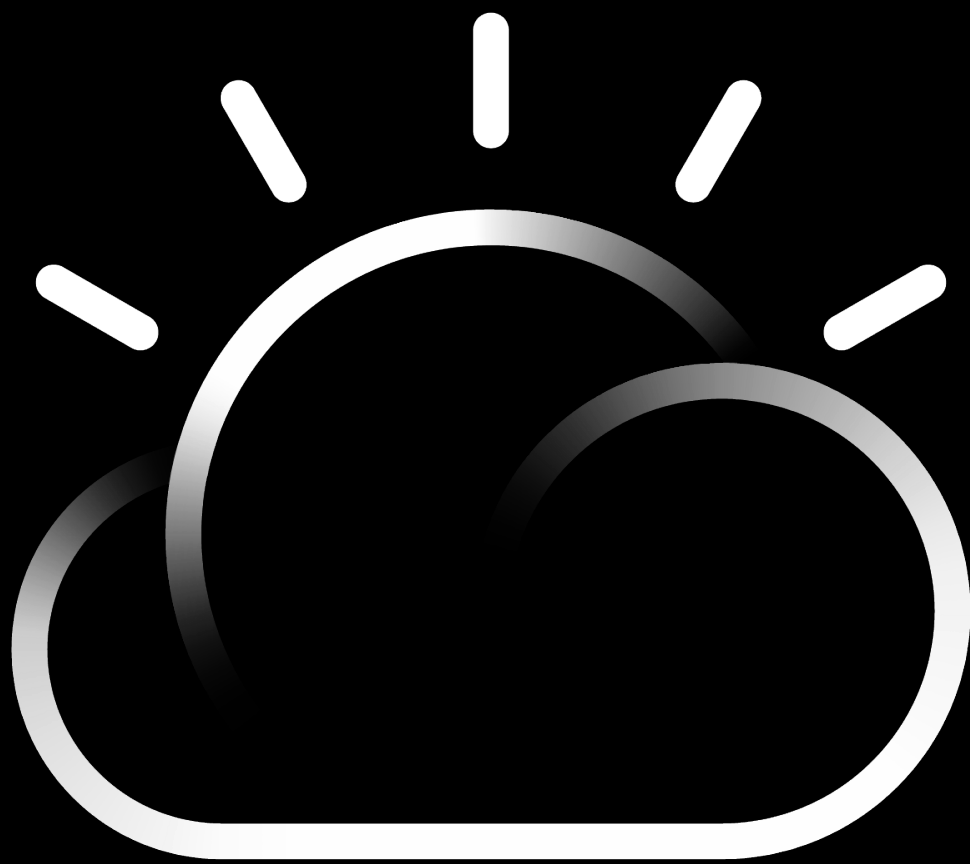
Example: A sample YAML for PVC of Data Security Broker Manager:

```
$ apiVersion: v1
kind: PersistentVolume
metadata:
```

```
name: pvc-dsb-manager
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 20Gi
  flexVolume:
    driver: ibm/ibmc-block
    fsType: ext4
  persistentVolumeReclaimPolicy: Delete
  storageClassName: ibmc-block-bronze
  volumeMode: Filesystem
```

A sample YAML for PVC of Data Security Broker MangoDB:

```
$ apiVersion: v1
kind: PersistentVolume
metadata:
  name: pvc-dsb-mongo
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 20Gi
  flexVolume:
    driver: ibm/ibmc-block
    fsType: ext4
  persistentVolumeReclaimPolicy: Delete
  storageClassName: ibmc-block-bronze
  volumeMode: Filesystem
```



IBM Cloud