CDP Private Cloud Data Services 1.5.0

CDP Private Cloud Experiences Management Console Administration

Date published: 2020-12-16
Date modified: 2023-01-25

https://docs.cloudera.com/

Legal Notice

Contents

Working with CDP Private Cloud diagnostic data

CDP Private Cloud enables you to generate and download diagnostic data associated with the various services and workloads for troubleshooting purposes. You can perform these tasks from the Diagnostic Data tab of the Management Console.

CDP Private Cloud uses fluentd to collect logs from the Management Console services, and the CML, CDW, and CDE workloads. If Ozone is configured to store logs from the workloads, the information collected by fluentd is stored on Ozone. Otherwise, the data is moved through a fluentd aggregator in the Management Console to an OpenShift persistent volume where all the Management Console logs are also stored.

When an administrator selects the option to generate the diagnostic bundle from the Management Console > Administration > Diagnostic Data page, Management Console generates a bundle containing the following details after querying the relevant components and services:

- Logs from the OpenShift persistent volume
- CDW, CML, and CDE logs stored on Ozone
- Kubernetes deployment information such as events, logs, details of the OpenShift pods and so on.
- Metrics for monitoring such as usage information and so on.
- Management Console information such as version, UUID, and so on.

Collecting CDP Private Cloud diagnostic data

You can collect and download CDP Private Cloud diagnostic data for different components and services from the Administration page of the Management Console. In addition, you can specify the type of workloads and the duration for which you want to collect the diagnostic data.

Procedure

1. On the Management Console home page, select Administration > Diagnostic Data.
2. Click Collect and Send Diagnostic Data.

   A pop-up window opens where you can customize the type of diagnostic data to collect.
3. Specify the various options on the Collect and Send Diagnostic Data window to gather the diagnostic data based on your requirements.

   You can specify the following options for data collection: the time duration, the associated environments and services, and the size of the file to include the diagnostic data. For more information on the diagnostic data options, see Options for generating the CDP Private Cloud diagnostic data on page 4.
4. Click Collect and Send Diagnostic Data in the pop-up window.

   Management Console initiates the process of generating the diagnostic data based on the options that you select. After the data is generated, the Last Collected Time field is updated with the collection time.

   If your system is connected to the internet, Management Console collects the diagnostic data and sends it to Cloudera Support.
5. To manually download the diagnostic data, click the Last Collected link.

   The diagnostic data is downloaded in a .zip file. You can send this file to Cloudera support.

Options for generating the CDP Private Cloud diagnostic data

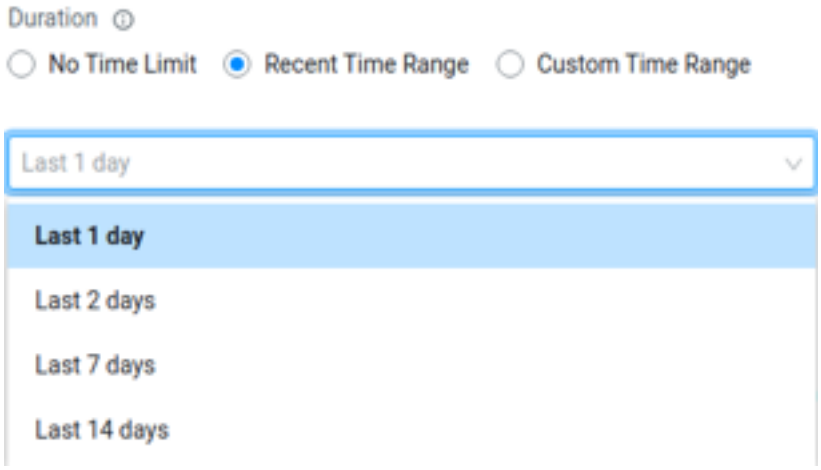
You can specify various options on the Collect and Send Diagnostic Data pop-up window to generate CDP Private Cloud diagnostic data based on your specific requirements of duration, scope, and size.

Duration

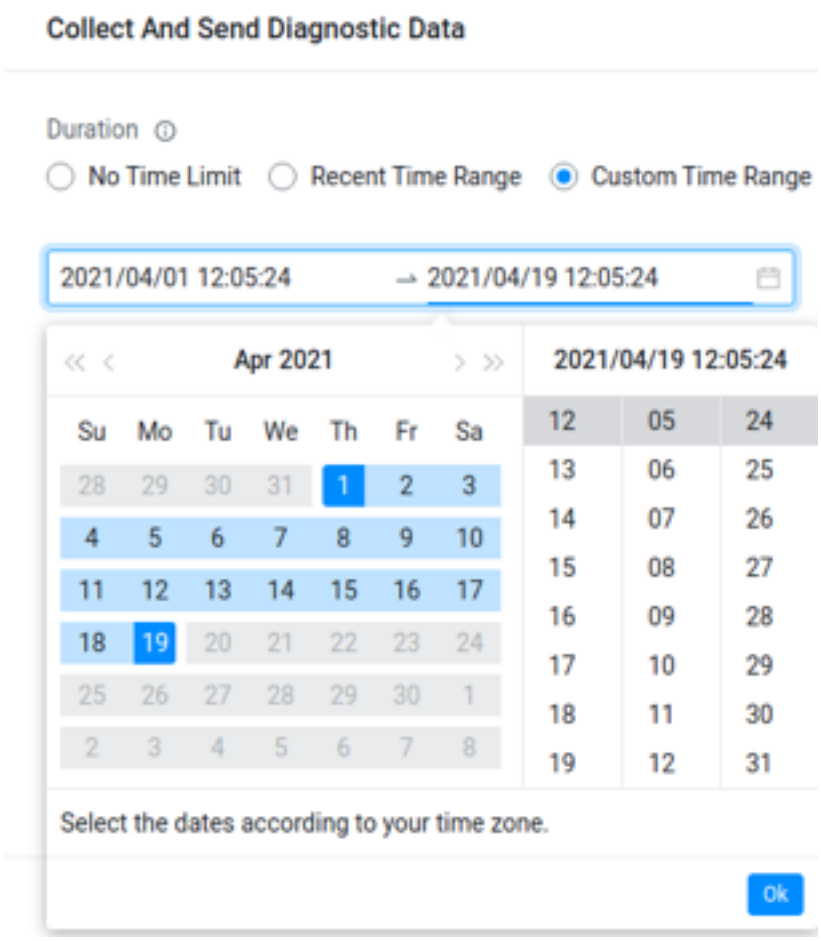
Select the duration for which you want to generate the diagnostic data.

You can select from the following options:

- No Time Limit: Select this value if you want all the diagnostic data to be generated from the time you deployed CDP Private Cloud.
- Recent Time Range: Select a specified time range, in hours, from the drop-down list.



- Custom Time Range: Select the start date and end date between which you want to generate the diagnostic data.

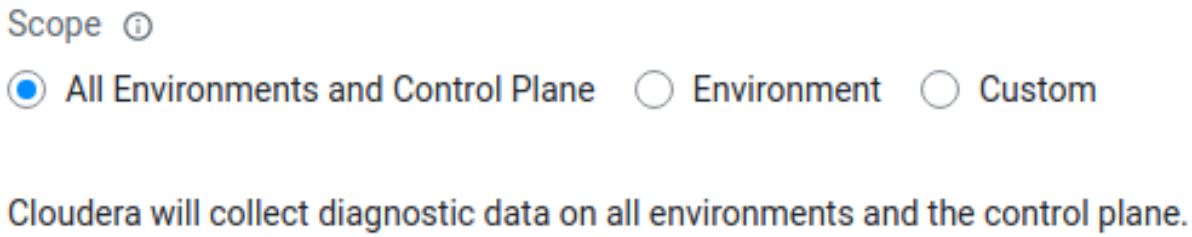


Scope

Select the services for which you want to collect the diagnostic data.

You can select environment- and workload-related options from the following:

- All Environments and Control Plane: Select this option to generate the diagnostic data for all the environments and workloads deployed on CDP Private Cloud.



- Environment: Select one or more environments from the drop-down list to generate the diagnostic data. You can also use regular expressions to filter the name of environments for which you want to generate the diagnostic data.
- Custom: You can filter your selections to specific environments and workloads, and accordingly generate the diagnostic data.

  You must click Add Scope to select an environment and provide information about its associated workloads for which you want to generate the diagnostic data. The details include the following:

  - Workload Status: For a selected environment, specify the option to select either a live or an archived workload.
  - Workload Type: For a Live workload status, you can specify the option to select CDW or CML or CDE.
  - Workload Name: For a selected workload type, you can specify the names of the workloads for which you want to generate the diagnostic data. You can add a single workload or multiple workloads as comma-separated values. If required, you can use regular expressions to filter the workload names.
  - Namespace: For an Archived workload status, specify the Kubernetes namespace corresponding to the workloads for which you want to generate the diagnostic data. You can add a single namespace or multiple namespaces as comma-separated values. If required, you can use regular expressions to filter the namespace values.
  - Pod: For an Archived workload status, specify the OpenShift pods on which the workloads are archived. You can add a single pod or multiple pods as comma-separated values. If required, you can use regular expressions to filter the names of the pods.

    Note:

    - If you do not specify any namespace, workload name, or pod, Management Console generates diagnostic data for all the namespaces, workloads, and pods associated with the specified environment-workload type combination.
    - Ensure that any regular expression that you use on the Collect and Send Diagnostic Data pop-up window is anchored by a ^ symbol at the beginning of the expression and a $ symbol at the end.

Size

You can specify the size of the diagnostic data to collect. The default size of the data after compression is 500 MB. If you want to change the default size, set the BUNDLE_SIZE_LIMIT_MB environment variable on the OpenShift pods on your CDP Private Cloud deployment.

Note: The actual size of the generated diagnostic bundle might vary from the size that you specify.

While CDP can exclude namespace pod logs, control plane archive logs, and other archive logs from the diagnostic bundle to be consistent with the specified size, CDP must always include certain type of diagnostic data in the bundle. Because the size of the diagnostic data that is always included in a bundle is not constant, the overall size of the generated diagnostic bundle might vary.

CDP Private Cloud Data Services diagnostic data

You can collect and download CDP Private Cloud Data Services diagnostic data using the Diagnostic Data tab on the Administration page of the Management Console. This topic lists the diagnostic data that is generated.

CDP Private Cloud Data Services diagnostics bundle

Namespace information (under <namespace>/)

- pods.json
- events.json
- replication.json
- services.json
- daemonset.json
- deployments.json
- replicasets.json
- configmap.json
- pv.json
- Pod logs (under logs/)
- Archive logs (under archived/)

bundle.json

- File name of diagnostics bundle
- Control plane version
- Control plane namespace

Environment information (under environment/)

- Per environment (under <environment-name>/)
  - Archive Logs (under archived/)
    - Non-Ozone logs (those stored in the persistent volume claim of the fluentd-aggregator pod)
    - Ozone logs (those stored in Ozone)
- cde
  - Separated into folders per namespace for the CDE services
  - Each folder contains namespace information as above
- dwx
  - Separated into folders per environment
  - Each folder contains namespace information as above
- mlx
  - Separated into folders per MLX workspace
  - Each folder contains namespace information as above
- environment.json
  - Contains the following:
    - CM version,
    - CDH version
    - CM License UUID
    - Health of all of the hosts
- client-configs.zip
  - Client configs of the base cluster

- details.json
  - Environment details

license.json

- License version
- License Name
- License UUID
- License Start date
- License Deactivation date
- License Expiration date

log.txt

- Good for debugging the diagnostics collection
- All log statements from the diagnostics collection are found here

monitoring-metrics.json

- The results of various monitoring queries

nodes.json

- Details of all cluster nodes
- Examples: labels, creationTimeStamp, etc.

Pod reaper logs (under pod-reaper/)

- Job logs for pod reaper

Configuring LDAP authentication for CDP Private Cloud

You can configure LDAP user authentication for CDP Private Cloud from the Administration page of the Management Console.

Before you begin

If you intend to use Hue as your SQL editor in CDW, you must use LDAP over SSL.

Procedure

1. Sign in to the CDP console.
2. Click Management Console.
3. On the Management Console home page, select Administration>Authentication.
4. Configure the following settings for LDAP authentication:

Property	Description	Sample values
LDAP URL	The LDAP server URL. The URL must be prefixed with ldap:// or ldaps://. The URL can optionally specify a custom port, for example: ldaps://ldap_server.example.com:1636.	ldap://<ldap-host>:389 or ldaps://<ldap-host>:636

Property	Description	Sample values

CA Certificate for Secure LDAP	The X.509 PEM certificate to be used to access secure LDAP (URLs starting with ldaps://). Ensure that at least one valid certificate is provided. A typical CA certificate is structured as follows:
	`-----BEGIN CERTIFICATE-----`
	`...`
	`-----END CERTIFICATE-----`
		If you add or update CA certificates and you have deployed the Cloudera Data Warehouse (CDW) service in your ECS cluster, you must refresh the affected Database Catalogs and Virtual Warehouses from the CDW UI. Go to the CDW UI, click on the more vertical menu on the Database Catalog or Virtual Warehouse and click Refresh.

LDAP Bind DN	The Distinguished Name of the user to bind to LDAP for user authentication search/bind and group lookup for role authorization.	Distinguished Name (DN) example: CN=cdh admin,OU=svcaccount,DC=example,DC=com FreeIPA example: uid=username,cn=users,cn=accounts,dc=example,dc=com
LDAP Bind Password	The bind user password.	
LDAP User Search Base	The distinguished name indicating the path within the directory information tree from which to begin user searches.	AD example: cn=users,dc=example,dc=com LDAP example: ou=people,dc=example,dc=com FreeIPA example: cn=accounts,dc=example,dc=com
LDAP User Search Filter	The search filter to use for finding users.	AD example: (sAMAccountName={0}) LDAP example: (uid={0}) Note that a custom attribute can also be used if the directory is configured differently for user names. The {0} expands the currently authenticating user's name entered in the login form for the query. FreeIPA example: (&(uid={0})(objectClass=person))
LDAP Group Search Base	The distinguished name indicating the path within the directory information tree to begin group searches from.	cn=accounts,dc=example,dc=com

Property	Description	Sample values

LDAP Group Search Filter	The search filter to use for finding groups for authorization of authenticated users for their roles. You must configure this value such that only the groups associated with the user logging in are fetched from the IdP.
	There are two placeholders that can be used to match the groups of a user, {0} and {1}. {0} expands into the user DN and {1} expands into the username.
		For Active Directory and openLDAP compatible directories this will usually be (member={0}), where {0} will be replaced by DN string for a successfully authenticated user through the search/bind process. This requires configuration of the LDAP Bind User Distinguished Name field.
		AD example:
		`(member={0})`
		LDAP/FreeIPA example:
		`(&(member={0})(objectClass=posixgroup)(!(cn=admins)))`

Email Mapping Attribute	The LDAP attribute to be used for mapping the email in Identity Management. If no value is provided, mail is used as the default email mapping attribute. Email is a mandatory value in CDP. If no value is found for the email attribute, a value of {userame}@cdp.example is assumed for the user.	

5. Select Show Other Options and configure the following setting:

Property	Description	Sample values
Username Mapping Attribute	The LDAP attribute to be used for mapping the userId in Identity Management. **Important:** This property must be provided in order for job restore to work in CDE Virtual Clusters.	

Note: If the username and email attributes on the LDAP server contain spaces, Management Console includes the spaces when copying the corresponding attribute information.

6. If required, select Show Other Options and configure the following additional settings:

Property	Description	Sample values
LDAP User Bind Property	The property of the LDAP user object to use when binding to verify the password.	This value should always be set to dn.
Groupname Mapping Attribute	The LDAP attribute to be used for mapping the groupId in Identity Management.	
Group DN Property	The property of user object to use in {{dn}} interpolation of groupSearchFilter.	This value should always be set to dn.
First Name Mapping Attribute	The LDAP attribute to be used for mapping the first name attribute in Identity Management.	
Last Name Mapping Attribute	The LDAP attribute to be used for mapping the last name attribute in Identity Management.	

Note: If the username and email attributes on the LDAP server contain spaces, Management Console includes the spaces when copying the corresponding attribute information.

7. Click Test Connection to verify whether the LDAP information you have provided is valid.

   Management Console attempts a connection with the LDAP source based on the information provided, and returns a confirmation message if the connection is successful.
8. Click Save. The LDAP users are listed on the User Management page.

Modifying database properties

You can modify external database properties such as hostname, port, username, password, and database name using the Cloudera Management Console.

About this task

If your cluster was installed using the OpenShift Container Platform, you can view, but not modify the database properties. If the cluster was installed using the Embedded Container Service, you can edit the properties.

Important:

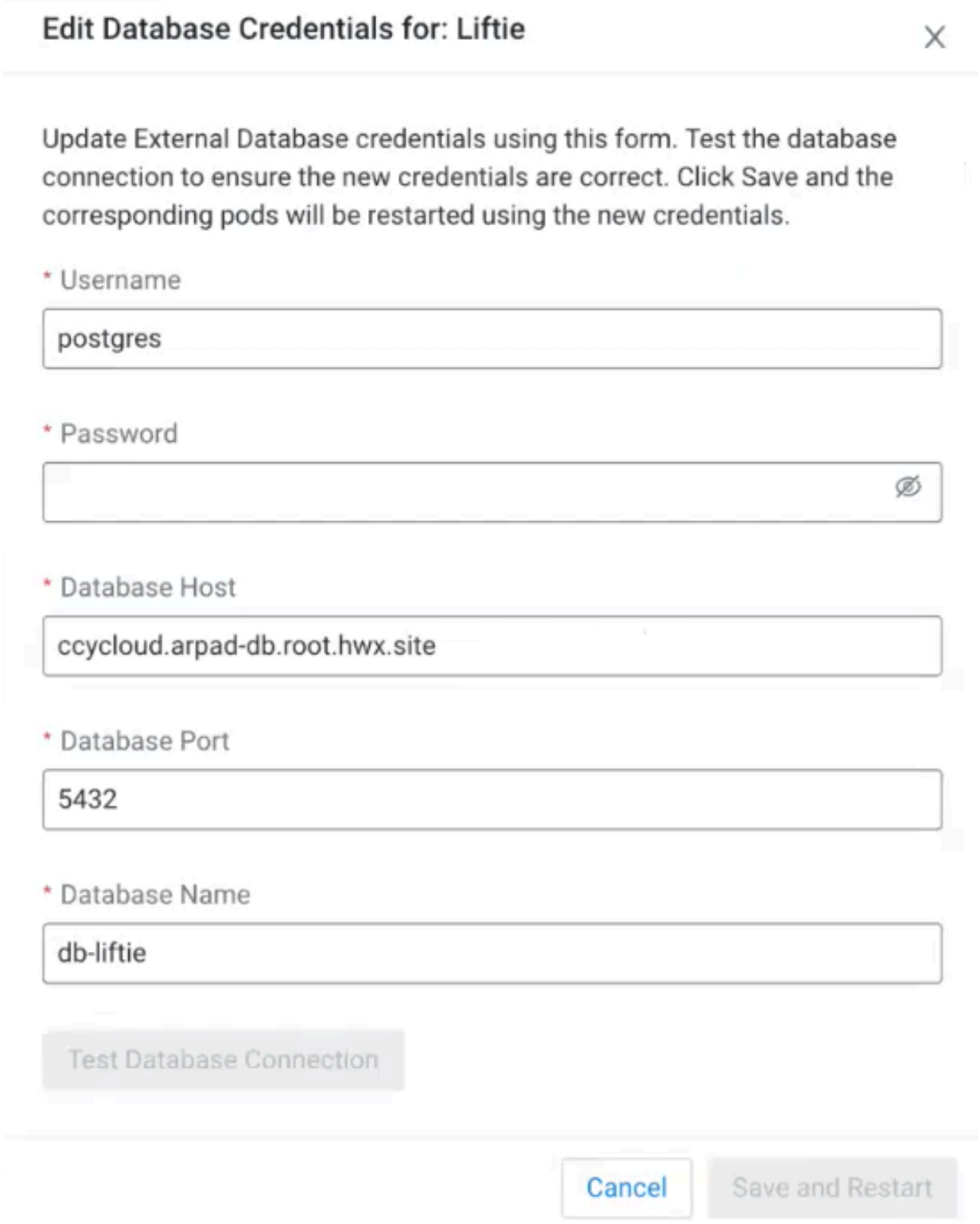
When updating the database password, if the User Management Service (UMS) and other CDP Private Cloud Data Services share the same database, then you must update the UMS database password first, followed by other services that share the database.

Procedure

1. Sign in to the CDP console.
2. Click Management Console.
3. On the Management Console home page, select Administration Databases to view the **Database administration** page.

   The **Database administration** page displays a list of databases.

4. Click located at the right side of each row, for the database you want to modify.
A dialog box appears where you can modify database properties. For example:



5. Modify any of the following properties:
   - Username
   - Password
   - Database Host
   - Database Port
   - Database Name
6. Click the Test Database Connection button to verify that CDP can connect to the database. If the test fails, re-check your modifications and click Test Database Connection again.
7. Click Save and Restart.
8. If you have deployed the Cloudera Data Warehouse (CDW) service in your ECS cluster, you must refresh the affected Database Catalogs and Virtual Warehouses from the CDW UI. Go to the CDW UI, click ⋮ on the Database Catalog or Virtual Warehouse and click Refresh.

Results
The database properties are updated.

Updating TLS certificates

From the Management Console of your CDP Private Cloud deployment, you can update TLS certificates that CDP uses to make secure connections with different types of services and workloads. For specified services, you can update the certificates whenever you rotate them.

Before you begin
You must ensure that all the services for which you want to update the certificates are TLS-enabled.

About this task

Important: The following procedure causes a restart of the dependent services after you have added the certificate. Therefore, Cloudera recommends that you plan for a down time before performing this procedure.

Procedure

1. Sign in to the CDP console.
2. Click Management Console.
3. On the Management Console home page, select Administration>CA Certificates.
4. From the CA Certificate Type dropdown list, select the type of service for which you want to upload a new TLS certificate.

   You can select from the following options for the types of secure connections:
   - Datalake: For secure connections with the CDP Private Cloud Base cluster services and Cloudera Manager.
   - Docker Registry: For a secure connection with the Docker Container registry containing the images for deployment.
   - External Database: For a secure connection with an external PostgreSQL database.
   - External Vault: For a secure connection with an external vault.
   - Miscellaneous: For a secure connection with services used during the installation and run time of CDP. For example, Custom Ingress, Custom Kubernetes API, and so on.

     Important: If your CDP Private Cloud deployment uses an external vault, then after updating the certificates for these services, you must reconfigure the certificates with the vault to ensure that it validates the certificates. For more information, see Updating custom certificates on page 14

5. Select the option to either browse and upload a certificate or directly enter the certificate details.

Important: The certificate must be in the X.509 PEM format and structured as follows:

----BEGIN CERTIFICATE-----..-----END CERTIFICATE----- .

Updating custom certificates

In a CDP Private Cloud deployment configured with an external vault, the CDP Management services such as Custom Ingress and Custom Kubernetes API authenticate to the vault with the help of the ServiceAccount's JSON Web Token (JWT) attached to the Kubernetes pod on which the services are running. The vault service validates the JWTs using the Kubernetes TokenReview API, and verifies the existence of the services. However, if the external service dependency being updated uses an entirely new CA certificate that CDP Private Cloud Data Services is not currently configured to trust, then that CA certificate should be updated in CDP Private Cloud Data Services first, then the certificate of the service dependency can be updated.

Before you begin

You must have updated the certificates for the services by using the Miscellaneous certificate type option mentioned in Updating TLS certificates on page 13.

Procedure

- Enter the following cURL statement to authenticate to the vault.

```
$ curl \
      --header "X-Vault-Token: <VAULT_TOKEN>" \
      --request POST \
      --data @payload.json \
      http://<VAULT_URL>/v1/auth/<KUBERNETES_PATH>/config
```

Here:

- <VAULT_TOKEN>: The privileged authorization token with write permissions on the vault.
- <KUBERNETES_PATH>: The path on which the Kubernetes login credentials information is mounted on the vault service. You can find this information specified as the value of the VAULT_AUTH_PATH property in the vault configmap. Ensure that this value follows the following naming convention: cloudera-<PROJECT_NAME>-<K8s-host> with the dot (.) replaced by an underscore (_).
- <VAULT_URL>: The URL of the vault service.

The payload.json contains the following information:

- kubernetes_host: The URL to access the Kubernetes API server from the vault service.
- token_reviewer_jwt: The JWT of the Kubernetes service account that the vault service uses to validate authentication requests from the Management Console services. The CDP Private Cloud installer creates a dedicated service named vault-auth for reviewing the requests.
- kubernetes_ca_cert: The CA certificate of the Kubernetes API server with newline characters replaced with '\n'.

The following example shows the contents of payload.json:

```
{
  "kubernetes_host": "https://api.examplehost.com:1111",
  "token_reviewer_jwt": "-----BEGIN CERTIFICATE-----\n.......\n-----END C
ERTIFICATE-----",
  "kubernetes_ca_cert": "-----BEGIN CERTIFICATE-----\n.......\n-----END CE
RTIFICATE-----"
}
```

For more information, see the vault documentation.

Configuring alerts for CDP Private Cloud

Management Console helps you configure alert receivers and rules to trigger automated notifications when specific events occur in your CDP Private Cloud deployment. You can view these alert notifications on the Management Console dashboard. Further, you can send the notifications to specified endpoints.

Configuring alert receivers

You can configure alert receivers on Management Console to trigger automated system-specific event notifications through external services such as emails, Slack channel messages, webhook notifications, PagerDuty messages, or SNMP traps. By configuring an alert receiver, you specify the details of an external service through which Management Console forwards the notification to the specified destination.

Before you begin

- You can configure alert receivers *only* if you have administrator privileges.
- Make sure you have the details of the external services for which you are configuring the alert receiver.
- If your CDP Private Cloud deployment is air gapped, you must have services available within your internal network that can help forward the notifications. For example, you must have one or more of the following services internally available: an SMTP server, a gateway or a proxy that can forward the notifications to external systems, a webhook endpoint that can externally forward the notifications, or an SNMP server.

  Further, you must configure your firewall policies to enable outgoing traffic to external systems.

Procedure

1. Sign in to the CDP console.
2. Click Management Console.
3. On the Management Console home page, select Administration>Alerts to view the Alerts page.
4. In the Alert Receivers section of the page, click Add Alert Receiver.
5. Configure the following options for adding an alert receiver in the pop-up window:

Note:

- You can configure *only one* alert receiver of type SNMP at a time.
- If your CDP Private Cloud Data Services cluster is deployed using the Experiences Compute Services (ECS), then Cloudera Manager automatically imports at the time of installation any non-default SMTP and SNMP alert configurations from the CDP Private Cloud Base cluster. Accordingly, alert receivers of type SNMP and SMTP are automatically configured when deploying Management Console for the first time.

Field	Description
Alert Scope	
Severity	The severity of the alert notifications to configure. You can select notifications of either Warning, or Critical, or both the types.
Source	The Management Console services for which you want to configure the alert notifications. You can select to configure notifications for the environments, or the control plane, or all the environments and control plane.

Field	Description
Receiver Type	The type of external service that receives the alert notification from Management Console. You can select from the following types: • Email • Slack • Webhook • PagerDuty • SNMP
Email	
Email To	The destination email address to which the alert notification email must be forwarded.
Email From	The source email address from where the alert notification email must be forwarded.
SMTP Server	The URL of the SMTP server through which the alert notification email is forwarded. You must specify the host name and port number as part of the SMTP server URL.
User name	The username to access the SMTP server.
Password	The password to access the SMTP server.
Connection Requires TLS	Toggle this option to On if you have a secure connection configured between your CDP Private Cloud deployment and the SMTP server such that the SMTP server's certificate is trusted by CDP Private Cloud.
Slack	
API URL	The URL of the Slack API associated with the channel to which the notification must be sent.
Channel	The name of the Slack channel. You can specify any name here for your reference because the Slack API derives the actual name from the specified URL.
Webhook	
URL	The URL of the webhook endpoint to which the alert notification must be sent.
Basic Auth User Name	The username for basic authentication to the webhook application.
Basic Auth Password	The password for basic authentication to the webhook application.
Bearer Token	The authentication header in case of using a bearer token.
Skip TLS Verification (Insecure)	Toggle this option to On if you have a secure connection configured between your CDP Private Cloud deployment and the webhook application. For the secure connection, you must use a certificate trusted by CDP Private Cloud.
PagerDuty	
URL	The URL of the PagerDuty platform to which the notification must be sent.
Routing Key	The PagerDuty Events API v2 integration key. For details, see PagerDuty Services and Integrations.
SNMP	
SNMP NMS Hostname	The DNS name or IP address of the SNMP Network Management Software (NMS) host listening for SNMP traps or notifications.
SNMP Server Port	The port number on which the SNMP server is listening for traps or notifications.
SNMP Retry Count	The maximum number of times to try an SNMP trap before the latter times out. If you specify '0', the trap is sent only once.

Field	Description
SNMP Timeout (Milliseconds)	The time, in milliseconds, to wait after which an SNMP trap times out.
SNMP Security Level	The level of security to use if you the select SNMPv3 protocol for the alert receiver. You can select either authNoPriv or noauthNoPriv security level. The details that you need to specify vary with the security level that you select. You can also select the SNMPv2 protocol, if required.
SNMPv2 Community String	The community string for identifying the SNMP service. Generated SNMPv2 traps use this string for authentication purposes. Specify this value if you select SNMPv2 as the SNMP Security Level.
SNMP Authentication Protocol	The authentication algorithm. The available options are MD5 and SHA. Specify this value if you select authNoPriv as the SNMP Security Level.
SNMP Server Engine Id	Used along with the pass phrase to generate keys for authentication and privacy protocols. The Engine ID is a hexadecimal number. Specify this value if you select authNoPriv as the SNMP Security Level.
SNMP Security Username	The name of the user to add for SNMP security. Specify this value if you select either authNoPriv or noauthNoPriv as the SNMP Security Level.
SNMP Authentication Protocol Pass Phrase	The pass phrase to use for the SNMP authentication protocol. Specify this value if you select authNoPriv as the SNMP Security Level.
Test receiver parameters	
Send Test Event	After providing the details to configure the external service, you can send a test notification to verify whether the message reaches the desired destination.

6. Click Add Alert Receiver.

   The Alert Receivers page now displays the details of the receiver that you just added.

7. Repeat steps 4 and 5 to add more alert receivers.

Note: If you want to edit the details of an alert receiver, select Edit Alert Receiver from the drop-down menu against that receiver entry, and update the desired fields. You can access the drop-down menu by clicking the vertical ellipsis (three dots) against the particular receiver entry on the table.

Similarly, if you want to delete an alert receiver, select Delete Alert Receiver, from the drop-down menu against that receiver entry.

Configuring alert rules

You can define alert rules for your CDP Private Cloud Data Services deployment based on PromQL expressions. The alerts are automatically triggered when specific events occur in your deployment. You can view the triggered alerts on the Management Console dashboard. Further, any alert receivers that you have already configured start sending notifications to specified endpoints.

Before you begin

You can configure alert rules *only* if you have administrator privileges.

About this task

Management Console supports two types of alert rules: built-in and custom.

- Built-in alert rules are system-generated, and therefore, you cannot add or remove them. You can only enable, disable, or edit certain details associated with them.

- You can create custom alert rules based on your requirements. Further, you can edit, delete, enable, or disable them.

Procedure

1. Sign in to the CDP console.
2. Click Management Console.
3. On the Management Console home page, select Administration>Alerts to view the Alerts page.
4. In the Alert Rules section of the page, click Add Alert Rule.
5. Configure the following options for adding an alert rule in the pop-up window:

Field	Description
Name	The name of the alert rule. You cannot use spaces or special characters in the name.
Severity	The severity of the alert rules to configure. You can select rules of severity Critical or Warning.
Enable Alert	Select this checkbox to ensure that the alert rule is enabled at the time of creation.
Message	The text of the alert rule. You can use PromQL labels to denote entities such as jobs in the text. For more information about using PromQL labels, see Alerting Rules.
Summary	Overview text of the alert rule.
For Clause	The duration for which the PromQL expression must be true. If the expression continues to be true after the specified duration, then the configured alert is automatically triggered.
Source	The Management Console services for which you want to configure the alert rules. You can select one of the following options as the source: • All Environments and Control Plane • Environments • Control Plane • A specific environment from the list of configured environments
Workload Type	For a selected source, the type of workload to which the alert rule applies. You can select workloads of type Data Warehouse, Machine Learning, Infrastructure, or all these types.
PromQL Expression	The query expression in PromQL. The alert is issued when this expression is true for the time period specified in For Clause. **Important:** Metrics reported by the environments always contain the following labels: appId and appName. Therefore, the result of the alert rule's query expression also must contain these labels. To ensure that the result contains the labels, include the by (appId, appName) clause when using aggregation operators in the query expression. For example, instead of the count(my_metric) > 0 expression, use the count(my_metric) by (app Id, appName) > 0 expression.
Generated Query	The query that is generated for a selected workload type depending on the specified PromQL expression. You can view the query by clicking Show Generated Query.

Field	Description
Test PromQL Expression	You can click this option to test the query expression generated for the combination of a selected source and workload type. If you select one of All Environments and Control Plane, Environments, or Control Plane as the source for the PromQL query, it runs *only* on the control plane. To run the query on an environment, you must select a specific environment as the source. **Note:** If you click this option *before* saving the alert rule and if the PromQL expression is invalid, then an unexpected error appears. In addition, you might lose all the information entered for configuring the alert rule. Therefore, you must save the alert rule and then test the PromQL expression.

6. Click Add Alert Rule.

   The page now displays the details of the alert rule that you just added.
7. Repeat steps 4 and 5 to add more alert rules.

Additional operations on the alert rules

You can perform different operations on the added alert rules from the Alerts page.

- If you want to edit the details of an alert rule, select Edit Alert Rule from the drop-down menu against that rule entry, and update the desired fields. You can access the drop-down menu by clicking the vertical ellipsis (three dots) against the particular rule entry on the table.

  The fields that you can edit vary with the type of the alert rule. For example, you can edit only the Severity, For Clause fields, and select or clear the Enable Alert checkbox for a built-in alert rule. However, you can edit almost all the available fields for a custom alert rule.
- If you want to enable an alert rule or disable an already enabled rule without opening the pop-up window, select Enable Alert Rule or Disable Alert Rule from the drop-down menu against that rule depending on your requirement.
- If you want to delete a custom alert rule, select Delete Alert Rule from the drop-down menu against that rule.

Configuring proxy hosts for CML workspace connections

If your CDP Private Cloud deployment uses network proxy, you can configure proxy hosts that the workloads can use for connections with Cloudera Machine Learning (CML) workspaces. You can configure the proxy configuration values from the Management Console.

About this task

The settings that you configure with the help this procedure reflect in newly provisioned CML workspaces *only* in those CDP Private Cloud Data Services that are deployed using the Embedded Container Service (ECS). In a deployment using an OpenShift cluster, the default values are used.

Procedure

1. Sign in to the CDP console.
2. Click Management Console.
3. On the Management Console home page, select Administration>Networks to view the Networks page.

4. Configure the following options for the proxy values:

Field	Description
HTTPS Proxy	The HTTP or HTTPS proxy connection string for use in connections with CML workspaces. You must specify this connection string in the form: https://<username>:<password>@<host>:<port>.  **Note:** Both <username> and <password> are optional parameters. You can specify the connection proxy string without these parameters.
HTTP Proxy	The HTTP or HTTPS proxy connection string for use in connections with CML workspaces. You must specify the connection string in the form: http://<username>:<password>@<host>:<port>  **Note:** Both <username> and <password> are optional parameters. You can specify the connection proxy string without these parameters.
No Proxy	Comma-separated list of hostnames, IP addresses, or hostnames and IP addresses that should not be accessed through the specified HTTPS or HTTP proxy URLs. In the case of CDP Private Cloud Data Services deployments that use ECS, must include URLs for the following: • All the ECS hosts in your deployment • Any CDP Private Cloud Base cluster that you want to access • CIDR IP addresses for internal operations in the ECS cluster: 10.42.0.0/16 and 10.43.0.0/16

5. Click Save.

Proxy setting best practices

Learn about best practices for using environment variables to propagate proxy settings.

In Kubernetes, proxy settings can be propagated to pods through the use of environment variables in pod spec configurations or through ConfigMaps or Secrets that are mounted as volumes within pods. This allows the proxy settings to be passed down to individual containers within the pods. However, it's important to note that not all applications may automatically inherit these settings, and some may require additional configuration within the container image or application code to properly utilize the proxy settings.

It's important to note that the use of no_proxy or NO_PROXY environment variables to bypass proxy settings may not be consistently respected by all third-party libraries or applications. While some libraries or applications may automatically honor these settings, others may not.

Many popular libraries and frameworks in various programming languages, such as Python, Java and Node.js, have their own way of handling proxy settings, which may not necessarily rely on the no_proxy or NO_PROXY environment variables. These libraries may have their own configuration files or internal settings that dictate how they handle proxy configurations, and these settings may not always align with the no_proxy or NO_PROXY environment variables set at the system or project level.

As a result, it's important to be aware that relying solely on no_proxy or NO_PROXY environment variables may not provide consistent results across all libraries or applications used in a project. In some cases, it may be necessary to clear out the http_proxy, https_proxy, no_proxy, or NO_PROXY environment variables in your project's environment variables or configuration files to ensure that the third-party libraries or applications do not attempt to apply proxy settings at all.

To ensure that proxy settings are consistently respected across all libraries and applications used in a project, it's recommended to carefully review the documentation and configuration options of each library or application, and configure them accordingly.

However, if you ever have to manage a stack written in multiple languages, you might need to consider some best practices for setting HTTP proxy configurations:

For http_proxy and https_proxy:

- Use lowercase form. HTTP_PROXY is not always supported or recommended.
- If you absolutely must use the uppercase form as well, be sure both versions share the same value.

For no_proxy:

- Use lowercase form.
- Use comma-separated hostname:port values.
- IP addresses are acceptable, but hostnames are never resolved.
- Suffixes are always matched (for example, example.com will match test.example.com).
- If top-level domains need to be matched, avoid using a leading dot (.).
- Avoid using CIDR matching since only the Go and Ruby languages support that.

PIP installs with Proxy

When using pip to install packages from external sources via a proxy, it is generally recommended to perform this installation in a separate session (project session), utilizing the existing proxy rules. Proxy settings, including proxy server addresses, usernames, passwords, and other configurations, are typically environment-specific. By using a separate session, you can ensure that the appropriate proxy settings are used for the specific installation task. This helps to avoid conflicts or misconfigurations with your main session's proxy settings, which may be required for other tasks or applications.

To resolve Python communication issues between pods with proxy setup, in Project Settings Advanced add the four environmental vatiables (HTTP_PROXY, http_proxy, HTTPS_PROXY, https_proxy) set to empty values. This will allow Python to run and use kubedns and kubeproxy properly.

Environment variables in the customer environment:

- HTTP_PROXY
- http_proxy
- HTTPS_PROXY
- https_proxy
- NO_PROXY
- no_proxy
- ALL_PROXY

The workaround of adding these environment variables in the project session helps resolve the proxy issue. Since Python does not support CIDR blocks for no_proxy, the request automatically gets directed to http_proxy or htpps_proxy, which causes the failure and prevents jobs from completing.

Related Information

We need to talk: Can we standardize NO_PROXY?

Configuring proxy hosts for CML workspace connections

Installing a non-transparent proxy in a CML environment