

Deployment Guide for HCL CDP v12.1.8 Marketing Automation Component on OpenShift



Contents

- Chapter 1. Introduction..... 3**
- Chapter 2. Prerequisites..... 4**
- Chapter 3. List of components for Marketing Automation modules..... 5**
- Chapter 4. Deployment Process..... 6**
 - Deploying CDP Trigger.....6
 - Deploying HCL CDP Core..... 11
 - Deploying CDP Flip..... 17
 - Deploying HCL CDP SMSEmailSender..... 23
 - Deploying HCL CDP ExternalAPI.....30
 - Deploying HCL CDP Tryitout..... 34

Chapter 1. Introduction

This document provides a comprehensive guide to installing and configuring the necessary services and components for deploying HCL CDP Marketing Automation on the Red Hat OpenShift platform using Devtron.

Chapter 2. Prerequisites

To deploy the micro-services based HCL CDP Marketing Automation components on a containerized platform using the Devtron CI/CD tool (with Red Hat OpenShift as the chosen container orchestration platform), ensure the following prerequisites are met:

1. Red Hat OpenShift Setup:

Verify that Red Hat OpenShift is installed and properly configured. For installation details, refer to the official Red Hat OpenShift documentation.

2. Required Tools:

The following tools must be available to facilitate the installation process:

- **OpenShift CLI:** Ensure the `oc` command is accessible and functional.
- **Kubernetes CLI:** Verify the availability of the `kubectl` command.
- **Helm CLI:** Ensure the `helm` command-line tool is installed.
- **Devtron Deployment Tool:** Confirm that Devtron is set up for managing deployments.

Chapter 3. List of components for Marketing Automation modules

The following components must be deployed and configured on the OpenShift cluster to support the deployment of HCL CDP Marketing Automation modules.

These components provide the essential functionalities required for end-to-end operations of the CDP:

Application/Services	Deployment Type
CDP Trigger	Helm chart
CDP Core	Helm chart
CDP Flip	Helm chart
CDP SMSEmailSender	Helm chart
CDP ExternalAPI	Helm chart
CDP Tryitout	Helm chart

Chapter 4. Deployment Process

This section provides step-by-step instructions for deploying each service, component, and module required for HCL CDP Marketing Automation.

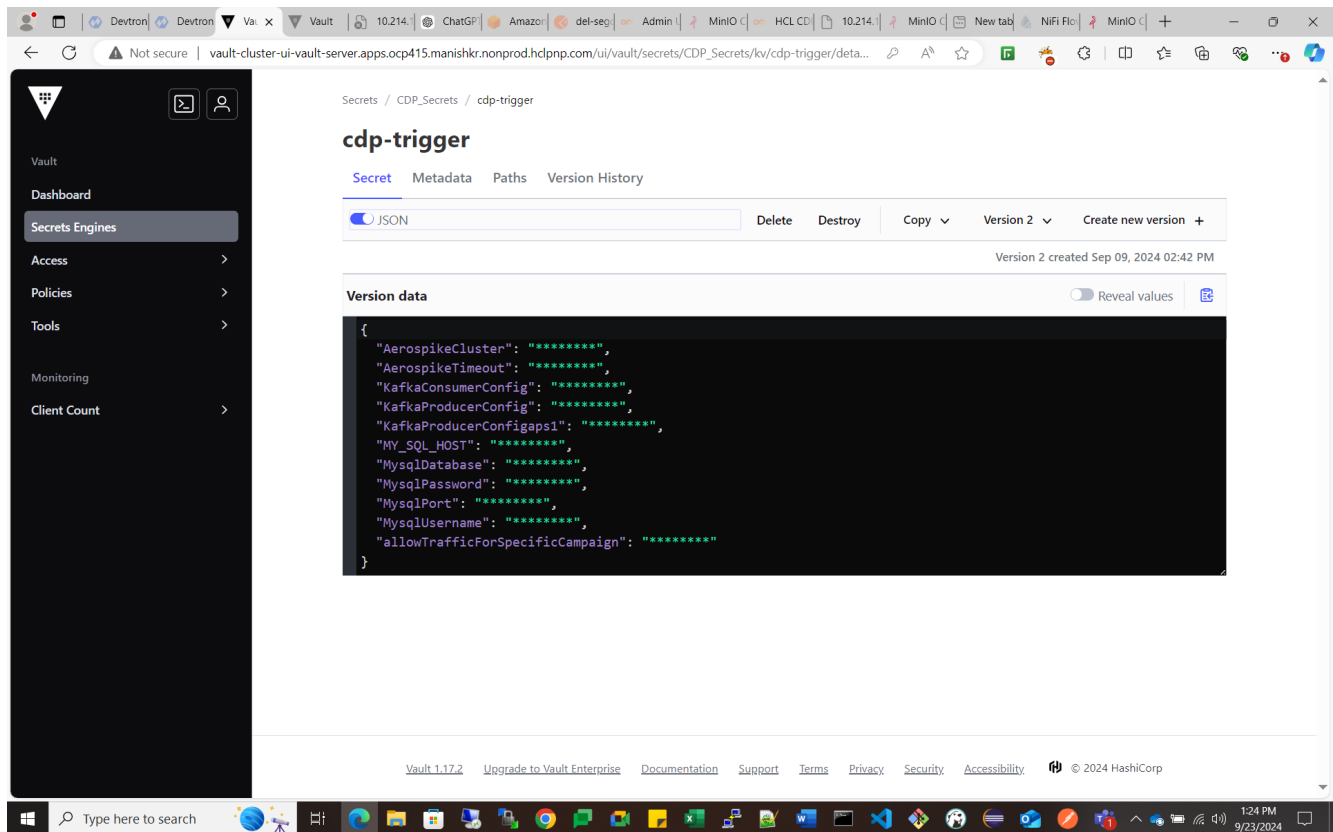
Follow the guidelines in each subsection to ensure the successful deployment and configuration of the respective elements.

Deploying CDP Trigger

This section provides detailed instructions on how to deploy HCL CDP Trigger using the Devtron in the OpenShift.

Prerequisites

Make sure to create cdp-trigger secret in HashiCorp vault before deploying CDP Trigger.



To create CDP Trigger secret in the HashiCorp vault, follow the steps below:

1. Create a cdp-trigger secret sample key and value, and update the actual values as shown below.

```
{
  "AerospikeCluster": "<AerospikeCluster>",
  "AerospikeTimeout": "10000",
  "KafkaConsumerConfig": "<KafkaConsumerConfig>",
  "KafkaProducerConfig": "<KafkaProducerConfig>",
  "KafkaProducerConfigaps1": "<KafkaProducerConfigaps1>",
}
```

```

"MY_SQL_HOST": "<MY_SQL_HOST>",
"MysqlDatabase": "<MysqlDatabase>",
"MysqlPassword": "<MysqlPassword>",
"MysqlPort": "<MysqlPort>",
"MysqlUsername": "<MysqlUsername>",
"allowTrafficForSpecificCampaign": "false"
}

```

2. Create a Kubernetes Secret with Vault credentials to fetch the secrets from the vault.

```

apiVersion: v1
kind: Secret
metadata:
  name: vault-creds
  namespace: cdp-dev-app
type: Opaque
data:
  username: <vault username>
  password: <vault password>

```

3. Create separate service account cdp-dev-app-sa with appropriate roles and permissions, and update ConfigMaps data with actual values.
4. Create required Kafka topics before deployment. A sample Kafka topic configuration is shown below.

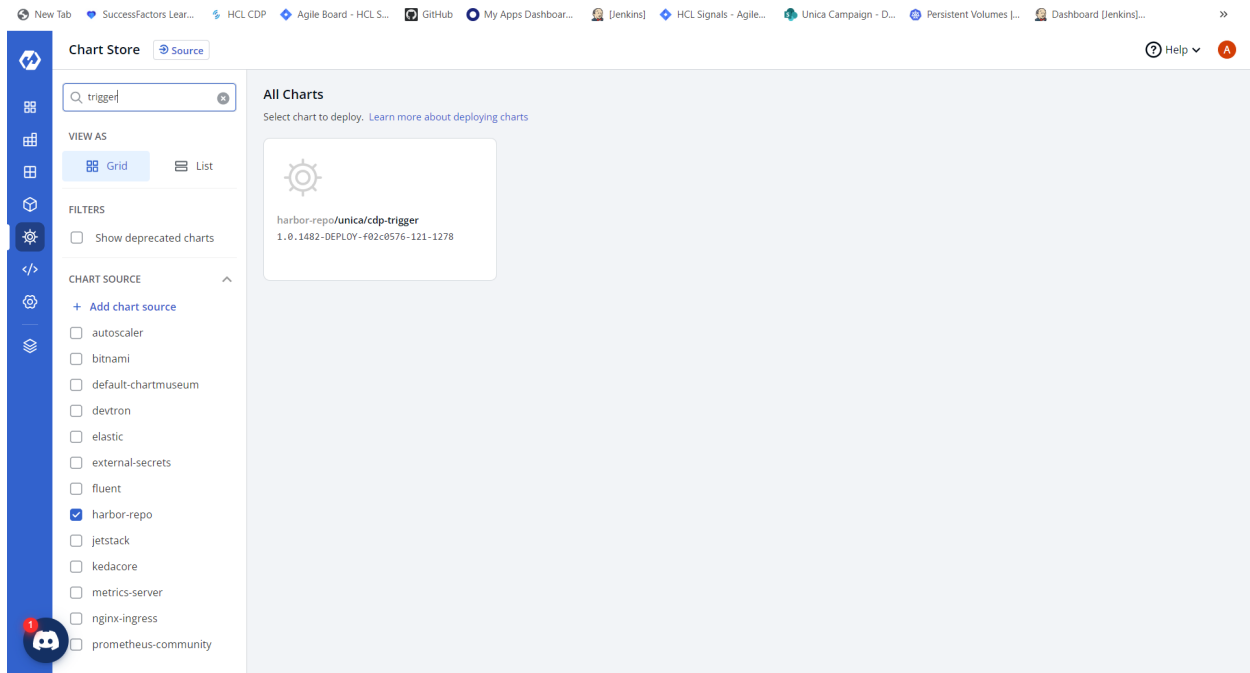
The screenshot shows the 'Topics / Create' page in a Kafka management console. The left sidebar contains navigation links: Dashboard, hclsw-sbi-dev..., Brokers, Topics (selected), Consumers, and ACL. The main form includes the following fields and options:

- Topic Name ***: analyze_post
- Number of Partitions ***: 12
- Cleanup policy**: Delete (dropdown menu)
- Min In Sync Replicas**: 1
- Replication Factor**: 2
- Time to retain data (in ms)**: 86400000 (with a 1d selector)
- Time to retain data (in ms)**: 12 hours, 1 day (selected), 2 days, 7 days, 4 weeks
- Max size on disk in GB**: Not Set (dropdown menu)
- Maximum message size in bytes**: Maximum message size
- Custom parameters**: + Add Custom Parameter
- Buttons**: Cancel, Create topic

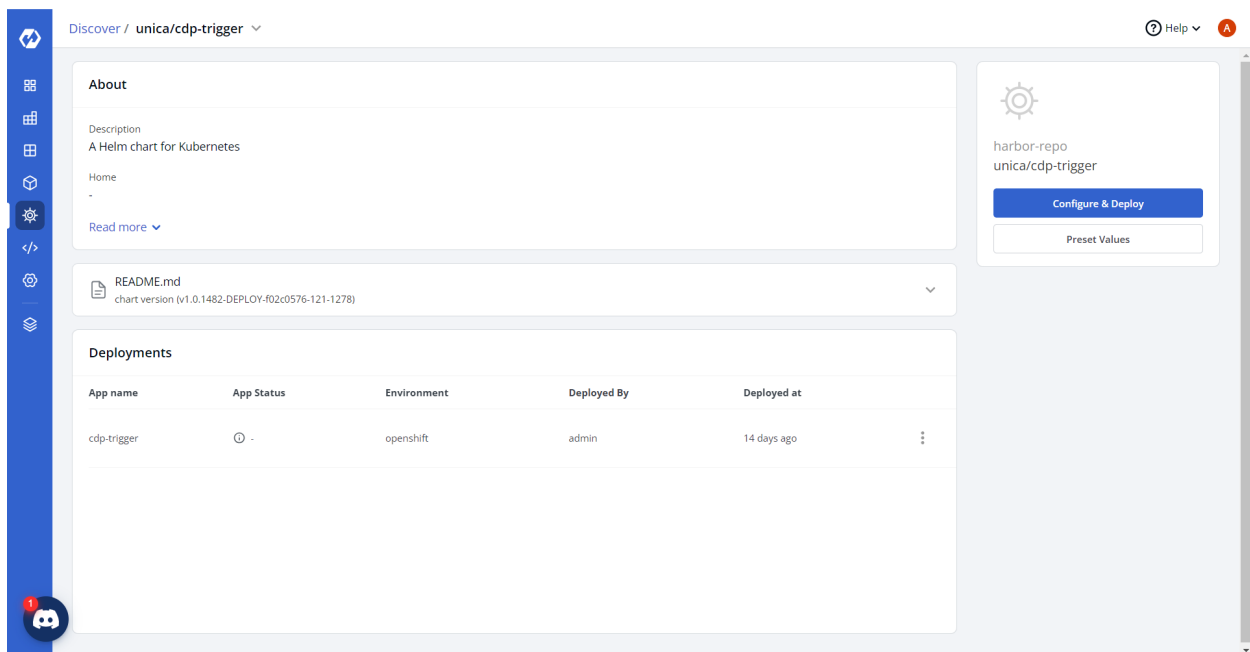
Deploy CDP Trigger

To deploy CDP Trigger using Devtron in Openshift platform, follow the steps below:

1. Navigate to the Devtron **Chart Store**, and select cdp-trigger chart to deploy.



2. Now, configure and deploy the cdp-trigger charts.



3. In the YAML section, update the ConfigMap using the below details and deploy the chart.

```
APP_INFRA: k8s
APP_REGION: ap-south-1
APP_TYPE: trigger--mu
AWSRegion: us-east-1
```

```

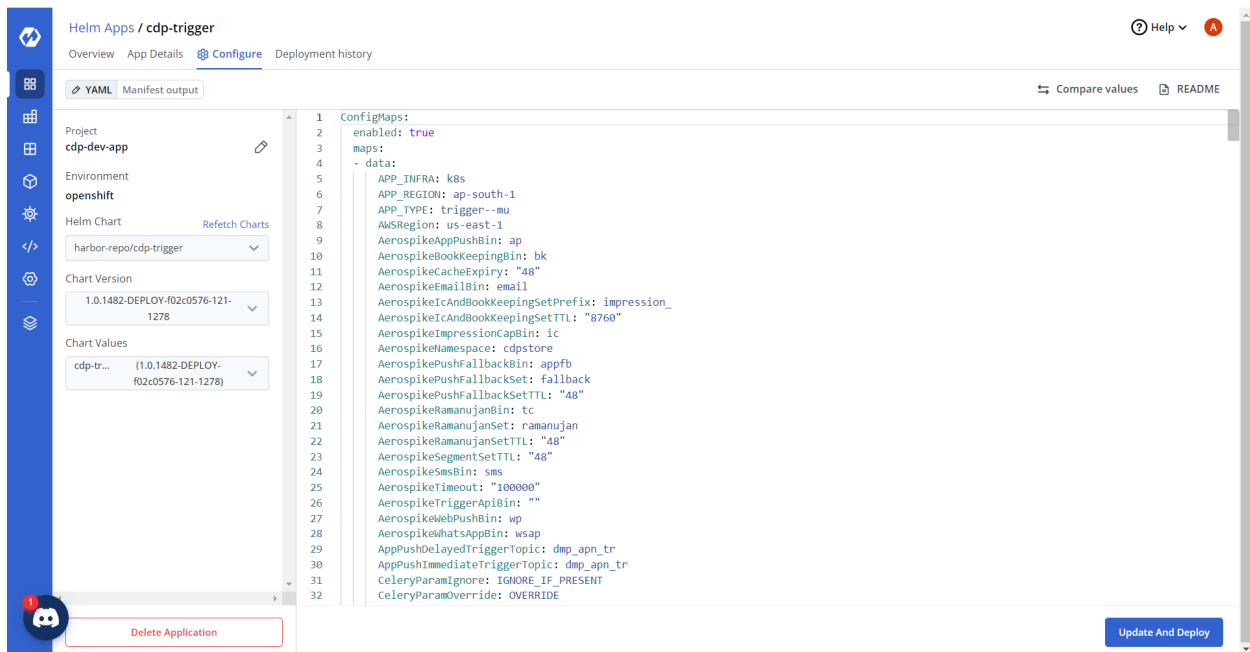
AerospikeAppPushBin: ap
AerospikeBookKeepingBin: bk
AerospikeCacheExpiry: "48"
AerospikeEmailBin: email
AerospikeIcAndBookKeepingSetPrefix: impression_
AerospikeIcAndBookKeepingSetTTL: "8760"
AerospikeImpressionCapBin: ic
AerospikeNamespace: cdpstore
AerospikePushFallbackBin: appfb
AerospikePushFallbackSet: fallback
AerospikePushFallbackSetTTL: "48"
AerospikeRamanujanBin: tc
AerospikeRamanujanSet: ramanujan
AerospikeRamanujanSetTTL: "48"
AerospikeSegmentSetTTL: "48"
AerospikeSmsBin: sms
AerospikeTimeout: "100000"
AerospikeTriggerApiBin: ""
AerospikeWebPushBin: wp
AerospikeWhatsAppBin: wsap
AppPushDelayedTriggerTopic: dmp_apn_tr
AppPushImmediateTriggerTopic: dmp_apn_tr
CeleryParamIgnore: IGNORE_IF_PRESENT
CeleryParamOverride: OVERRIDE
CeleryParamTrigger: TRIGGER
CeleryTimeThresholdMinutes: "30"
CeleryTriggerTopic: CeleryETALem
EXECUTOR_SERVICE_MAX_THREADS: "4"
EXECUTOR_SERVICE_MIN_THREADS: "2"
EXECUTOR_SERVICE_QUEUE_SIZE: "100"
EXECUTOR_SERVICE_THREAD_TTL: "5"
EmailDelayedTriggerTopic: dmp_email_tr
EmailImmediateTriggerTopic: dmp_email_tr
InboundKafkaTopic: trigger_inbound
KAFKA_CONSUMER_GROUP_ID: -trigger
KAFKA_POLL_INTERVAL: "200"
KAFKA_PRODUCER_EVENT_TYPE: "5656"
KAFKA_PRODUCER_META: "true"
KAFKA_PRODUCER_POOL_SIZE: "10"
KAFKA_TOPIC_CONSUMER_BATCH_SIZE: 10,10,10
KAFKA_TOPIC_CONSUMER_DEFAULT_BATCH_TIMEOUT_MS: 1000,1000,1000
KAFKA_TOPIC_CONSUMER_DEFAULT_BATCH_TIMEOUT_TRIGGER: 5000,5000,5000
KAFKA_TOPIC_CONSUMERS: 6,6,6
KAFKA_TOPIC_LIST: dmp_rt_segments,ETANotify,dmp_journey_trigger
KafkaCeleryNotify: ETANotify
KafkaJourneyTopic: dmp_journey_trigger
KafkaRTSegment: dmp_rt_segments
KafkaRTSegmentOffline: dmp_rt_segments_offline
LOG_LEVEL_APP: debug
LOG_LEVEL_METRICS: info
LOG_LEVEL_ROOT: info
LOG_LEVEL_UTILS: info
MAX_BATCH_EXECUTE_RETRIES: "3"
MAX_EVENT_BATCH_LAG: "8"
MysqlMapRefreshIntervalMinute: "1"
OPEN_TSDB_HOST: <OPEN_TSDB_HOST>
OPEN_TSDB_HOST_TO_BE_ADDED: "true"
OPEN_TSDB_PORT: "80"

```

```

ReportingIntervalInMin: "55"
RtsSegOfflineRefreshDsName: rtsegoofflineRefresh
RtsSegOnlineRefreshDsName: rtsegonlineRefresh
RtsSegRefreshDsName: rtsegrefresh
SmsDelayedTriggerTopic: dmp_sms_tr
SmsImmediateTriggerTopic: dmp_sms_tr
TriggerApiDelayedTriggerTopic: dmp_trapi_tr
TriggerApiImmediateTriggerTopic: dmp_trapi_tr
UserFailureKafkaTopic: trigger_failure
WebPushDelayedTriggerTopic: dmp_wpn_tr
WebPushImmediateTriggerTopic: dmp_wpn_tr
WhatsAppDelayedTriggerTopic: dmp_wsap_tr
WhatsAppImmediateTriggerTopic: dmp_wsap_tr
alertToEmailId: ""
emailType: SMTP
fromEmailId: ""
hostname: trigger-prod-mu
smtpHost: <OPEN_TSDB_HOST>
smtpPort: "<smtpPort>"

```



The screenshot shows the Helm Apps / cdp-trigger configuration page. The left sidebar contains navigation links: Overview, App Details, Configure (selected), and Deployment history. Below these are tabs for YAML and Manifest output. The main content area displays the configuration for the cdp-trigger application, including the ConfigMaps section with a list of key-value pairs.

Project: cdp-dev-app

Environment: openshift

Helm Chart: harbor-repo/cdp-trigger

Chart Version: 1.0.1482-DEPLOY-f02c0576-121-1278

Chart Values: cdp-tr... (1.0.1482-DEPLOY-f02c0576-121-1278)

ConfigMaps:

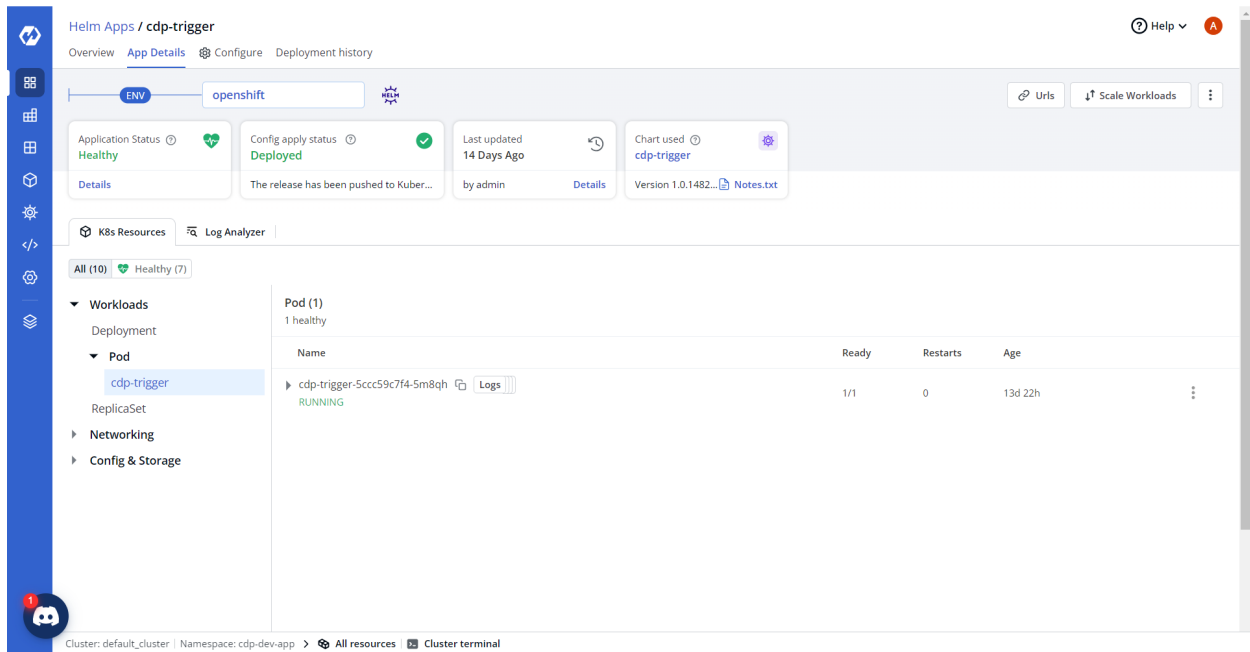
```

1 enabled: true
2 maps:
3   - data:
4     APP_INFRA: k8s
5     APP_REGION: ap-south-1
6     APP_TYPE: trigger--mu
7     AWSRegion: us-east-1
8     AerospikeAppPushBin: ap
9     AerospikeBookKeepingBin: bk
10    AerospikeCacheExpiry: "48"
11    AerospikeEmailBin: email
12    AerospikeICAndBookKeepingSetPrefix: impression_
13    AerospikeICAndBookKeepingSetTTL: "8760"
14    AerospikeImpressionCapBin: ic
15    AerospikeNamespace: cdpstore
16    AerospikePushFallbackBin: appfb
17    AerospikePushFallbackSet: fallback
18    AerospikePushFallbackSetTTL: "48"
19    AerospikeRamanujanBin: tc
20    AerospikeRamanujanSet: ramanujan
21    AerospikeRamanujanSetTTL: "48"
22    AerospikeSegmentSetTTL: "48"
23    AerospikeSmsBin: sms
24    AerospikeTimeout: "180000"
25    AerospikeTriggerApiBin: ""
26    AerospikeWebPushBin: wp
27    AerospikeWhatsAppBin: wsap
28    AppPushDelayedTriggerTopic: dmp_apn_tr
29    AppPushImmediateTriggerTopic: dmp_apn_tr
30    CeleryParamIgnore: IGNORE_IF_PRESENT
31    CeleryParamOverride: OVERRIDE
32

```

Buttons at the bottom: Delete Application, Update And Deploy.

4. On successful deployment, validate the deployment as shown below.

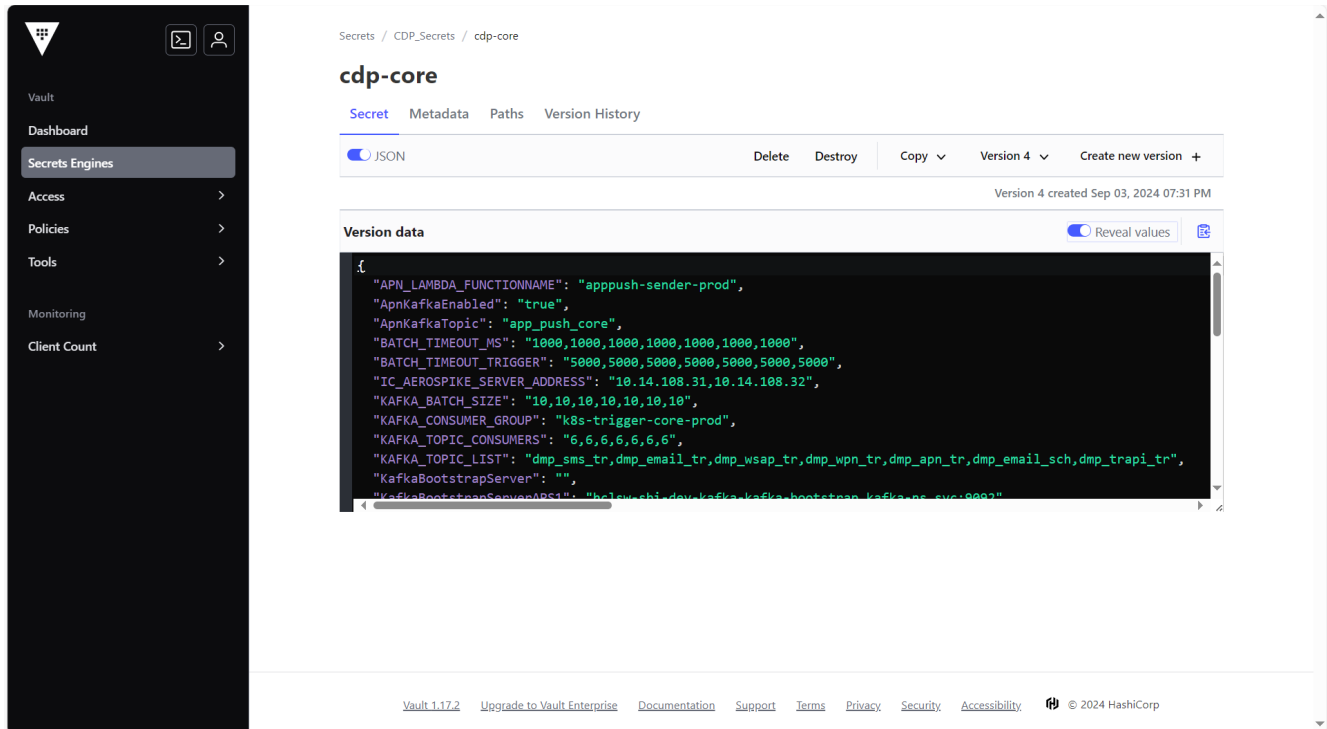


Deploying HCL CDP Core

This section provides detailed instructions on how to deploy HCL CDP Core using the Devtron in the OpenShift.

Prerequisites

- Make sure to create cdp-core secret in HashiCorp vault before deploying HCL CDP Trigger.



To create CDP Core secret in the HashiCorp vault, follow the steps below:

1. Create a cdp-core secret sample key and value, and update the actual values as shown below.

```
{
  "APN_LAMBDA_FUNCTIONNAME": "apppush-sender-prod",
  "ApnKafkaEnabled": "true",
  "ApnKafkaTopic": "app_push_core",
  "BATCH_TIMEOUT_MS": "1000,1000,1000,1000,1000,1000,1000",
  "BATCH_TIMEOUT_TRIGGER": "5000,5000,5000,5000,5000,5000,5000",
  "IC_AEROSPIKE_SERVER_ADDRESS": "10.14.108.31,10.14.108.32",
  "KAFKA_BATCH_SIZE": "10,10,10,10,10,10,10",
  "KAFKA_CONSUMER_GROUP": "k8s-trigger-core-prod",
  "KAFKA_TOPIC_CONSUMERS": "6,6,6,6,6,6,6",
  "KAFKA_TOPIC_LIST":
"dmp_sms_tr,dmp_email_tr,dmp_wsap_tr,dmp_wpn_tr,dmp_apn_tr,dmp_email_sch,dmp_trapi_tr",
  "KafkaBootstrapServer": "",
  "KafkaBootstrapServerAPS1": "<KafkaBootstrapServerAPS1>",
  "KafkaBootstrapServerUSE1": "",
  "KafkaConsumerConfig": "<KafkaConsumerConfig>",
  "KafkaPassword": "<KafkaPassword>",
  "KafkaPasswordAPS1": "<KafkaPassword>",
  "KafkaPasswordUSE1": "",
  "KafkaProducerConfig": "<KafkaProducerConfig>",
  "KafkaProducerConfigaps1": "<KafkaProducerConfigaps1>",
  "KafkaProducerHostName": "trigger-core-us",
  "KafkaProducerRegion": "",
  "KafkaUsername": "<KafkaUsernameAPS1>",
  "KafkaUsernameAPS1": "<KafkaUsernameAPS1>",
  "KafkaUsernameUSE1": "",
  "LIVSPACE_IC_AEROSPIKE_SERVER_ADDRESS": "<LIVSPACE_IC_AEROSPIKE_SERVER_ADDRESS>",
  "MAX_EVENT_BATCH_LAG": "32",
}
```

```

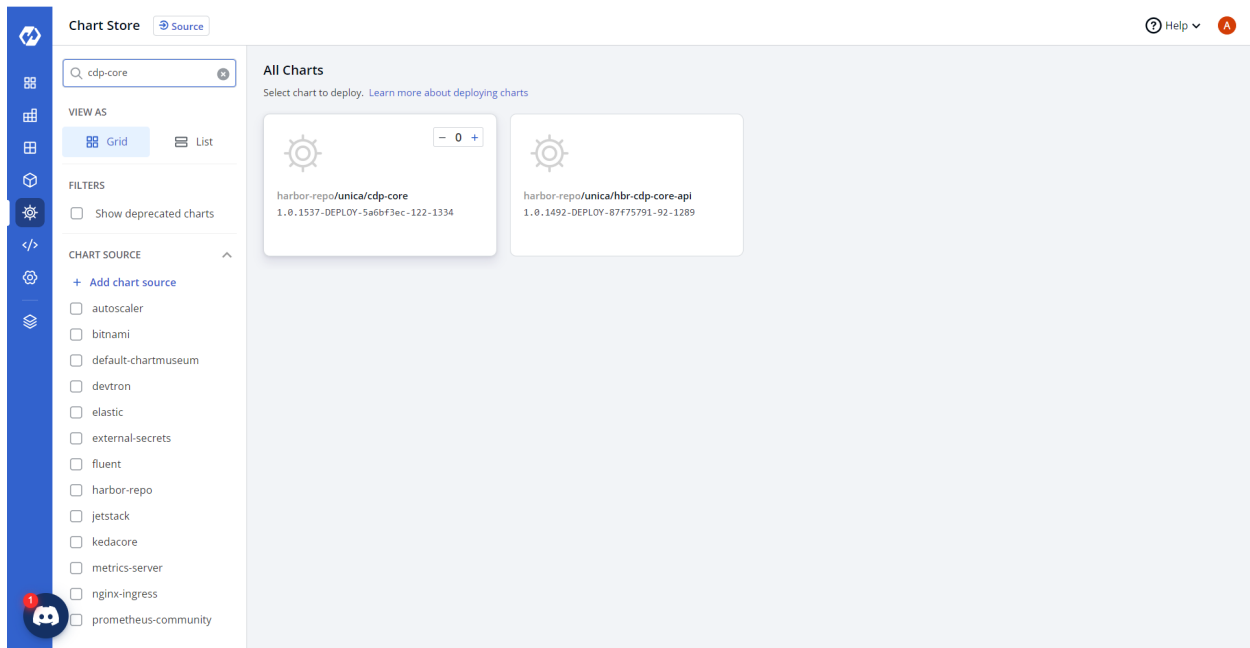
"MY_SQL_DB": "vrm?autoReconnect=true ",
"MY_SQL_HOST": "<MY_SQL_HOST>",
"MY_SQL_PORT": "<MY_SQL_PORT>",
"NBA_AEROSPIKE_SERVER_ADDRESS": "<NBA_AEROSPIKE_SERVER_ADDRESS>",
"SesEnabled": "true",
"TSDB_HOST": "<TSDB_HOST>",
"USE_RTS_MS": "false",
"VRM_DB_PASSWORD": "<VRM_DB_PASSWORD>",
"VRM_DB_USERNAME": "<VRM_DB_USERNAME>",
"WPN_LAMBDA_FUNCTIONNAME": "web-push-sender",
"WpnKafkaEnabled": "true",
"WpnKafkaTopic": "web_push_core",
"allowTrafficForSpecificCampaign": "false"
}

```

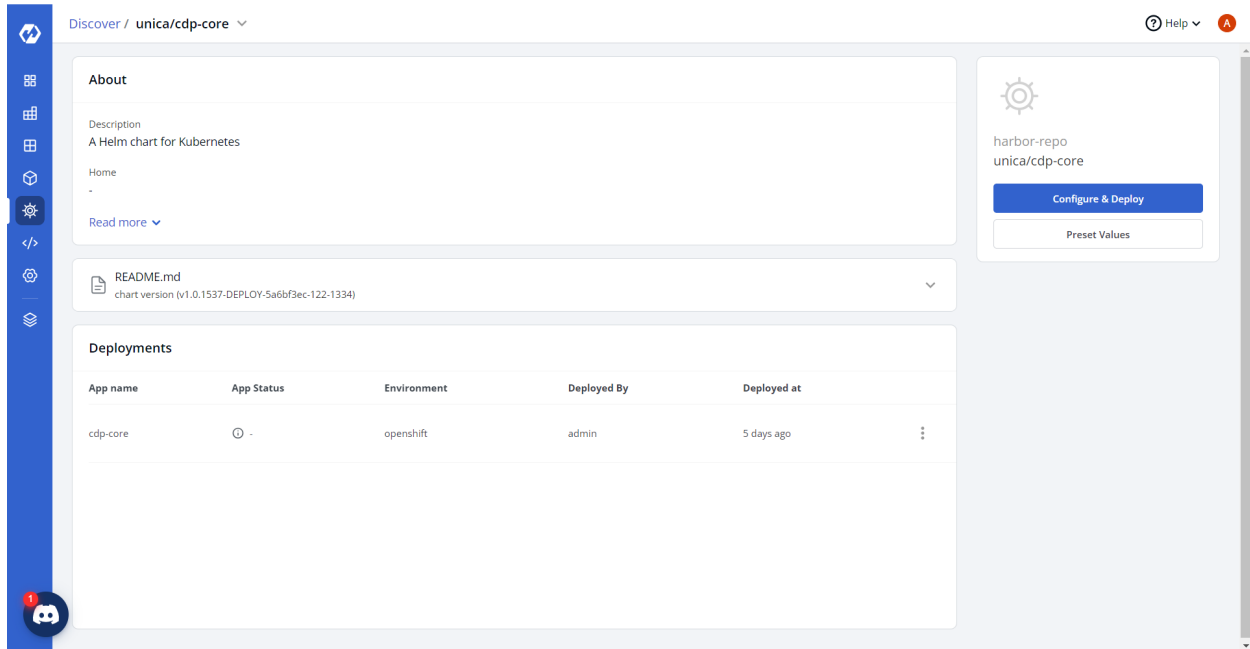
2. Update ConfigMaps data with actual values, and create required Kafka topics before deployment.

Deploy CDP Core

1. Navigate to the Devtron **Chart Store**, and select cdp-core chart to deploy.



2. Now, configure and deploy the cdp-core charts.



3. In the YAML section, update the ConfigMap using the below details and deploy the chart.

```
ALERT_TO_EMAIL_ID: ""
APN_LAMBDA_REGION: ap-south-1
APN_LAMBDA_RETRY_TOPIC: apppush_retry
APN_LAMBDA_VERSION: $LATEST
APN_LAMBDA_VERSIONING_ENABLED: "true"
APP_INFRA: k8s
APP_PUSH_ISACTIVE: "true"
APP_REGION: us-east-1
APP_TYPE: trigger-core-us
AWSRegion: ap-south-1
AerospikeBulkInsertionInitialDelay: "2"
AerospikeBulkRetryCount: "3"
BLACKLIST_EVENT_LIST: hard_bounce,soft_bounce,reject
BLACKLIST_TTL_DAYS: "30"
BUCKET_TYPE: minio
CHANNEL_PERF_METRIC: 4_ctr,5_openrate,10_ctr
CeleryParamIgnore: IGNORE_IF_PRESENT
CeleryParamSender: CORE
CeleryTopic: CeleryETALem
DB_REFRESH_INTERVAL_MS: "60000"
DBDRIVER: jdbc:mariadb
DMP_APP_PUSH_TOPIC_PREFIX: dmp_apn
DMP_EMAIL_TOPIC_PREFIX: dmp_email
DMP_FEEDBACK_TOPIC: dmp_feedback
DMP_SMS_TOPIC_PREFIX: dmp_sms
DMP_TRIGGER_API_TOPIC_PREFIX: dmp_trapi
DMP_WEB_PUSH_TOPIC_PREFIX: dmp_wpn
DMP_WHATSAPP_TOPIC_PREFIX: dmp_wsap
EMAIL_ISACTIVE: "true"
EMAIL_SENDER_KAFKA_TOPIC: dmp_email_core
ENCRYPTION_ENABLED: "false"
FROM_EMAIL_ID: ""
```

```

HTTPS_CONNECTION_COUNT: "8"
IC_AEROSPIKE_NAMESPACE: cdpstore
IC_AEROSPIKE_PROFILE_BIN: ic
IC_AEROSPIKE_SERVER_PORT: "13000"
IC_AEROSPIKE_TIMEOUT: "100000"
IC_AEROSPIKE_TTL: "31536000"
INBOUND_KAFKA_TOPIC: core_inbound
KAFKA_CONSUMER_PREFIX: core
KAFKA_EXECUTOR_SERVICE_MAX_THREADS: "64"
KAFKA_EXECUTOR_SERVICE_MIN_THREADS: "32"
KAFKA_EXECUTOR_SERVICE_QUEUE_SIZE: "1000"
KAFKA_EXECUTOR_SERVICE_THREAD_TTL: "5"
KAFKA_PRODUCER_META: "true"
KAFKA_PUSH_SENT_TOPIC: webpush_sent_metrics
KAFKA_SENDER: tctrcore
KEEP_ALIVE_SECONDS: "600"
KMS_DECRYPT_TIMEOUT: "100"
LAMBDA_MAX_PAYLOAD_SIZE: "128000"
LAMBDA_RETRY_COUNT: "3"
LAMBDA_RETRY_INTERVAL: "1000"
LOG_LEVEL_APP: debug
LOG_LEVEL_METRICS: "off"
LOG_LEVEL_ROOT: debug
LOG_LEVEL_UTILS: "off"
MAPPING_BIN: mp
MAXIMUM_IDLE_THREADS: "6"
MINIMUM_IDLE_THREADS: "3"
NBA_AEROSPIKE_TIMEOUT: "100000"
NBA_AEROSPIKE_BIN: nba
NBA_AEROSPIKE_NAMESPACE: cdpstore
NBA_AEROSPIKE_SERVER_PORT: "13000"
OPEN_TRACKER: <OPEN_TRACKER>/emailOpen?
OUTBOUND_KAFKA_TOPIC: core_outbound
PROCESSING_RETRY_COUNT: "3"
PRODUCER_EVENT_TYPE: "5656"
PRODUCER_POOL_SIZE: "16"
PUSH_ISACTIVE: "true"
PUSH_SERVER_ENDPOINT: ""
ProducerEventType: "5650"
REPORT_SCHEDULER_ISACTIVE: "true"
REPORTING_INTERVAL_IN_MIN: "55"
RTS_MS_ENDPOINT: <RTS_MS_ENDPOINT>/usermacros
S3_ENDPOINT: <S3_ENDPOINT>
S3_MAX_CONNECTIONS: "10"
S3_MAX_ERROR_RETRY: "3"
S3_MAX_RW_RETRY: "3"
S3_PROTOCOL: HTTPS
S3_REGION: ap-south-1
S3_TIMEOUT_CLIENT_EXECUTION: "50000"
S3_TIMEOUT_CONNECTION: "50000"
S3_TIMEOUT_REQUEST: "10000"
S3_TIMEOUT_SOCKET: "10000"
SHORTEN_OPEN_TRACKER:<SHORTEN_OPEN_TRACKER>/eos
SMS_ISACTIVE: "true"
SMS_SENDER_KAFKA_TOPIC: dmp_sms_core
SubscriptionEventInsertionRetry: "3"
SubscriptionEventInsertionTimeout: "3000"
SubscriptionEventTTL: "-1"

```

```

TAXIMAIL_EMAIL_SENDER_KAFKA_TOPIC: dmp_taximail_email_core
TRIGGER_API_ISACTIVE: "true"
TRIGGER_API_SENDER_KAFKA_TOPIC: dmp_trapi_core
URL_SHORTENER_MULTIPLY_FACTOR: "10000"
URL_SHORTNER_AVOID_CONFLICT: "true"
URL_SHORTNER_CONFLICT_MAX_RETRIES: "3"
URL_SHORTNER_S3_BUCKET: lem-email-url-shortener
USER_FAILURE_KAFKA_TOPIC: core_failure
VRM_DB_DRIVER: org.mariadb.jdbc.Driver
WEB_PUSH_ISACTIVE: "true"
WHATSAPP_ISACTIVE: "true"
WHATSAPP_SENDER_KAFKA_TOPIC: dmp_wsap_core
WPN_CUSTOM_LAMBDA_CAMPAIGNLIST: "0"
WPN_LAMBDA_REGION: ap-south-1
WPN_LAMBDA_RETRY_TOPIC: webpush_retry
WPN_LAMBDA_VERSION: $LATEST
WPN_LAMBDA_VERSIONING_ENABLED: "true"
defaultBusinessRegion: aps1
emailType: SMTP
s3AccessKey: <s3AccessKey>
s3AccessSecret: <s3AccessSecret>
smtpHost: <smtpHost>
smtpPort: "20225"

```

Helm Apps / cdp-core

Overview App Details **Configure** Deployment history

YAML Manifest output

Project: cdp-dev-app

Environment: openshift

Helm Chart: harbor-repo/cdp-core

Chart Version: 1.0.1537-DEPLOY-5a6bf3ec-122-1334

Chart Values: cdp-... (1.0.1537-DEPLOY-5a6bf3ec-122-1334)

ConfigMaps:

```

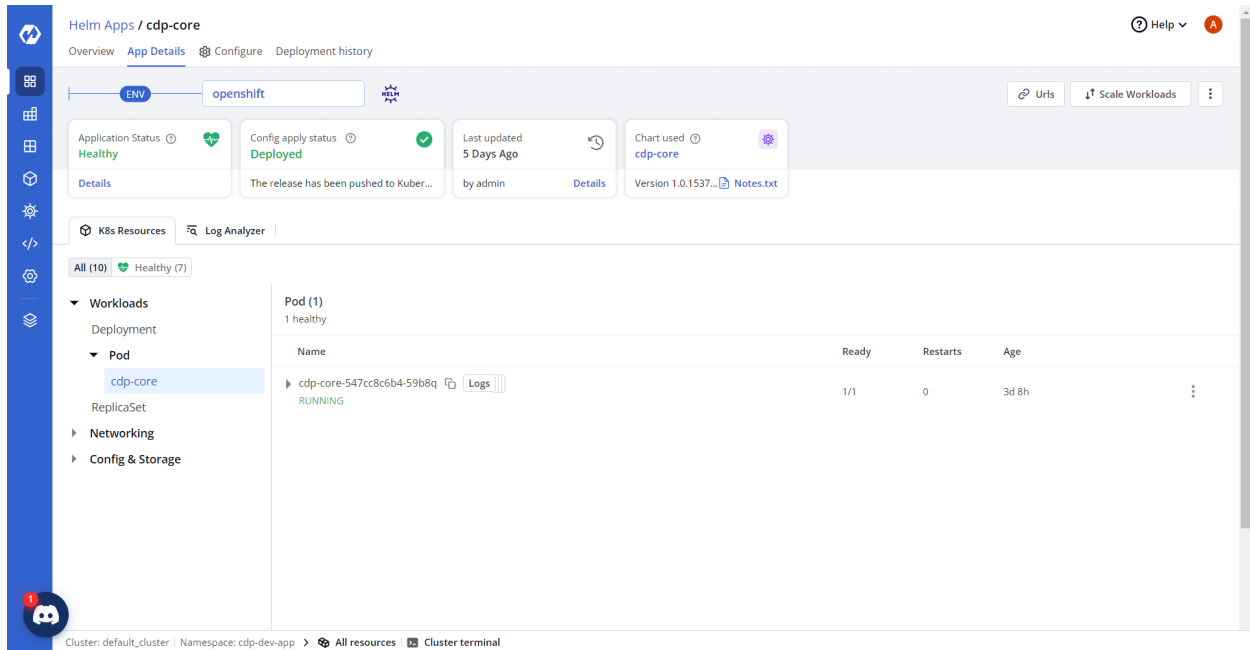
1 enabled: true
2 maps:
3   - data:
4     ALERT_TO_EMAIL_ID: ""
5     APN_LAMBDA_REGION: ap-south-1
6     APN_LAMBDA_RETRY_TOPIC: apppush_retry
7     APN_LAMBDA_VERSION: $LATEST
8     APN_LAMBDA_VERSIONING_ENABLED: "true"
9     APP_INFRA: k8s
10    APP_PUSH_ISACTIVE: "true"
11    APP_REGION: us-east-1
12    APP_TYPE: trigger-core-us
13    AWSRegion: ap-south-1
14    AerospikeBulkInsertionInitialDelay: "2"
15    AerospikeBulkRetryCount: "3"
16    BLACKLIST_EVENT_LIST: hard_bounce,soft_bounce,reject
17    BLACKLIST_TTL_DAYS: "30"
18    BUCKET_TYPE: minio
19    CHANNEL_PERF_METRIC: 4_ctr,5_opensrate,10_ctr
20    CeleryParamIgnore: IGNORE_IF_PRESENT
21    CeleryParamSender: CORE
22    CeleryTopic: CeleryETAlem
23    DB_REFRESH_INTERVAL_MS: "60000"
24    DBDRIVER: jdbc:mariadb
25    DMP_APP_PUSH_TOPIC_PREFIX: dmp_apn
26    DMP_EMAIL_TOPIC_PREFIX: dmp_email
27    DMP_FEEDBACK_TOPIC: dmp_feedback
28    DMP_SMS_TOPIC_PREFIX: dmp_sms
29    DMP_TRIGGER_API_TOPIC_PREFIX: dmp_trapi
30    DMP_WEB_PUSH_TOPIC_PREFIX: dmp_wpn
31    DMP_WHATSAPP_TOPIC_PREFIX: dmp_wsap
32

```

Delete Application

Update And Deploy

4. On successful deployment, validate the deployment as shown below.

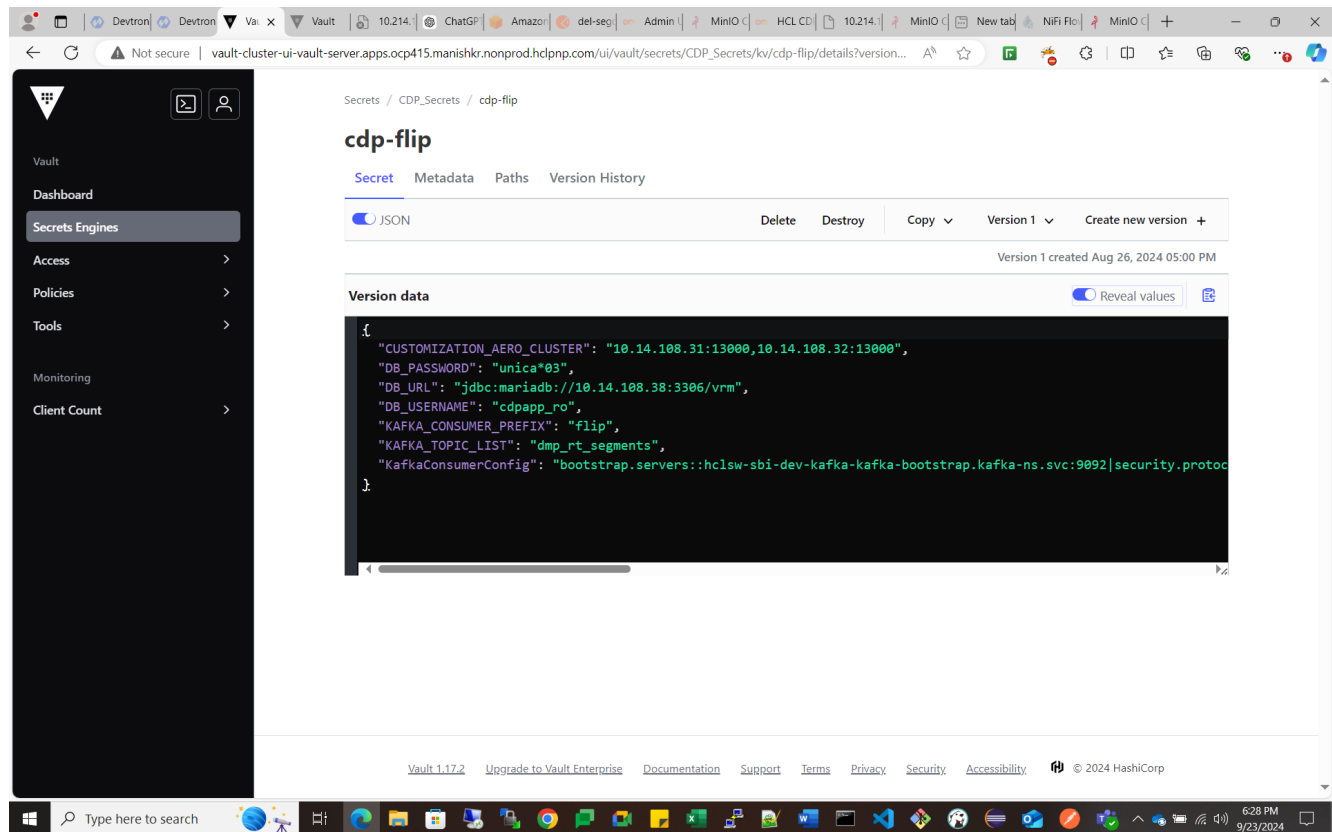


Deploying CDP Flip

This section provides detailed instructions on how to deploy CDP Flip using the Devtron in the OpenShift.

Prerequisites

Make sure to create cdp-flip secret in HashiCorp vault before deploying CDP Trigger.



To create CDP Flip secret in the HashiCorp vault, follow the steps below:

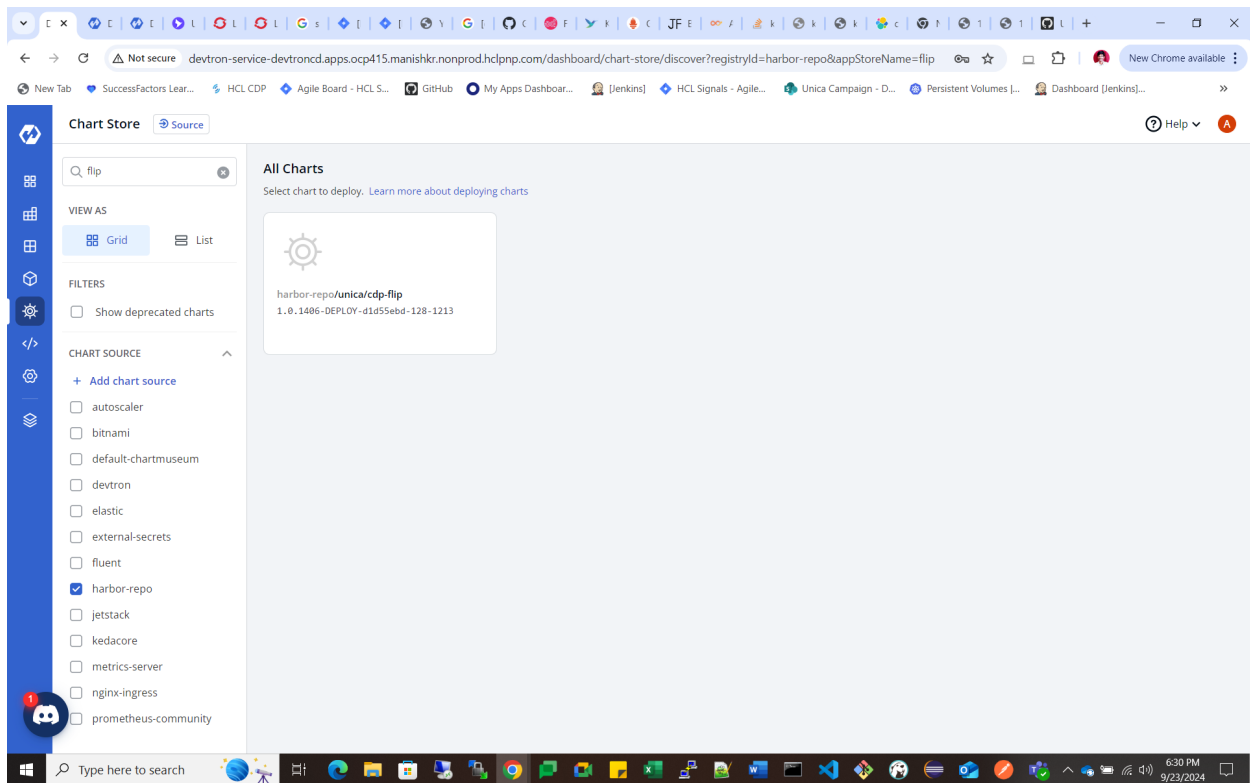
1. Create a cdp-flip secret sample key and value, and update the actual values as shown below.

```
{
  "CUSTOMIZATION_AERO_CLUSTER": "<CUSTOMIZATION_AERO_CLUSTER>",
  "DB_PASSWORD": "<DB_PASSWORD>",
  "DB_URL": "<DB_URL>",
  "DB_USERNAME": "<DB_USERNAME>",
  "KAFKA_CONSUMER_PREFIX": "flip",
  "KAFKA_TOPIC_LIST": "dmp_rt_segments",
  "KafkaConsumerConfig": "<KafkaConsumerConfig>"
}
```

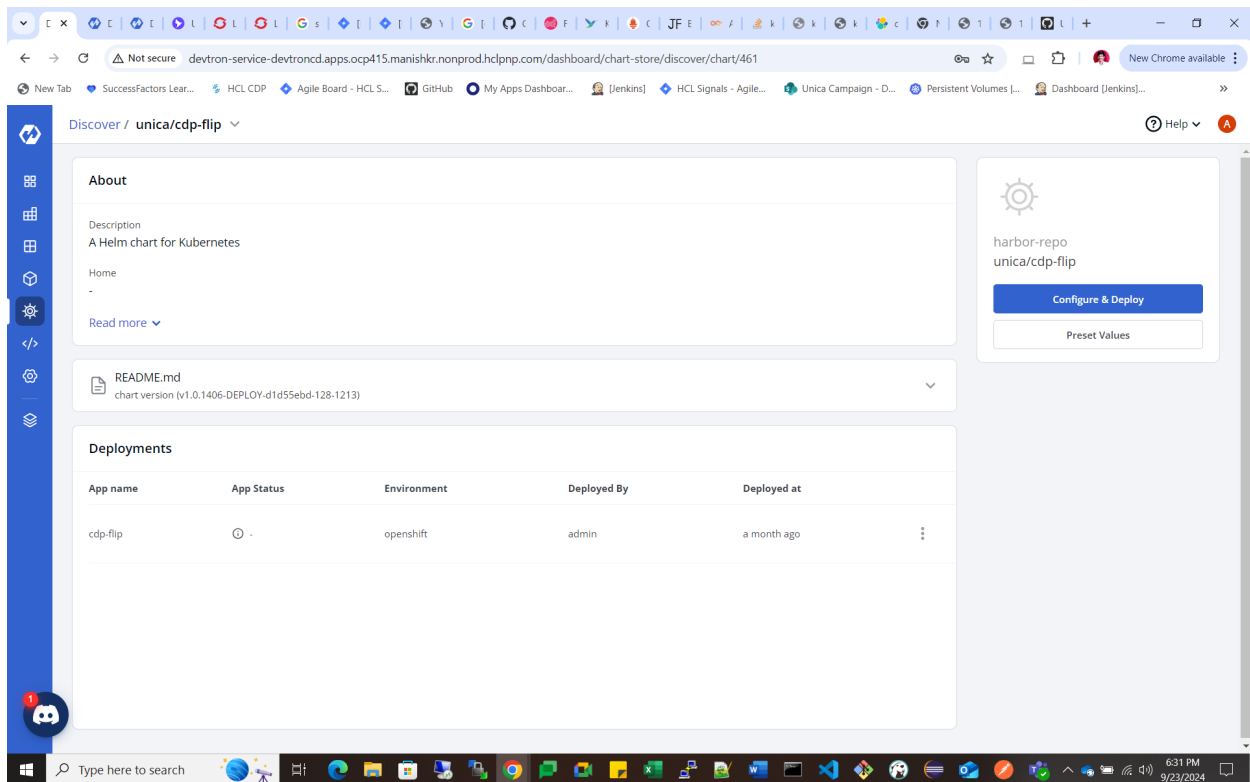
2. Create separate service account cdp-dev-app-sa with appropriate roles and permissions, and update ConfigMaps data with actual values.
3. Create required Kafka topics before deployment.

Deploy CDP Flip

1. Navigate to the Devtron **Chart Store**, and select the **cdp-flip** chart to deploy.



2. Now, configure and deploy the cdp-trigger charts.



3. In the YAML section, update the ConfigMap using the below details and deploy the chart.

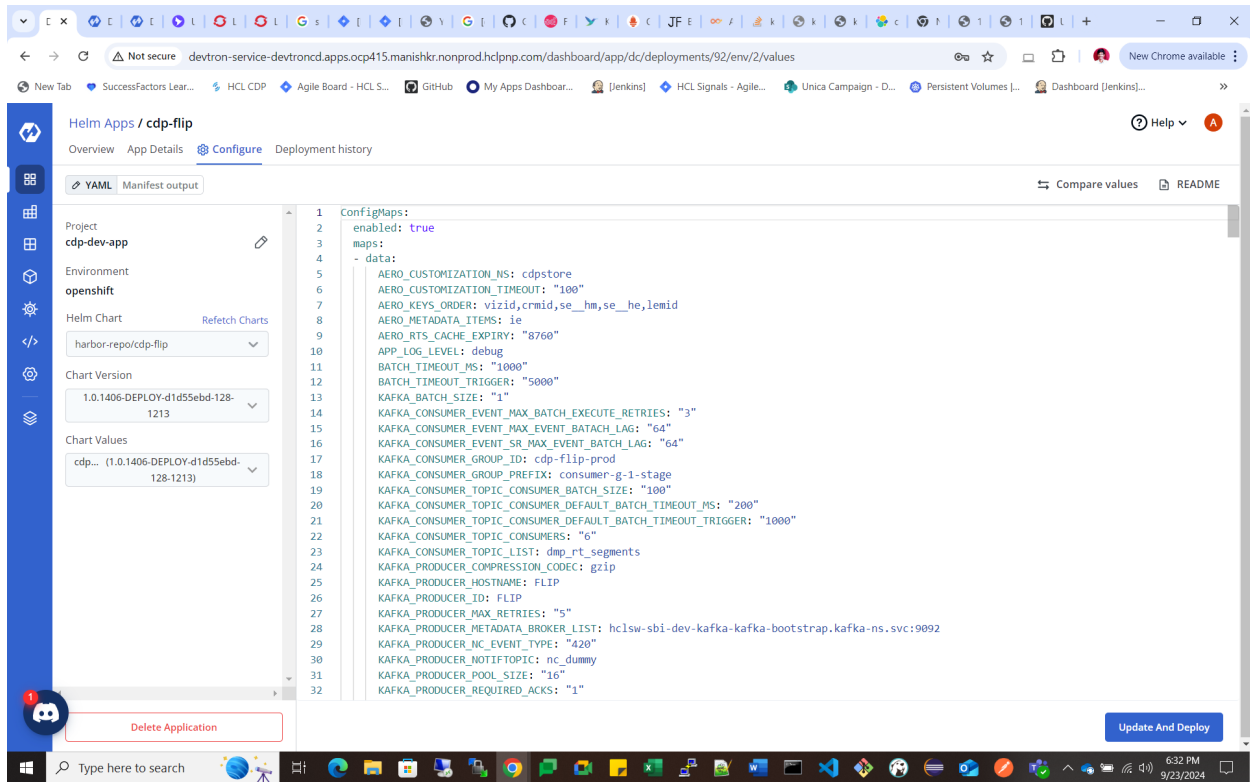
```
APP_INFRA: k8s
APP_REGION: ap-south-1
APP_TYPE: trigger--mu
AWSRegion: us-east-1
AerospikeAppPushBin: ap
AerospikeBookKeepingBin: bk
AerospikeCacheExpiry: "48"
AerospikeEmailBin: email
AerospikeIcAndBookKeepingSetPrefix: impression_
AerospikeIcAndBookKeepingSetTTL: "8760"
AerospikeImpressionCapBin: ic
AerospikeNamespace: cdpstore
AerospikePushFallbackBin: appfb
AerospikePushFallbackSet: fallback
AerospikePushFallbackSetTTL: "48"
AerospikeRamanujanBin: tc
AerospikeRamanujanSet: ramanujan
AerospikeRamanujanSetTTL: "48"
AerospikeSegmentSetTTL: "48"
AerospikeSmsBin: sms
AerospikeTimeout: "100000"
AerospikeTriggerApiBin: ""
AerospikeWebPushBin: wp
AerospikeWhatsAppBin: wsap
AppPushDelayedTriggerTopic: dmp_apn_tr
AppPushImmediateTriggerTopic: dmp_apn_tr
CeleryParamIgnore: IGNORE_IF_PRESENT
CeleryParamOverride: OVERRIDE
```

```

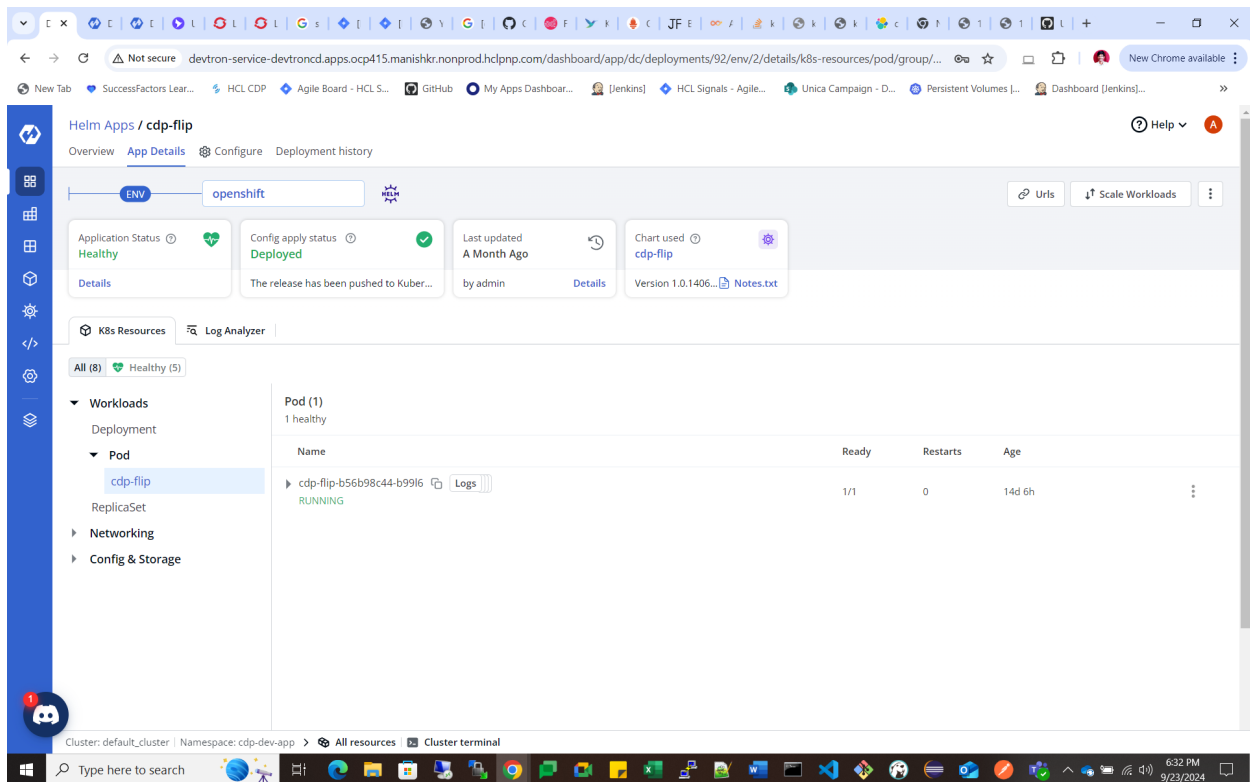
CeleryParamTrigger: TRIGGER
CeleryTimeThresholdMinutes: "30"
CeleryTriggerTopic: CeleryETALem
EXECUTOR_SERVICE_MAX_THREADS: "4"
EXECUTOR_SERVICE_MIN_THREADS: "2"
EXECUTOR_SERVICE_QUEUE_SIZE: "100"
EXECUTOR_SERVICE_THREAD_TTL: "5"
EmailDelayedTriggerTopic: dmp_email_tr
EmailImmediateTriggerTopic: dmp_email_tr
InboundKafkaTopic: trigger_inbound
KAFKA_CONSUMER_GROUP_ID: -trigger
KAFKA_POLL_INTERVAL: "200"
KAFKA_PRODUCER_EVENT_TYPE: "5656"
KAFKA_PRODUCER_META: "true"
KAFKA_PRODUCER_POOL_SIZE: "10"
KAFKA_TOPIC_CONSUMER_BATCH_SIZE: 10,10,10
KAFKA_TOPIC_CONSUMER_DEFAULT_BATCH_TIMEOUT_MS: 1000,1000,1000
KAFKA_TOPIC_CONSUMER_DEFAULT_BATCH_TIMEOUT_TRIGGER: 5000,5000,5000
KAFKA_TOPIC_CONSUMERS: 6,6,6
KAFKA_TOPIC_LIST: dmp_rt_segments,ETANotify,dmp_journey_trigger
KafkaCeleryNotify: ETANotify
KafkaJourneyTopic: dmp_journey_trigger
KafkaRTSegment: dmp_rt_segments
KafkaRTSegmentOffline: dmp_rt_segments_offline
LOG_LEVEL_APP: debug
LOG_LEVEL_METRICS: info
LOG_LEVEL_ROOT: info
LOG_LEVEL_UTILS: info
MAX_BATCH_EXECUTE_RETRIES: "3"
MAX_EVENT_BATCH_LAG: "8"
MysqlMapRefreshIntervalMinute: "1"
OPEN_TSDB_HOST: <OPEN_TSDB_HOST>
OPEN_TSDB_HOST_TO_BE_ADDED: "true"
OPEN_TSDB_PORT: "80"
ReportingIntervalInMin: "55"
RtsSegOfflineRefreshDsName: rtsegofflinerefresh
RtsSegOnlineRefreshDsName: rtsegonlinerefresh
RtsSegRefreshDsName: rtsegrefresh
SmsDelayedTriggerTopic: dmp_sms_tr
SmsImmediateTriggerTopic: dmp_sms_tr
TriggerApiDelayedTriggerTopic: dmp_trapi_tr
TriggerApiImmediateTriggerTopic: dmp_trapi_tr
UserFailureKafkaTopic: trigger_failure
WebPushDelayedTriggerTopic: dmp_wpn_tr
WebPushImmediateTriggerTopic: dmp_wpn_tr
WhatsAppDelayedTriggerTopic: dmp_wsap_tr
WhatsAppImmediateTriggerTopic: dmp_wsap_tr
alertToEmailId: ""
emailType: SMTP
fromEmailId: ""
hostname: trigger-prod-mu
smtpHost: <OPEN_TSDB_HOST>
smtpPort: "<smtpPort>"

```

Deployment Guide for HCL CDP v12.1.8 Marketing Automation Component on OpenShift



4. On successful deployment, validate the deployment as shown below.

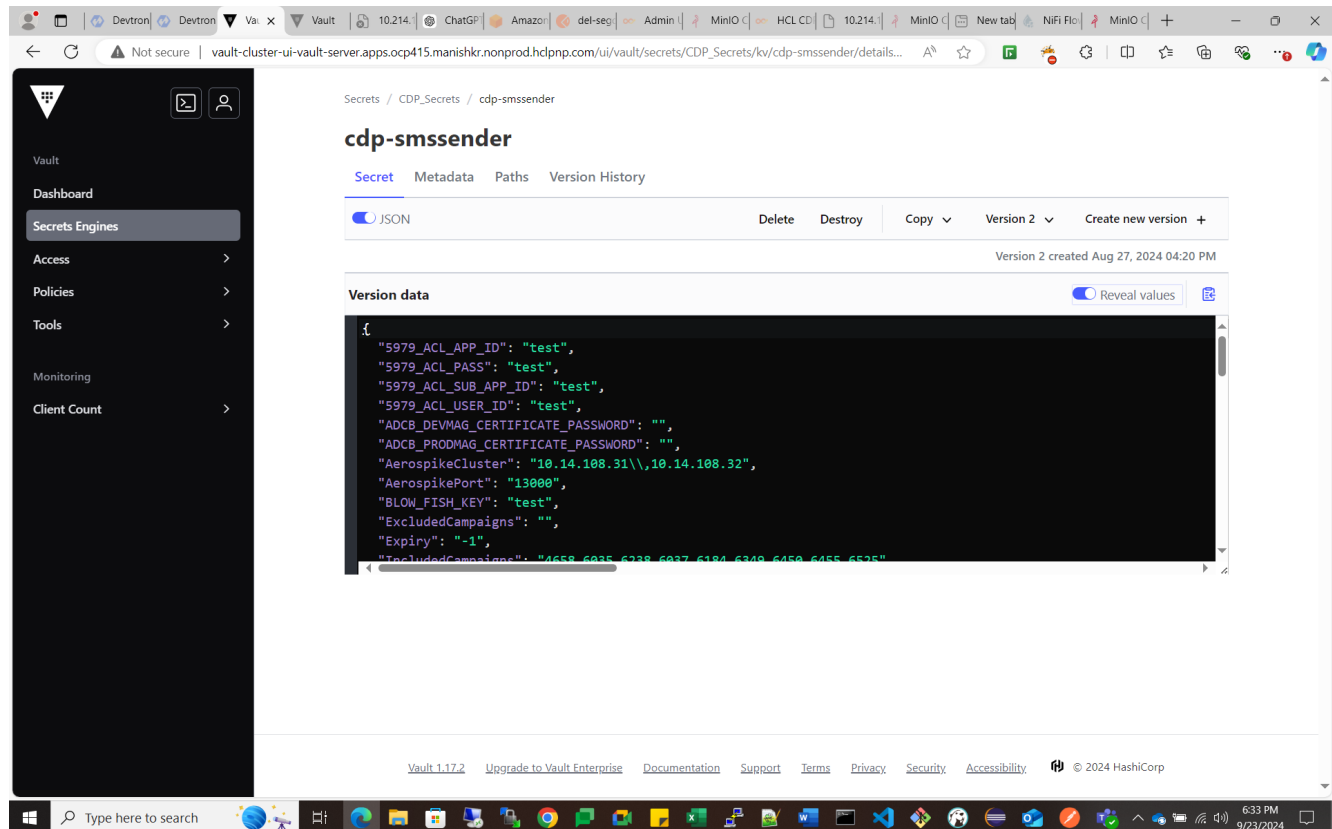


Deploying HCL CDP SMSEmailSender

This section provides detailed instructions on how to deploy HCL CDP SMS Email Sender using the Devtron in the OpenShift.

Prerequisites

Make sure to create cdp-smsemailsender secret in HashiCorp vault before deploying HCL CDP Trigger.



To create CDP SMS Email Sender secret in the HashiCorp vault, follow the steps below:

1. Create a cdp-smsemailsender secret sample key and value, and update the actual values as shown below.

```
{
  "5979_ACL_APP_ID": "test",
  "5979_ACL_PASS": "test",
  "5979_ACL_SUB_APP_ID": "test",
  "5979_ACL_USER_ID": "test",
  "ADCB_DEVMAG_CERTIFICATE_PASSWORD": "",
  "ADCB_PRODMAG_CERTIFICATE_PASSWORD": "",
  "AerospikeCluster": "<AerospikeCluster>",
  "AerospikePort": "13000",
  "BLOW_FISH_KEY": "test",
  "ExcludedCampaigns": "",
  "Expiry": "-1",
  "IncludedCampaigns": "4658,6035,6238,6037,6184,6349,6450,6455,6525",
  "KAFKA_TOPIC_CONSUMERS": "6,6,6,6",
  "KAFKA_TOPIC_CONSUMER_BATCHSIZE": "10,10,10,10",
}
```

```

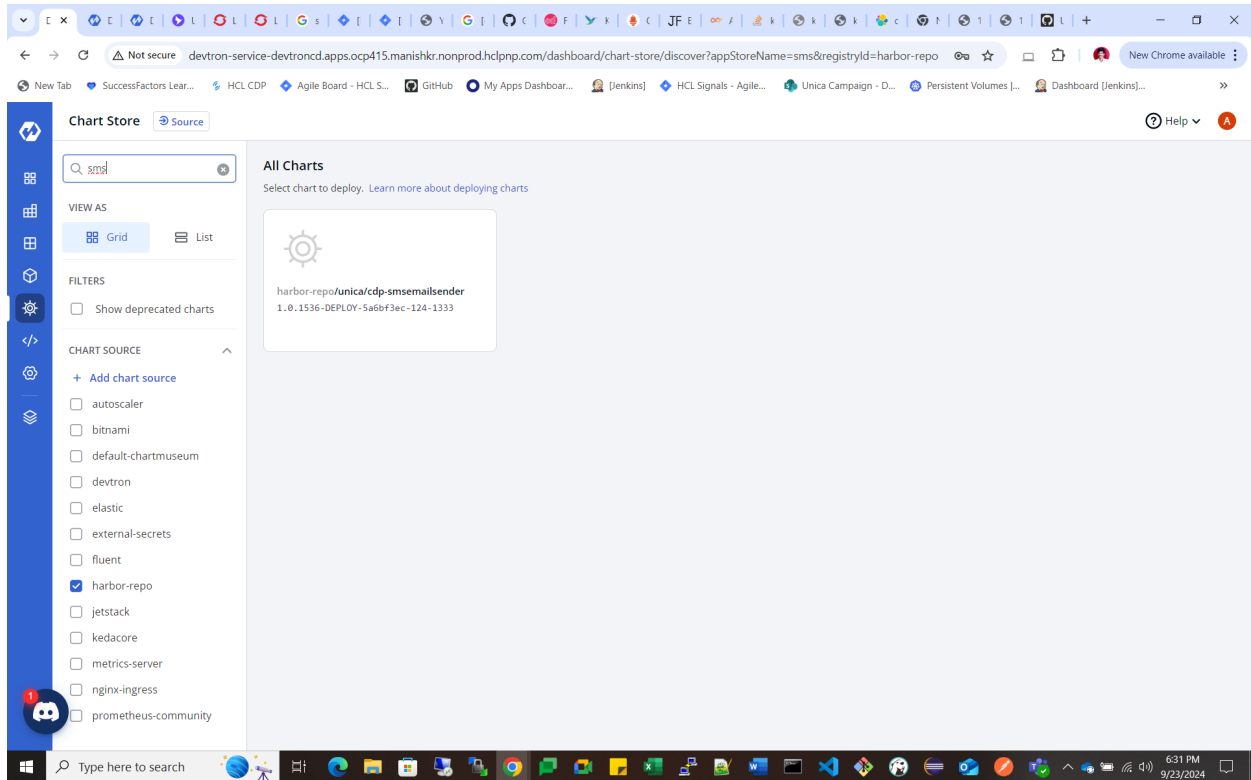
"KAFKA_TOPIC_CONSUMER_DEFAULT_BATCHTIMEOUT_TRIGGER": "5000,5000,5000,5000",
"KAFKA_TOPIC_CONSUMER_DEFAULT_BATCH_TIMEOUT_MS": "1000,1000,1000,1000",
"KAFKA_TOPIC_LIST": "dmp_sms_core_aps1,dmp_email_core_aps1,dmp_wsap_core_aps1,dmp_trapi_core_aps1",
"KENSICIO_CLIENT_END_POINT_URL": "",
"KENSICIO_MESSAGE_ID": "",
"KENSICIO_MESSAGE_ID_KEY": "",
"KENSICIO_PASSWORD": "",
"KENSICIO_SYSTEM_ID": "",
"KENSICIO_SYSTEM_ID_KEY": "",
"KENSICIO_USERNAME": "",
"KafkaBootstrapServer": "<KafkaBootstrapServer>",
"KafkaBootstrapServerUS": "",
"KafkaConsumerConfig": "<KafkaConsumerConfig>",
"KafkaPassword": "<KafkaPassword>",
"KafkaPasswordUS": "",
"KafkaProducerConfig": "<KafkaProducerConfig>",
"KafkaProducerConfigaps1": "<KafkaProducerConfigaps1>",
"KafkaProducerHostName": "trigger-sender-mu",
"KafkaProducerRegion": "aps1",
"KafkaUsername": "<KafkaUsername>",
"KafkaUsernameUS": "",
"MGAGE_XML_API": "test",
"MUAerospokeCluster": "<MUAerospokeCluster>",
"MUAerospokePort": "13000",
"MU_NAMESPACE": "cdpstore",
"MySqlDb": "vrm?autoReconnect=true ",
"MySqlHost": "<MySqlHost>",
"MySqlPort": "3306",
"Namespace": "cdpstore",
"PIILookUpEnabled": "true",
"PII_MAX_IDLE_THREADS": "4",
"PII_MIN_IDLE_THREADS": "2",
"RedisIP": "10.14.108.45",
"RedisPort": "6379",
"SesEnabled": "true",
"SupportedRegion": "aps1",
"TSDB_HOST": "<TSDB_HOST>",
"Timeout": "3000",
"URL_SHORTENER_S3_BUCKET": "lem-url-shortener",
"URL_SHORTENER_S3_BUCKET_EMAIL": "lem-email-url-shortener",
"USAerospokeCluster": "",
"USAerospokePort": "13000",
"US_NAMESPACE": "",
"VrmDbPassword": "<VrmDbPassword>",
"VrmDbUsername": "<VrmDbUsername>",
"allowTrafficForSpecificCampaign": "false",
"s3.accessKeyId": "",
"s3.region": "ap-south-1",
"s3.secretKey": ""
}

```

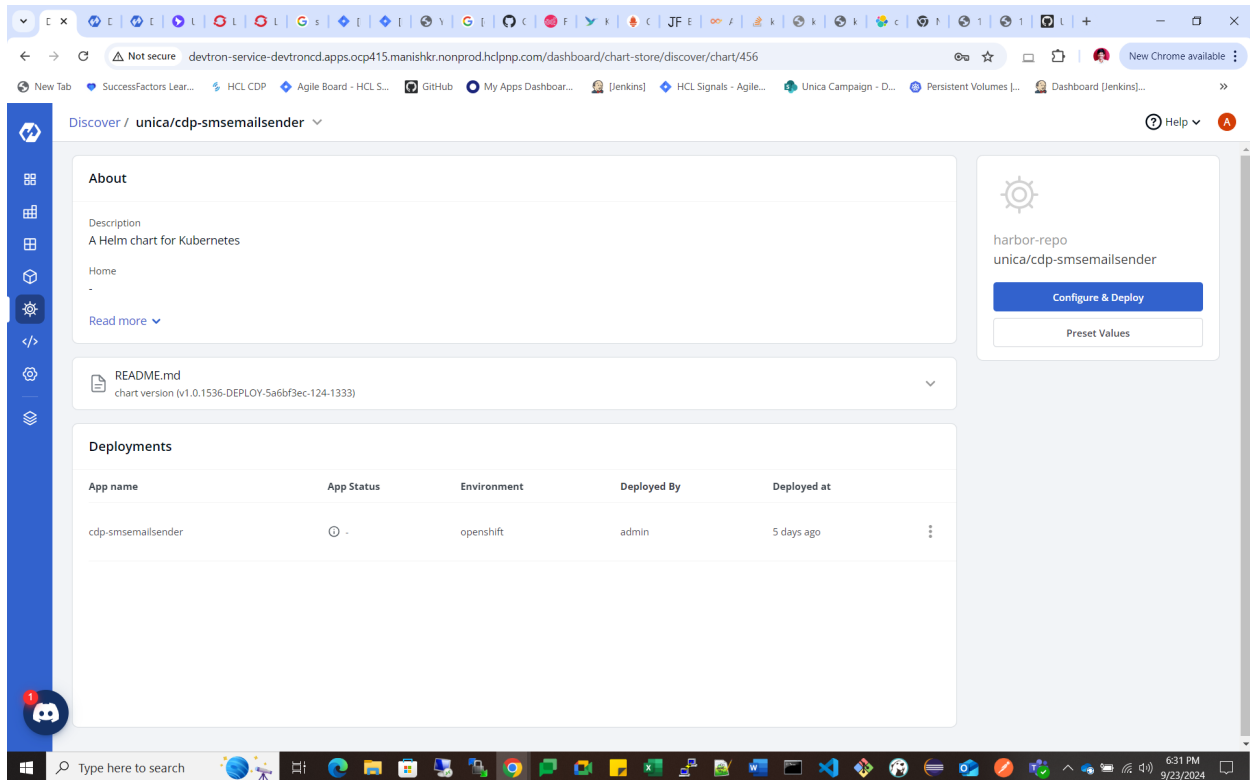
2. Update ConfigMaps data with actual values, and create required Kafka topics before deployment.

Deploy CDP SMS Email Sender

1. Navigate to the Devtron **Chart Store**, and select the `cdp-smsemailsender` chart to deploy.



2. Now, configure and deploy the `cdp-trigger` charts.



3. In the YAML section, update the ConfigMap using the below details and deploy the chart.

```
5979_ACL_APP_ID: test
5979_ACL_PASS: test
5979_ACL_SUB_APP_ID: test
5979_ACL_USER_ID: test
ADCB_DEVMAG_CERTIFICATE_PASSWORD: test
ADCB_DEVMAG_TRAPI_CERTIFICATE_FILE_PATH: test
ADCB_DEVMAG_TRAPI_CERTIFICATE_S3_FILE_PATH: test
ADCB_PRODMAG_TRAPI_CERTIFICATE_FILE_PATH: test
ADCB_PRODMAG_TRAPI_CERTIFICATE_S3_FILE_PATH: test
ADCB_STAGE_HTTPS_CONNECT_TIMEOUT_SECONDS: "130"
ADCB_STAGE_HTTPS_CONNECTION_COUNT: "10"
ADCB_STAGE_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "600"
ADCB_STAGE_HTTPS_CONNECTION_READ_TIMEOUT_SECONDS: "125"
ADCB_STAGE_HTTPS_CONNECTION_WRITE_TIMEOUT_SECONDS: "120"
ADOBE_HTTPS_CONNECT_TIMEOUT_SECONDS: "124"
ADOBE_HTTPS_CONNECTION_COUNT: "15"
ADOBE_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "120"
ADOBE_HTTPS_CONNECTION_READ_TIMEOUT_SECONDS: "900"
ADOBE_HTTPS_CONNECTION_WRITE_TIMEOUT_SECONDS: "900"
ALERT_TO_EMAIL_ID: test
APP_INFRA: k8s
APP_REGION: ap-south-1
APP_TYPE: trigger-sender-us
AWSRegion: us-east-1
BLOW_FISH_KEY: test
CELERY_PARAM_IGNORE: IGNORE_IF_PRESENT
CELERY_PARAM_SENDER: SENDER_aps1
CELERY_TOPIC: CeleryETA
```

```

CLICK_TRACKER: <CLICK_TRACKER>/emailClick?
CLICK_TRACKER_SMS: <CLICK_TRACKER>/smsClick?
COMMON_HTTPS_CONNECTION_COUNT: "100"
COMMON_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "60"
COMPONENT_NAME: trigger_sender_us
DB_REFRESH_INTERVAL_MS: "120000"
DBDRIVER: jdbc:mariadb
EMAIL_HTTPS_CONNECTION_COUNT: "10"
EMAIL_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "6000"
EMAIL_IMPRESSION_TOPIC: sms_email_events
FROM_EMAIL_ID: test
HTTPS_CONNECTION_COUNT: "10"
HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "60"
KAFKA_CONSUMER_GROUP: k8s-trigger-sender
KAFKA_EXECUTOR_SERVICE_MAX_THREADS: "64"
KAFKA_EXECUTOR_SERVICE_MIN_THREADS: "32"
KAFKA_EXECUTOR_SERVICE_QUEUES_SIZE: "1000"
KAFKA_EXECUTOR_SERVICE_THREAD_TTL: "5"
KAFKA_PRODUCER_CHANNELS: email,sms,trapi,wasp
KAFKA_PRODUCER_META: "true"
KAFKA_PRODUCER_POOL_SIZE: "10"
KENSICIO_MESSAGE_ID: test
KENSICIO_MESSAGE_ID_KEY: test
KENSICIO_PASSWORD: test
KENSICIO_SYSTEM_ID: test
KENSICIO_SYSTEM_ID_KEY: test
KENSICIO_USERNAME: test
KMS_HASHICORPVault_ENDPOINT: <KMS_HASHICORPVault_ENDPOINT>
KMS_HASHICORPVault_ENDPOINT_TOKEN: <KMS_HASHICORPVault_ENDPOINT_TOKEN>
KMS_IMPLEMENTATION: HashiCorpVault
KafkaProducerConfigtawl: test
KafkaProducerConfigusel: test
LIVSPACE_IVR_BPEU_APIID: "89"
LIVSPACE_IVR_LPEU_APIID: "99"
LOG_LEVEL_APP: debug
LOG_LEVEL_KMS: debug
LOG_LEVEL_METRICS: debug
LOG_LEVEL_PII_STORAGE: debug
LOG_LEVEL_ROOT: debug
LOG_LEVEL_UTILS: debug
MANDRILL_CLIENT_END_POINT_URL: test
MAX_EVENT_BATCH_LAG: "16"
MAXIMUM_IDLE_THREADS: "6"
METRIC_DUMPER_FILE_PATH: /disk1/config/sms-email-sender/metricDumper.ini
METRIC_FILE_PATH: /disk1/config/sms-email-sender/metrics.ini
MGAGE_XML_API: test
MINIMUM_IDLE_THREADS: "3"
OPEN_TRACKER: <CLICK_TRACKER>/emailOpen?
PASS_THROUGH_SUPPORTED_VENDOR: Phonon
PRODUCER_EVENT_TYPE: "1234"
REDIS_KEY_TTL_IN_SEC: "7776000"
REDIS_MAX_CONNECTIONS: "20"
REPORTING_INTERVAL_IN_MIN: "55"
S3_ACCESSKEYID: ""
S3_ENDPOINT: s3.amazonaws.com
S3_MAX_CONNECTIONS: "10"
S3_MAX_ERROR_RETRY: "3"
S3_MAX_RW_RETRY: "3"

```

```

S3_PROTOCOL: HTTPS
S3_REGION: ap-south-1
S3_REQUEST_METADATA_FOLDER: trapi_req_metadata
S3_SECRETKEY: ""
S3_TIMEOUT_CLIENTEXECUTION: "50000"
S3_TIMEOUT_CONNECTION: "10000"
S3_TIMEOUT_REQUEST: "5000"
S3_TIMEOUT_SOCKET: "5000"
SENDER_FEEDBACK_TOPIC: dmp_sst_nba_di_api
SERVER_PORT: "8085"
SHORTEN_CLICK_TRACKER_EMAIL: <CLICK_TRACKER>/ecs
SMS_HTTPS_CONNECTION_COUNT: "10"
SMS_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "6000"
SMS_IMPRESSION_TOPIC: sms_email_events
TRAPI_CERTIFICATE_CAMPAGN_SUFFIX: test
TRAPI_CERTIFICATE_ENDPOINT: test
TRAPI_CONFIG_REFRESH_INTERVAL: "600000"
TRAPI_FEEDBACK_IMPRESSION_TOPIC: sms_email_events
TRAPI_HTTPS_CONNECT_TIMEOUT_SECONDS: "124"
TRAPI_HTTPS_CONNECTION_COUNT: "45"
TRAPI_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "4000"
TRAPI_HTTPS_CONNECTION_READ_TIMEOUT_SECONDS: "122"
TRAPI_HTTPS_CONNECTION_WRITE_TIMEOUT_SECONDS: "123"
TRAPI_IMPRESSION_TOPIC: trapi_events
TRAPI_KAFKA_EXECUTOR_SERVICE_MAX_THREADS: "64"
TRAPI_KAFKA_EXECUTOR_SERVICE_MIN_THREADS: "32"
TRAPI_KAFKA_EXECUTOR_SERVICE_QUEUES_SIZE: "1000"
TRAPI_KAFKA_EXECUTOR_SERVICE_THREAD_TTL: "5"
TRAPI_TOPIC_PREFIX: dmp_trapi_core
TSDB_PORT: "80"
URL_SHORTENER_AVOID_CONFLICT: "true"
URL_SHORTENER_CONFLICT_MAX_RETRIES: "3"
URL_SHORTENER_DOMAIN: https://s.cdp.bz
URL_SHORTENER_DOMAIN_EMAIL: https://<url>/ck/ecs
URL_SHORTENER_MULTIPLY_FACTOR: "10000"
USER_FAILURE_KAFKA_TOPIC: sender_failure
VRM_DRIVER: org.mariadb.jdbc.Driver
WASP_IMPRESSION_TOPIC: whatsapp_events
WHATSAPP_HTTPS_CONNECTION_COUNT: "120"
WHATSAPP_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "10"
defaultBusinessRegion: aps1
emailType: SMTP
ergoLeadApi: ""
inboundKafkaTopic: sender_inbound
KMS_DECRYPT_TIMEOUT: "3"
smtpHost: <smtpHost>
smtpPort: "20225"

```

Project: cdp-dev-app

Environment: openshift

Helm Chart: harbor-repo/cdp-smsemailsender

Chart Version: 1.0.1536-DEPLOY-5a6bf3ec-124-1333

Chart Values: cdp-smse... (1.0.1536-DEPLOY-5a6bf3ec-124-1333)

```

1 ConfigMaps:
2 enabled: true
3 maps:
4 - data:
5   5979_ACL_APP_ID: test
6   5979_ACL_PASS: test
7   5979_ACL_SUB_APP_ID: test
8   5979_ACL_USER_ID: test
9   ADCB_DEVVAG_CERTIFICATE_PASSWORD: test
10  ADCB_DEVVAG_TRAPI_CERTIFICATE_FILE_PATH: test
11  ADCB_DEVVAG_TRAPI_CERTIFICATE_S3_FILE_PATH: test
12  ADCB_PRODVMAG_TRAPI_CERTIFICATE_FILE_PATH: test
13  ADCB_PRODVMAG_TRAPI_CERTIFICATE_S3_FILE_PATH: test
14  ADCB_STAGE_HTTPS_CONNECT_TIMEOUT_SECONDS: "130"
15  ADCB_STAGE_HTTPS_CONNECTION_COUNT: "10"
16  ADCB_STAGE_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "600"
17  ADCB_STAGE_HTTPS_CONNECTION_READ_TIMEOUT_SECONDS: "125"
18  ADCB_STAGE_HTTPS_CONNECTION_WRITE_TIMEOUT_SECONDS: "120"
19  ADOBE_HTTPS_CONNECT_TIMEOUT_SECONDS: "124"
20  ADOBE_HTTPS_CONNECTION_COUNT: "15"
21  ADOBE_HTTPS_CONNECTION_KEEP_ALIVE_SECONDS: "120"
22  ADOBE_HTTPS_CONNECTION_READ_TIMEOUT_SECONDS: "900"
23  ADOBE_HTTPS_CONNECTION_WRITE_TIMEOUT_SECONDS: "900"
24  ALERT_TO_EMAIL_ID: test
25  APP_INFRA: k8s
26  APP_REGION: ap-south-1
27  APP_TYPE: trigger-sender-us
28  AWSRegion: us-east-1
29  BLOW_FISH_KEY: test
30  CELERY_PARAM_IGNORE: IGNORE_IF_PRESENT
31  CELERY_PARAM_SENDER: SENDER_aps1
32  CELERY_TOPIC: CeleryETA
  
```

Buttons: Delete Application, Update And Deploy

4. On successful deployment, validate the deployment as shown below.

Project: cdp-dev-app

Environment: openshift

Application Status: Healthy

Config apply status: Deployed

Last updated: 5 Days Ago

Chart used: cdp-smsemailsender

Version: 1.0.1536-DEPLOY-5a6bf3ec-124-1333

Workloads: Deployment

Pod (1) healthy

Name	Ready	Restarts	Age
cdp-smsemailsender-55c659b9d8-f9cdn	1/1	0	4d 15h

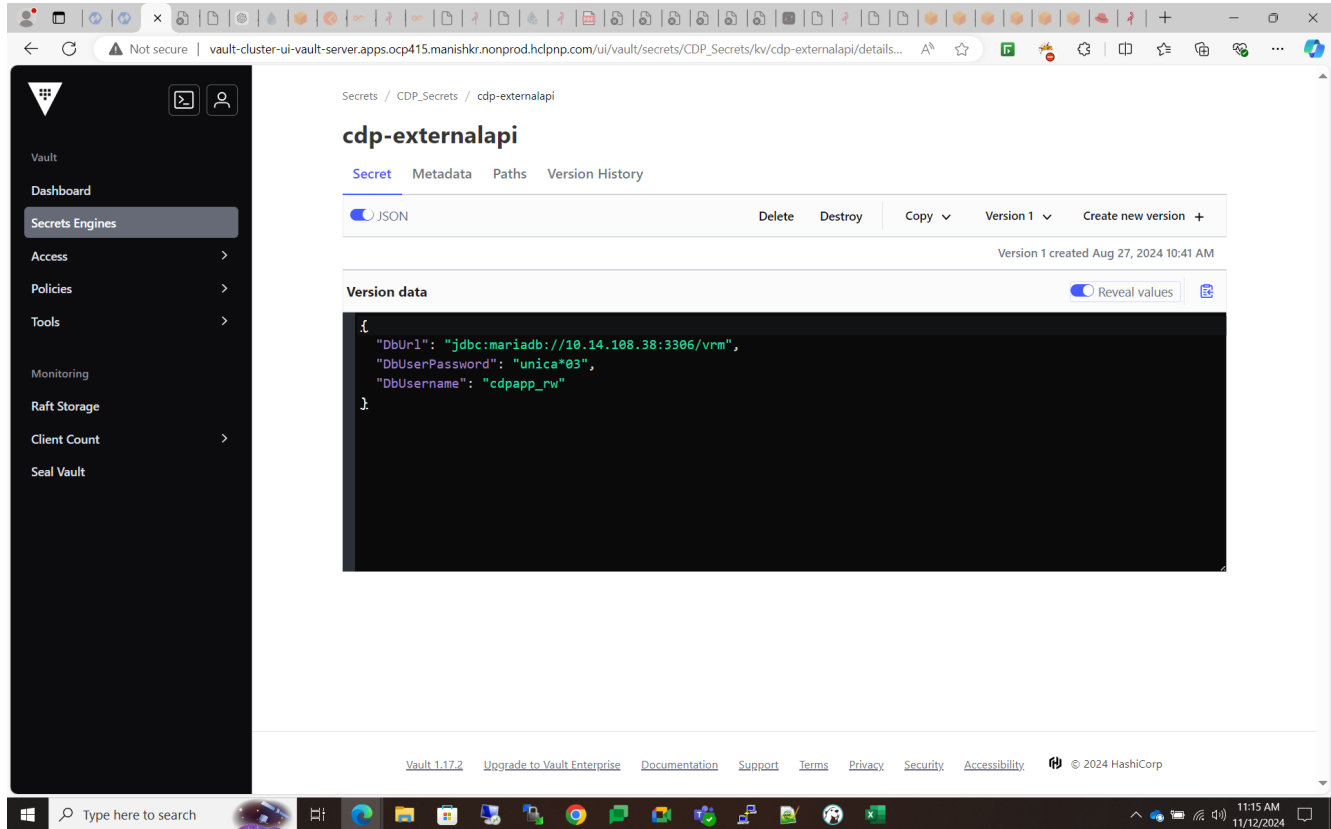
Cluster: default_cluster | Namespace: cdp-dev-app

Deploying HCL CDP ExternalAPI

This section provides detailed instructions on how to deploy HCL CDP External API using the Devtron in the OpenShift.

Prerequisites

Make sure to create cdp-externalapi secret in HashiCorp vault before deploying HCL CDP Trigger.



To create HCL CDP Trigger secret in the HashiCorp vault, follow the steps below:

1. Create a cdp-smsemailsender secret sample key and value, and update the actual values as shown below.

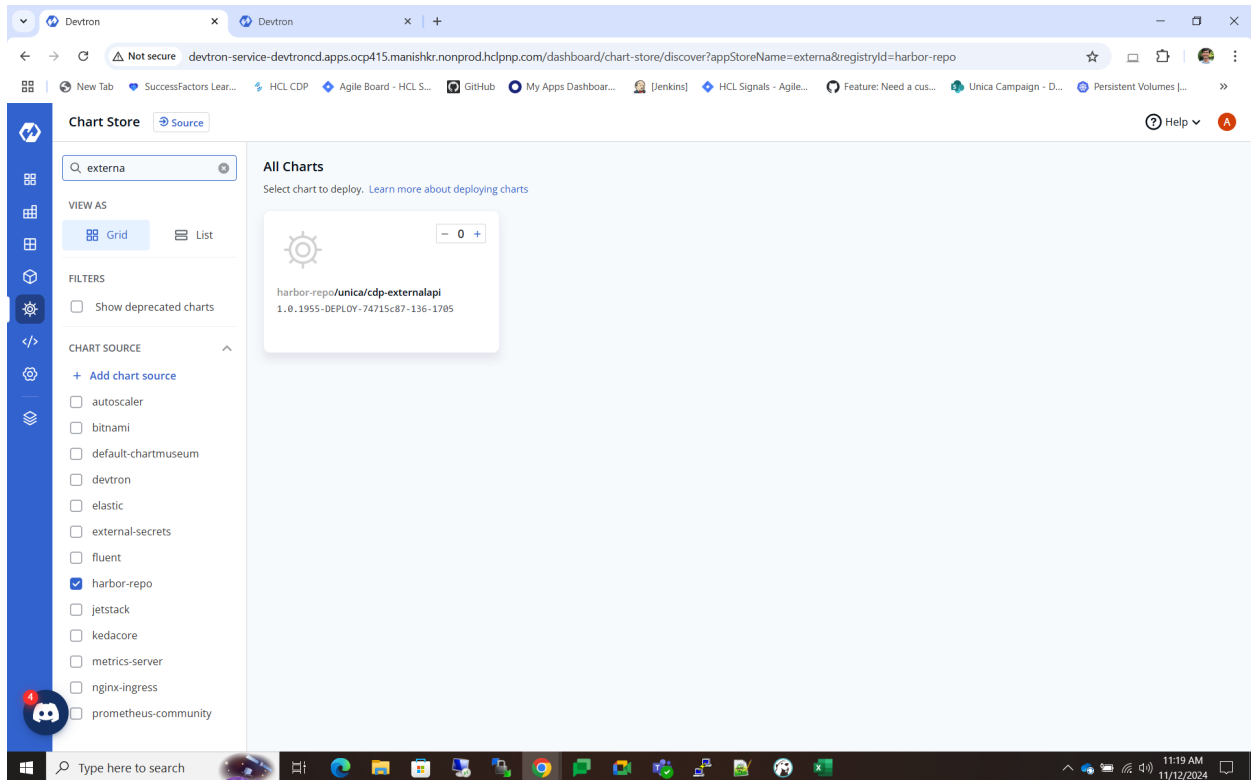
```
{
  "DbUrl": "jdbc:mariadb://ip:port/dbName",
  "DbUserPassword": "dbPwd",
  "DbUsername": "dbUser"
}
```

2. Update ConfigMaps data with actual values, and create required Kafka topics before deployment.

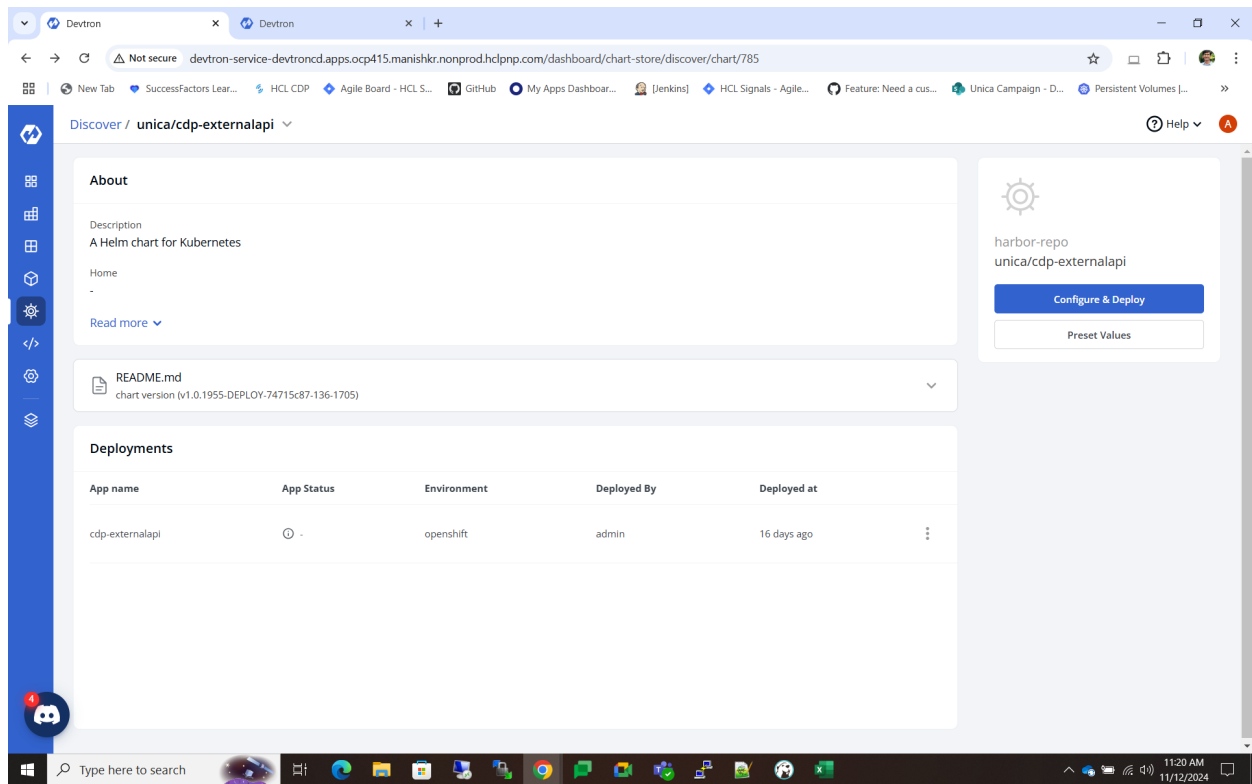
Deploy HCL CDP External API

To deploy HCL CDP External API using Devtron in Openshift platform, follow the steps below:

1. Navigate to the Devtron **Chart Store**, and select the `cdp-externalapi` chart to deploy.

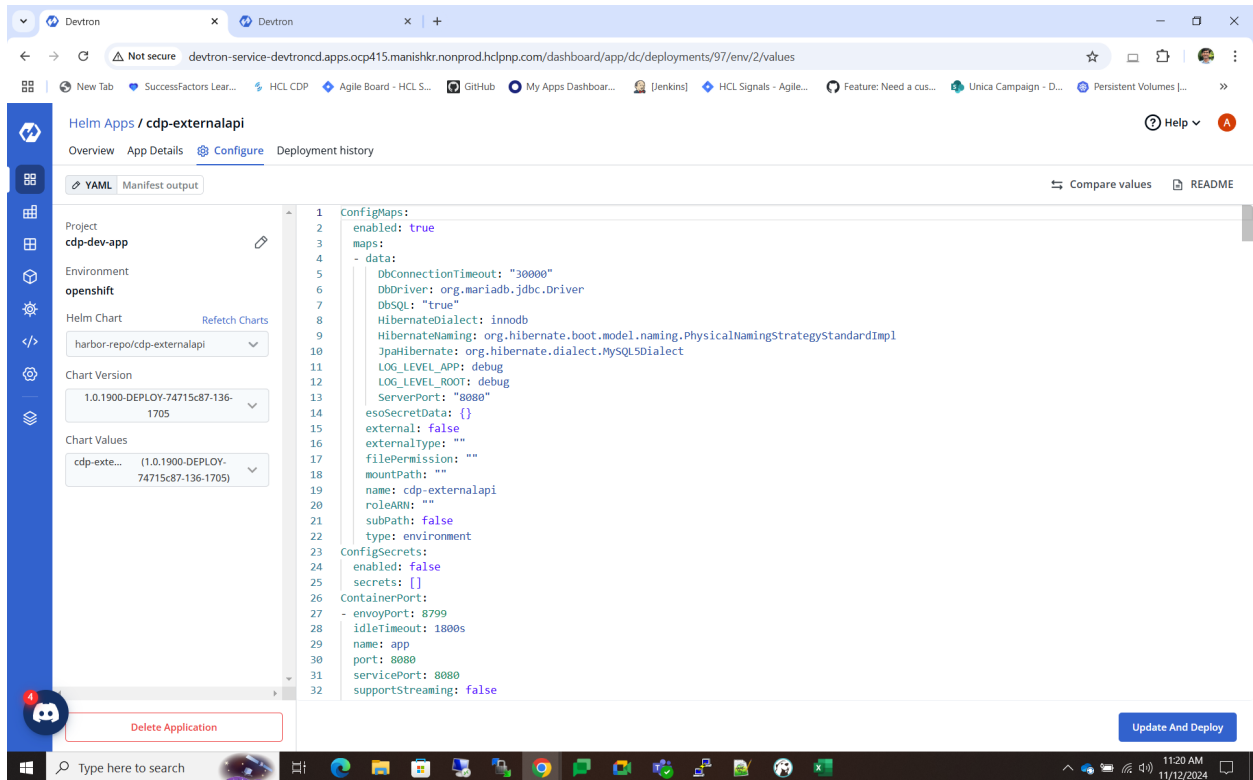


2. Now, configure and deploy the `cdp-trigger` charts.

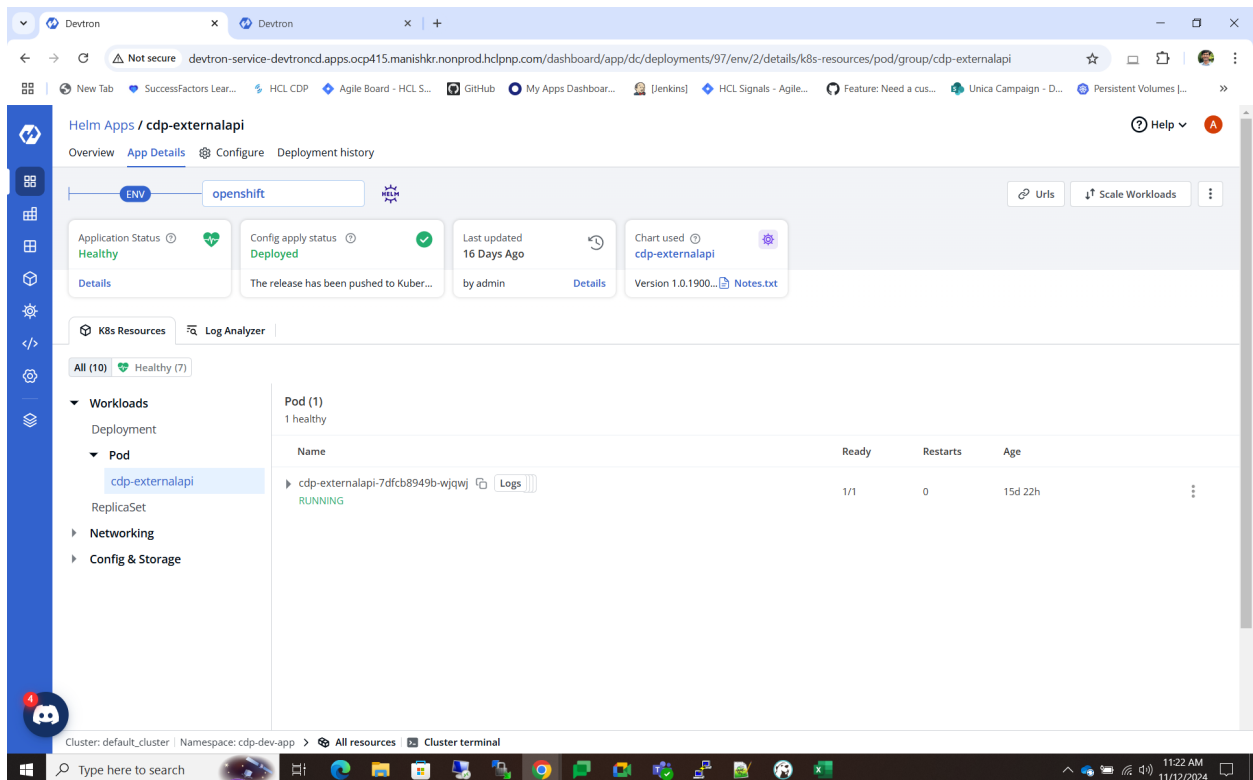


3. In the YAML section, update the ConfigMap using the below details and deploy the chart.

```
DbConnectionTimeout: "30000"
DbDriver: org.mariadb.jdbc.Driver
DbSQL: "true"
HibernateDialect: innodb
HibernateNaming: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
JpaHibernate: org.hibernate.dialect.MySQL5Dialect
LOG_LEVEL_APP: debug
LOG_LEVEL_ROOT: debug
ServerPort: "8080"
```



4. On successful deployment, validate the deployment as shown below.

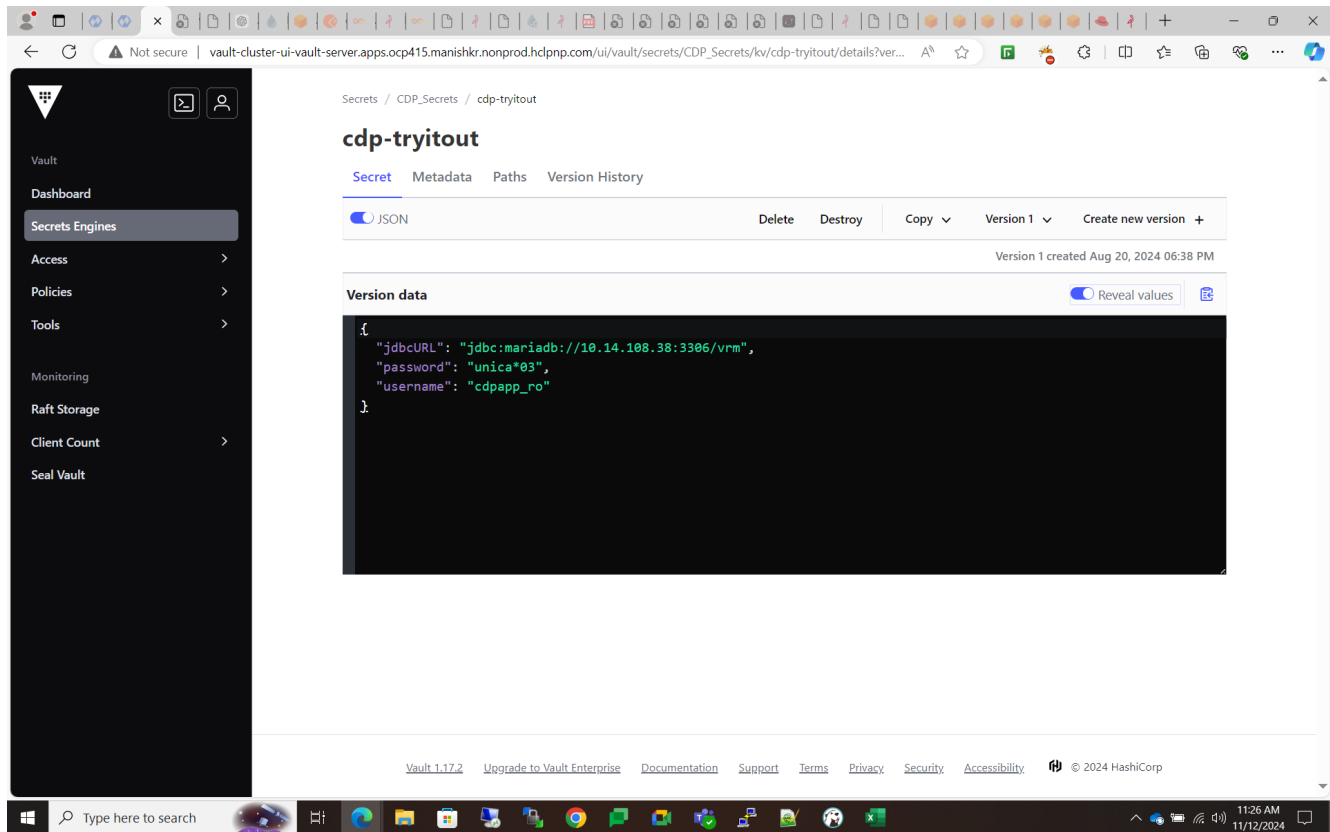


Deploying HCL CDP Tryitout

This section provides detailed instructions on how to deploy HCL CDP Tryitout using the Devtron in the OpenShift.

Prerequisites

Make sure to create cdp-tryitout secret in HashiCorp vault before deploying CDP Trigger.



To create CDP Tryitout secret in the HashiCorp vault, follow the steps below:

1. Create a cdp-tryitout secret sample key and value, and update the actual values as shown below.

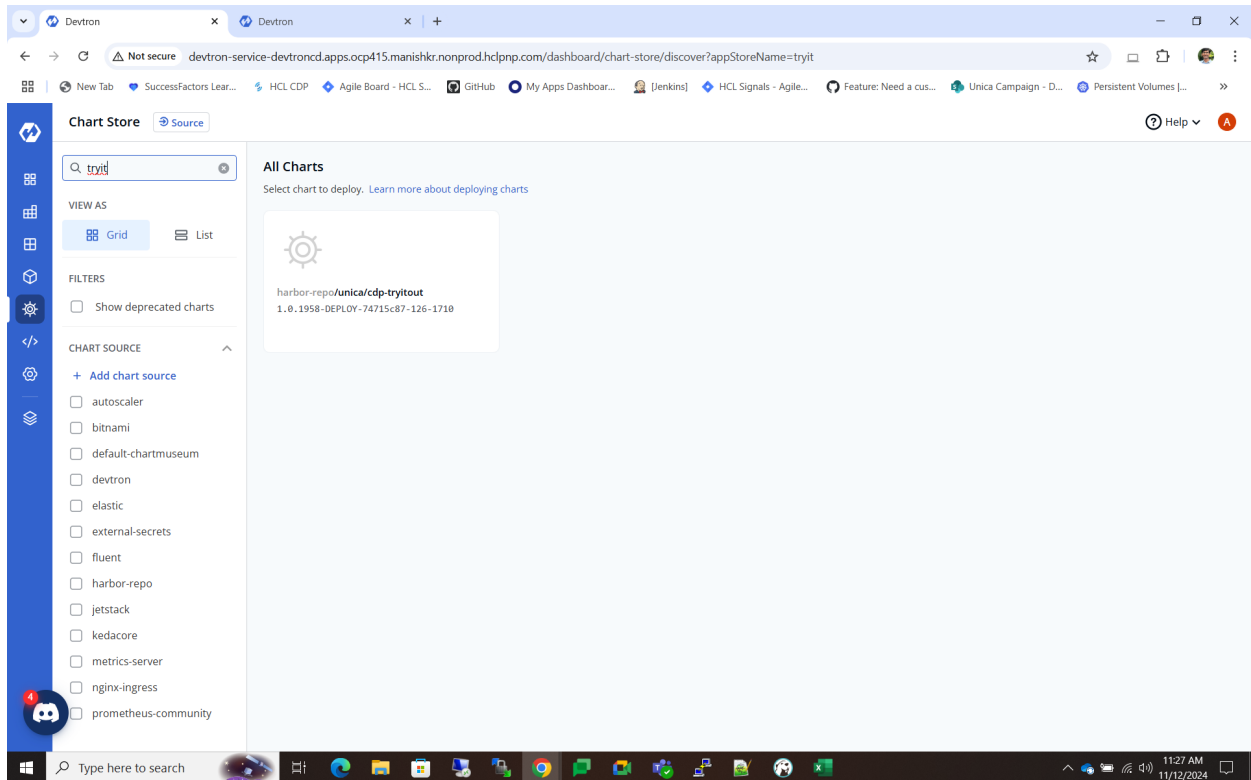
```
{
  "DbUrl": "jdbc:mariadb://ip:port/dbName",
  "DbUserPassword": "dbPwd",
  "DbUsername": "dbUser"
}
```

2. Update ConfigMaps data with actual values, and create required Kafka topics before deployment.

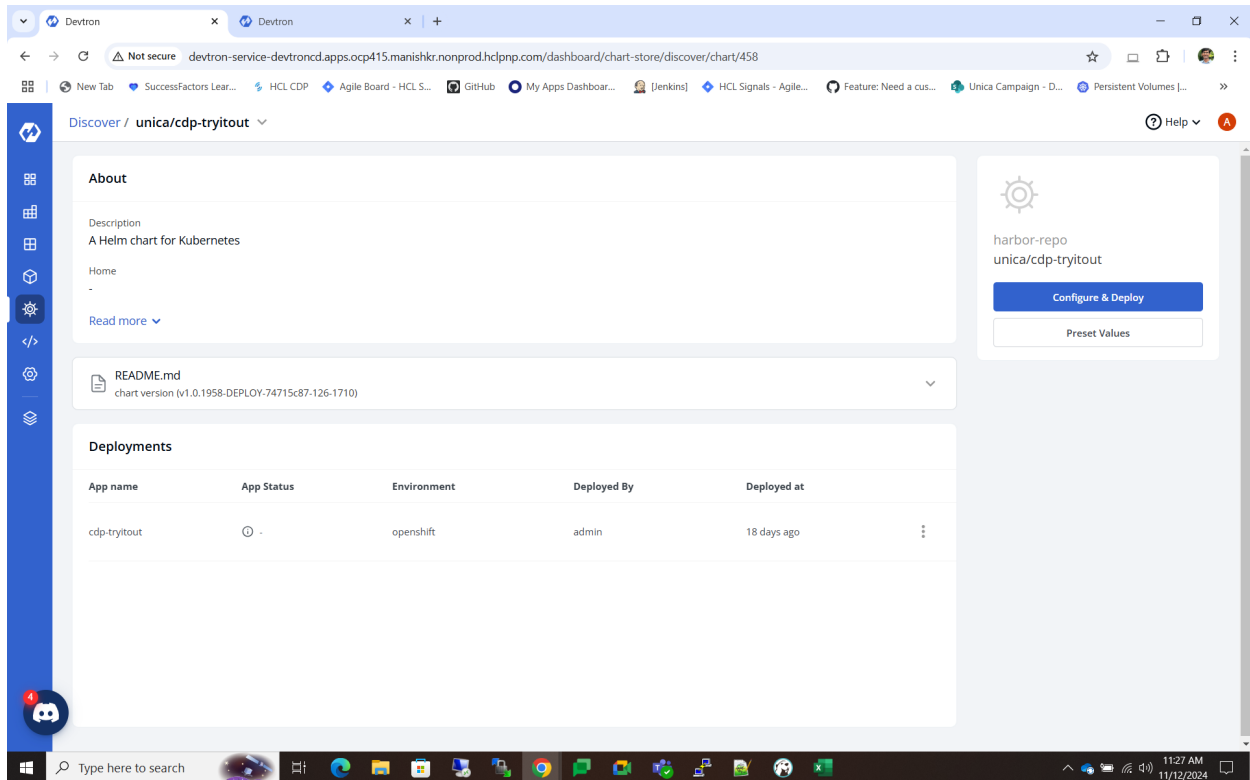
Deploy CDP TryItOut

To deploy CDP TryItOut using Devtron in Openshift platform, follow the steps below:

1. Navigate to the Devtron **Chart Store**, and select cdp-tryitout chart to deploy.



2. Now, configure and deploy the cdp-tryitout charts.



3. In the YAML section, update the ConfigMap using the below details and deploy the chart.

```
APP_DEBUG: DEBUG
APP_LOG_LEVEL: debug
KMS_HASHICORPVAULT_ENDPOINT: <KMS_HASHICORPVAULT_ENDPOINT>
KMS_HASHICORPVAULT_ENDPOINT_TOKEN: <KMS_HASHICORPVAULT_ENDPOINT_TOKEN>
KMS_IMPLEMENTATION: HashiCorpVault
addRecipientBaseUrl: http://<ip>:<port>
adobeEndpoint: http://<ip>:<port>
connectionTimeoutInMilis: "500000"
dialect: org.hibernate.dialect.MySQL5Dialect
driver: org.mariadb.jdbc.Driver
executorServiceMaxThreads: "80"
executorServiceMinThreads: "80"
executorServiceQueueSize: "100000"
executorServiceThreadTTL: "5"
hikariIdleTimeout: "10000"
hikariPoolSize: "5"
hikariTimeout: "30000"
httpTimeout: "30000"
httpsConnectTimeoutSeconds: "1240"
httpsConnectionCount: "30"
httpsConnectionKeepAliveSeconds: "120"
httpsConnectionReadTimeoutSeconds: "1220"
httpsConnectionWriteTimeoutSeconds: "1230"
lifecycle: 60s
managementEndpoints: health,info,metrics
minimumIdle: "0"
physicalStrategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
readTimeout: "500000"
```

```

responseTimeout: "500000"
serverPort: "8080"
serverShutdown: graceful
smsEmailSenderBaseUrl: <smsEmailSenderBaseUrl>
sqlStatement: "true"
storageEngine: innodb
writeTimeOut: "50000"

```

The screenshot shows the Devtron web interface for a Helm chart named 'cdp-tryitout'. The interface includes a sidebar with navigation options like Project, Environment, Helm Chart, Chart Version, and Chart Values. The main content area displays the configuration details for the chart, including the chart version (1.0.1890-DEPLOY-74715c87-126-1710) and the chart values. The values are listed in a table with columns for the value name and its value. The values include:

- APP_DEBUG: DEBUG
- APP_LOG_LEVEL: debug
- KMS_HASHICORPVault_ENDPOINT: http://vault-cluster-ui-vault-server.apps.ocp415.manishkr.nonprod.hclpnp.com
- KMS_HASHICORPVault_ENDPOINT_TOKEN: hvs.1wUeORbdlVRuA8QgBrSb2C
- KMS_IMPLEMENTATION: HashiCorpVault
- addRecipientBaseUrl: http://<ip>:<port>
- adobeEndpoint: http://<ip>:<port>
- connectionTimeoutInMills: "500000"
- dialect: org.hibernate.dialect.MySQL5Dialect
- driver: org.mariadb.jdbc.Driver
- executorServiceMaxThreads: "80"
- executorServiceMinThreads: "80"
- executorServiceQueueSize: "100000"
- executorServiceThreadTTL: "5"
- hikariIdleTimeout: "10000"
- hikariPoolSize: "5"
- hikariTimeout: "30000"
- httpTimeout: "30000"
- httpsconnectTimeoutSeconds: "1240"
- httpsconnectionCount: "30"
- httpsconnectionKeepAliveSeconds: "120"
- httpsconnectionReadTimeoutSeconds: "1220"
- httpsconnectionWriteTimeoutSeconds: "1230"
- lifecycle: 60s
- managementEndpoints: health,info,metrics
- minimumIdle: "0"
- physicalStrategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
- readTimeout: "500000"
- responseTimeout: "500000"
- serverPort: "8080"
- serverShutdown: graceful

At the bottom of the interface, there are buttons for 'Delete Application' and 'Update And Deploy'.

4. On successful deployment, validate the deployment as shown below.

