CX Contact Deployment Guide

Deploying with Kubernetes

4/4/2025

Contents

Deploying with Kubernetes

To deploy CX Contact by using Kubernetes, complete the following deployment procedures. The first three procedures are common to both deployment methods. Click the link to go to that topic:

Summary of deployment procedures
1. Ensure the Prerequisites are met
2. Review the Recommendations
3. Create the Outbound Database
4. Create the Outbound Database Access Point
5. Start Outbound Contact Server (OCS)
6. Deploy with Kubernetes

Deploy with Kubernetes

Summary of Procedures: Deploy with Kubernetes
1. Deploy CX Contact using Kubernetes and Helm charts. (CX Contact deployment with Kubernetes using shell scripts is obsolete.) • Complete the Prerequisites (for using Helm Charts) • Install CX Contact using Helm Charts • Upgrade CX Contact using Helm Charts • Configure the Helm Charts
2. Enable TLS Termination at Ingress Controller
3. Set Connectivity to the Compliance Data Provider
4. Log in to CX Contact

Deploy CX Contact using Helm Charts

Prerequisites

To begin, ensure your system contains the following prerequisite software:

- Helm 2.8+ client (without Tiller) or Helm 3
- GWS Services installed:
  - gws-core-auth
  - gws-core-environment

- gws-platform-configuration
- gws-platform-ocs
- gws-platform-voice
- gws-platform-statistics
- gws-platform-setting

- Local Docker Repository (the location of the stored CX Contact Docker images and Helm Charts).

Install CX Contact using Helm Charts

1. Select one of the following options to obtain the CX Contact Helm chart:
   - **If you have access to the local Docker Repository:** Access the Helm charts repository and run the following two commands:
     ```
     helm repo add <repo_name> <helm_charts_repo>

     helm fetch <repo_name>/cxcontact
     ```
     As a result, the `cxcontact-<version>.tgz` archive file is added to the current working directory.
   - **If you do not have access to the local Docker Repository:** Obtain the `cxcontact-<version>.tgz` archive file and save the file in your current working directory.
2. Obtain the `yaml` default values from the following location and file:
   ```
   helm inspect values cxcontact-<version>.tgz > overrides.yaml
   ```
3. Edit **overrides.yaml** and change the default parameter values to values that match your environment. See Configure the Helm Charts table for the parameters, their description and default values.
4. Using one of the following commands, install CX Contact:
   - Helm 2: `helm template cxc cxcontact-<version>.tgz -f overrides.yaml | kubectl -n <namespace> apply -f -`
   - Helm 3: `helm -n <namespace> install cxc cxcontact-<version>.tgz -f overrides.yaml`

Upgrade CX Contact using Helm Charts

1. Select one of the following options to obtain the CX Contact Helm chart:
   - Access the Helm charts repo and run the following two commands:
     ```
     helm repo update

     helm fetch <repo_name>/cxcontact
     ```
     As a result, the **cxcontact-<new_version>.tgz** archive file is added to the current working directory.
   - From the FTP Server, obtain the **.tgz** archive file.
2. Obtain the files used for the previous deployment:
   - When working with Helmp 2, obtain the **overrides.yaml** file used for the initial deployment.

- When working with Helm 3, access `helm -n <namespace> get values cxc -o yaml > overrides.yaml` to obtain the parameters used for the initial deployment.

3. Upgrade the Helm deployment:
   - When working with Helm 2, perform the following command:
     ```
     helm template cxc cxcontact-<new_version>.tgz -f overrides.yaml | kubectl -n <namespace> apply -f -
     ```
   - When working with Helm 3, perform the following command:
     ```
     helm -n <namespace> upgrade cxc cxcontact-<new_version>.tgz -f overrides.yaml
     ```

Configure the Helm Charts

Parameter	Description	Default Value
image.registry	The Docker registry base-path, where CX Contact images are stored.	pureengage-docker-staging.jfrog.io/cxcontact
image.imagePullSecrets	Kubernetes imagePullSecrets	
image.pullPolicy	Kubernetes imagePullPolicy	IfNotPresent
configserver.user_name	The Configuration Server user name. This user name should be created during provisioning and stored in Users Secret.	cloudcon
configserver.user_password	The Configuration Server user password in plain text. This password should be stored in Users Secret.	
configserver.DAP_name	Database access point application. The DAP_name should be used to connect from CX Contact.	OCSDAP_usw1
configserver.OCS_name	The Outbound Contact Server application name.	OCS_usw1
configserver.tenant_dbid	The Configuration Server Tenant DBID.	1
configserver.gws_server_app_name	The server application name that is used by GWS Services.	CloudCluster
cxcontact.replicas	The number of pod replicas that should be deployed. The recommended amount is N+1.	2
cxcontact.environment	Changes the log level of errors displayed in the UI. The environment can be either "development" or "prod".	prod
cxcontact.region	The CX Contact region. Region can be used for the deployment of multiple CX Contact installations with the same GWS Services and Redis.	g0-usw0

Parameter	Description	Default Value
cxcontact.existingPGPSecretName	The name of the existing Kubernetes Secret with PGP. existingPGPSecretName should contain the following data: • cxc_pgp_private_key • cxc_pgp_public_key • passphrase • user_id	
cxcontact.existingUsersSecretName	The name of the existing Kubernetes Secret with user credentials. existingUsersSecretName should contain the following data: • gws_client_id • gws_client_secret • configserver_user • configserver_user_pass • dial_manager_dial_api_key (optional)	
cxcontact.rbac.enabled	Configures Role Based Access Control for CX Contact.	false
cxcontact.pgp.enabled	text	Configures PGP encryption.
cxcontact.pgp.passphrase	The passphrase for the private key.	
cxcontact.pgp.user_id	The user_id for the private key.	customercare@genesys.com
cxcontact.pgp.create_k8s_secret	When set to `true`, CX Contact creates a new Secret in kubernets with pgp keys. When set to `false`, CX Contact uses the Secret from existingPGPSecretName.	false
cxcontact.pgp.private_key	The contents of the PGP private key.	
cxcontact.pgp.public_key	The contents of the PGP public key.	
cxcontact.log.level	Configures the log level for all CX Contact pods. Permitted values: • trace • debug • info	info

Parameter	Description	Default Value
	• error • fatal	
cxcontact.log.log_to_file	Configures writing logs to log files located in **/mnt/log/cxc-***.	false
cxcontact.override.amark-app.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.amark-app.env	Extra environment variables that will be appended for the container env: definition. Env can be specified as: VAR_NAME: VAR_VAL	{}
cxcontact.override.amark-app.resources	Overrides the resources for a specific micro-service.	{}
override.amark-app.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.amark-app.livenessProbe	livenessProbe	true
cxcontact.override.job-scheduler.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.job-scheduler.env	Extra environment variables that will be appended for the container env: definition. Env can be specified as: VAR_NAME: VAR_VAL	{}
cxcontact.override.job-scheduler.resources	Overrides the resources for a specific micro-service.	{}
cxcontact.override.job-scheduler.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.job-scheduler.livenessProbe	livenessProbe	true
cxcontact.override.campaign-manager.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.campaign-manager.env	Extra environment variables that will be appended for the container env: definition. Env can be specified as: VAR_NAME: VAR_VAL	{}
cxcontact.override.campaign-manager.resources	Overrides the resources for a specific micro-service.	{}
cxcontact.override.campaign-manager.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.campaign-	livenessProbe	true

Parameter	Description	Default Value
manager.livenessProbe		
cxcontact.override.list-manager.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.list-manager.env	Extra environment variables that will be appended for the container env: definition. Env can be specified as: VAR_NAME: VAR_VAL	{}
cxcontact.override.list-manager.resources	Overrides the resources for a specific micro-service.	{}
cxcontact.override.list-manager.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.list-manager.livenessProbe	livenessProbe	true
cxcontact.override.complaince.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.complaince.env	Extra environment variables that will be appended for the container env: definition. Env can be specified as: VAR_NAME: VAR_VAL	{}
cxcontact.override.complaince.resources	Overrides the resources for a specific micro-service.	{}
cxcontact.override.complaince.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.complaince.livenessProbe	livenessProbe	true
cxcontact.override.amark-ui.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.amark-ui.env	Extra environment variables that will be appended for the container env: definition. Env can be specified as: VAR_NAME: VAR_VAL	{}
cxcontact.override.amark-ui.resources	Overrides the resources for a specific micro-service.	{}
cxcontact.override.amark-ui.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.amark-ui.livenessProbe	livenessProbe	true
cxcontact.override.list builder.replicas	Overrides the number of pod replicas for a specific micro-service.	2
cxcontact.override.list builder.env	Extra environment variables that will be appended for the container env: definition. Env can	{}

Parameter	Description	Default Value
	be specified as: VAR_NAME: VAR_VAL	
cxcontact.override.list builder.resources	Overrides the resources for a specific micro-service.	{}
cxcontact.override.list builder.readinessProbe	Enables/Disables readinessProbe	true
cxcontact.override.list builder.livenessProbe	livenessProbe	true
cxcontact.override.dial-manager.enabled	Enables/Disables Dial Manager service deployment.	false
cxcontact.override.dial-manager.nexus.host	Configures the Nexus service host.	
cxcontact.override.dial-manager.nexus.port	Configures the Nexus service port.	
cxcontact.override.dial-manager.api_key	The API key used to access Nexus. The api_key should be in plain text and will be stored in Users Secret.	
cxcontact.compliance_data.cdp_url	When configured cdp_url overrides the compliance data provider URL.	false
cxcontact.compliance_data.proxy	Configures the proxy connection to CDP. Disabled if false.	false
cxcontact.compliance_data.list_builder_test_files_Data	Configures List Builder Container Data Mode for debug purposes only.	false
cxcontact.compliance_data.list_builder_test_dnc_Data	Configures List Builder Container Data Mode for debug purposes only.	false
cxcontact.initContainers	Enables the configuration of extra initContainers for CX Contact pods.	[ ]
cxcontact.deployDefaultInitContainer	Allows you to disable the default initContainer if you mount Storage with uid:guid – 500:500.	true
k8s_optional.podSecurityContext	Enables you to set the securityContext for the pod.	{}
k8s_optional.securityContext	Enables you to set the securityContext for the container.	{}
k8s_optional.nodeSelector	Enables you to configure nodeSeclector to target specific nodes.	{}
k8s_optional.tolerations	Enables you to configure tolerations.	[ ]
k8s_optional.affinity	Enables you to configure	[ ]

Parameter	Description	Default Value
	affinity.	
k8s_optional.strategy	Enables you to configure strategy.	type: RollingUpdate rollingUpdate: maxSurge: 1 maxUnavailable: 25%
redis.enabled	Enables/Disables the Reddis connection.	true
redis.cluster	Enables you to configure Redis.	true
redis.nodes	The Redis node URL.	redis://redis-cluster:6379
elasticsearch.enable	Enables/Disables the Elasticsearch Cluster connection.	true
elasticsearch.host	Elasticsearch host	http://elasticsearch
elasticsearch.port	Elasticsearch port	9200
gws.client_id	The client_id is created by the CX Contact provisioning service and is stored in the Users Secret.	cx_contact
gws.client_secret	The client_secret is created by the CX Contact provisioning service and is stored in the Users Secret.	
gws.frontend_host	Represents the GWS front end http/https URL. frontend_host is used for browser user authentication.	http://active.gke.local
gws.frontend_port	The GWS front end port.	80
loadbalander.host	GWS backend Load balacer host (optional).	
loadbalander.port	GWS backend Load balacer host (optional).	
loadbalander.core.auth.host	GWS Core Auth host	http://gws-core-auth-srv
loadbalander.core.auth.port	GWS Core Auth port	80
loadbalander.core.environment.host	GWS Core Environment host	http://gws-core-environment-srv
loadbalander.core.environment.port	GWS Core Environment port	80
loadbalander.platform.ocs.host	GWS Platform OCS host	http://gws-platform-configuration-srv
loadbalander.platform.ocs.port	GWS Platform OCS port	80
loadbalander.platform.configuration.host	GWS Platform Configuration host	http://gws-platform-configuration-srv
loadbalander.platform.configuration.port	GWS Platform Configuration port	80
loadbalander.platform.statistics.host	GWS Platform Statistics host	http://gws-platform-statistics -srv
loadbalander.platform.statistics.port	GWS Platform Statistics port	80

Parameter	Description	Default Value
loadbalander.platform.setting.host	GWS Platform Setting host	http://gws-platform-setting-srv
loadbalander.platform.setting.port	GWS Platform Setting port	80
loadbalander.platform.voice.host	GWS Platform Voice host	http://gws-platform-voice-srv
loadbalander.platform.voice.port	GWS Platform Voice port	80
ingress.enabled	Enables/Disables the deployment of the built-in ingress resource.	true
ingress.tls_enabled	HTTPS	false
ingress.cxc_frontend	The host used by ingress for all inbound traffic.	cxcontact.gke.local
ingress.annotations	The ingress resource annotations.	• nginx.ingress.kubernetes.io/ affinity: cookie • nginx.ingress.kubernetes.io/ session-cookie-samesite: "Strict" • nginx.ingress.kubernetes.io/ session-cookie-name: "cxc-session-cookie" • nginx.ingress.kubernetes.io/ proxy-body-size: "0"
ingress.tls	TLS configuration. When enabled TLS is True.	[ ]
internal_ingress.enabled	Enables/Disables the deployment of the built-in ingress resource for back-end services. When false, all endpoints are exposed on ingress with cxc_frontend.	false
internal_ingress.tls_enabled	HTTPS	false
internal_ingress.cxc_backend	The host used by ingress for all inbound traffic.	cxcontact-int.gke.local
internal_ingress.annotations	The ingress resource annotations.	• nginx.ingress.kubernetes.io/ proxy-body-size: "0" • nginx.ingress.kubernetes.io/ ssl-redirect: 'false'
internal_ingress.tls	TLS configuration. When enabled TLS is True.	[ ]
storage.pvc.enabled	Enables/Disables storage mounts.	true
storage.pvc.create	Enable pvc deployment.	true
storage.pvc.size	The size of the deployed pvc.	100Gi
storage.pvc.name	The name of the deployed pvc.	cxc-claim

Parameter	Description	Default Value
storage.pvc.storageClassName	The storageClass name that should be used when creating pvc. If storageClassName is empty it will not be used. storageClassName should be assigned accessModes: ReadWriteMany.	files-standard-zrs
storage.pv.create	Enables the creation of pv.	false
storage.pv.name	The pv name that should be created and used by pvc.	cxc-volume

storage.pv.spec	PV specification.	`capacity:`
		`  storage: 100Gi`
		`accessModes:`
		`  - ReadWriteMany`
		
		`persistentVolumeReclaimPolicy: Retain`
		`nfs:`
		`  path: /data`
		`  server: 10.128.0.42`

amark-app	docker image tag	Dependent on the CX Contact release.
job-scheduler	docker image tag	Dependent on the CX Contact release.
campaign-manager	docker image tag	Dependent on the CX Contact release.
list-manager	docker image tag	Dependent on the CX Contact release.
compliance	docker image tag	Dependent on the CX Contact release.
amark-ui	docker image tag	Dependent on the CX Contact release.
list-builder	docker image tag	Dependent on the CX Contact release.
dial-manager	docker image tag	Dependent on the CX Contact release.

Enable TLS Termination at Ingress Controller

1. Prepare the k8s secret with the SSL Certificate using the following code: `kubectl create secret cxc-tls ${CERT_NAME} --key ${KEY_FILE} --cert ${CERT_FILE}`
   Note: Skip this step if the kubernetes cluster has a cert-manager installed.
2. Update **overrides.yaml** that is used for the CX Contact installation as follows:

```
ingress:
  enabled: true
  tls_enabled: true
```

```
cxc_frontend: <fqdn>
# if kubernetes cluster has a cert-manager installed:
annotations:
        cert-manager.io/cluster-issuer: <name of cert-manager>
tls:
  - hosts:
     - <fqdn>
     secretName: cxc-tls
```

Note: The same configuration can be applied to **internal_ingress**. If the configuration is applied to **internal_ingress**, you must add the CX Contact FQDN and a certificate of the host where Configuration Server runs.

3. Prepare the k8s secret with the SSL Certificate as follows: `kubectl create secret cxc-int-tls ${CERT_NAME} --key ${KEY_FILE} --cert ${CERT_FILE}`

```
internal_ingress:
  enabled: true
  cxc_backend: <int_fqdn>
# if kubernetes cluster has a cert-manager installed:
  annotations:
        cert-manager.io/cluster-issuer: <name of cert-manager>
  tls:
    - hosts:
       - <int_fqdn>
       secretName: cxc-int-tls
```

4. Apply the following new configuration:
   `helm -n <namespace> upgrade cxc cxc -f overrides.yaml`

5. Whitelist a new **<fqdn>** on the **auth service** using one of the following methods:

   - Manually via the REST API:

```
        curl -u <GWS_BASIC_AUTH_USER>:<GWS_BASIC_AUTH_PASSWORD> -L -X PUT
'<GWS_LB_HOST>/auth/v3/ops/clients/<GWS_CLIENT_ID>' \
               -H 'Content-Type: application/json' \
               -d '{
                   "data": {
                       "redirectURIs": [
                           "https://<fqdn>/cx-contact/v3/login-callback",
                           "http://<fqdn>/cx-contact/v3/login-callback"
                       ]
                   }
               }'
```

   - Using the **cxcontact provisioning service** (cxc-app.sh), update **CXC_EXTERNAL_URL** in the **.env** file and execute: `./cxc-app.sh provision`

Set Connectivity to the Compliance Data Provider

As of CX Contact 9.0.025.xx, CDP NG is used by default. The following Helm Chart settings control the CDP NG connectivity:

```
cxcontact:
  compliance_data:
    cdp_ng:
      url: "https://api.usw2.pure.cloud/api/v2/outbound/compliancedata"
      gcloud_auth: "https://login.usw2.pure.cloud/oauth/token"
      gcloud_id:
```

```
    gcloud_secret:
  # LIST_BUILDER_DATA_EMBEDDED_BASEPATH
  embedded_basepath: "/list_builder/data/ng_init_data"
  rule_set:
    areacode: "AU,CA,GB,NZ,US"
    geo: "AU,CA,GB,NZ,US"
    postal: "CA,GB,US"
    dnc: "GB,US"
```

> **Important**
>
> The **gcloud_id** and **gcloud_secret** parameters are required and do not have default values.

The following parameters can be used to switch to legacy CDP:

```
cxcontact:
  compliance_data:
    cdp_ng:
      url: false
      gcloud_auth: false
      gcloud_id: false
      gcloud_secret: false
    # LIST_BUILDER_DATA_EMBEDDED_BASEPATH
    embedded_basepath: "/list_builder/data/init_data"
```

Log in to CX Contact

Log in to the CX Contact user interface with the URL `http://<your-docker-hostname>/ui/cxcontact/`

> **Important**
>
> You must include the backslash (/) after **cxcontact** (cxcontact/)