



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Outbound (CX Contact) Private Edition Guide

1/17/2022

Table of Contents

Overview	
About CX Contact	6
Architecture	7
High availability and disaster recovery	11
Configure and deploy	
Before you begin	12
Configure CX Contact	18
Deploy CX Contact	22
Provision CX Contact	29
Upgrade, rollback, or uninstall CX Contact	35
Observability	
Observability in Outbound (CX Contact)	37
API Aggregator metrics and alerts	41
Campaign Manager metrics and alerts	45
Compliance Manager metrics and alerts	48
Dial Manager metrics and alerts	52
Job Scheduler metrics and alerts	56
List Builder metrics and alerts	60
List Manager metrics and alerts	63

Contents

- [1 Overview](#)
- [2 Configure and deploy](#)
- [3 Observability](#)

Find links to all the topics in this guide.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

CX Contact is a service available with the Genesys Multicloud CX private edition offering.

Overview

Learn more about CX Contact, its architecture, and how to support high availability and disaster recovery.

- About CX Contact
- Architecture
- High availability and disaster recovery

Configure and deploy

Find out how to configure and deploy CX Contact.

- Before you begin
- Configure CX Contact
- Deploy CX Contact
- Provision CX Contact
- Upgrade, rollback, or uninstall CX Contact

Observability

Learn how to perform observability tasks in CX Contact.

- Monitoring
 - Alerting
 - Logging
-

About CX Contact

Learn about CX Contact and how it works in Genesys Multicloud CX private edition.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Welcome to the *Genesys Outbound (CX Contact) Private Edition Guide*. This document explains the provisioning, deployment, configuration, and start procedures for Outbound (CX Contact). The microservice that provides the outbound functionality is called Outbound (CX Contact) Service (CXCS). Because this guide covers the deployment of the service, CX Contact and CXCS terminology is used in much of the descriptive text and in any sample commands.

CX Contact is an omnichannel, outbound campaign management solution that enables you to proactively reach out to your customers in an agile and fully compliant way. It's designed to be easily managed by business users, providing the agility your organization needs when it comes to how and when to communicate with customers and prospects.

The CX Contact application provides a web UI, and contains a set of components that enable you to create, run, and manage outbound voice, SMS, and email campaigns. It is equipped with a built in self-service, email, and SMS content management system that enables easy and repeated use of pre-set campaign strategies. The list manager needs no database manipulation skills, and allows users to easily set profiles and segments to leverage different contact strategies and channels. Every uploaded contact record is enriched with global compliance data, enabling the business user to consistently manage all regulatory requirements in global, regional, or local level.

Architecture

Contents

- [1 Included services](#)

Learn about CX Contact architecture.

Related documentation:

-
-

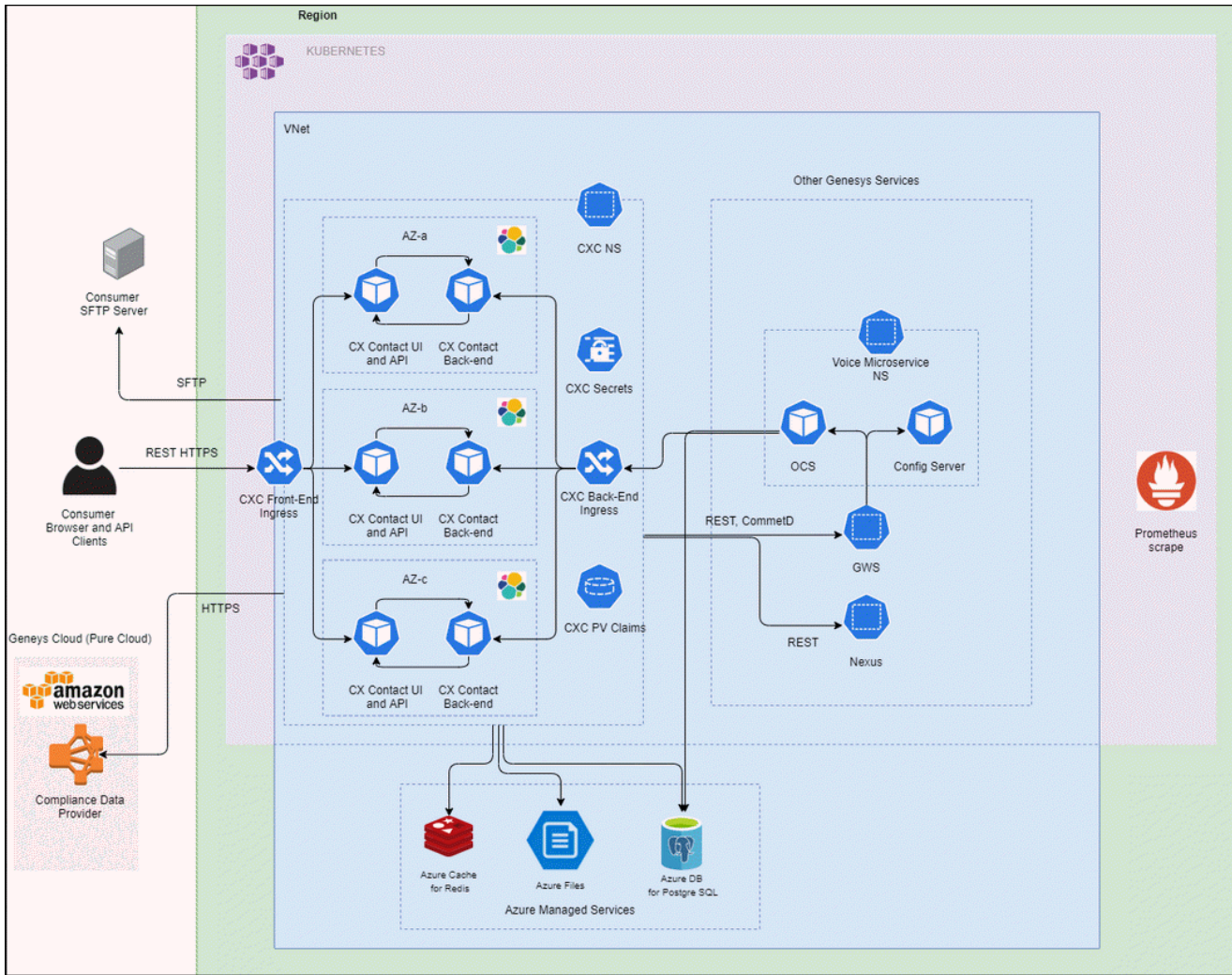
Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

CX Contact is set of microservices that run in Kubernetes containers, each scalable in N+1 horizontal mode. It has a state-of-the-art user interface (UI) and middleware components, and uses Genesys servers on the back end (Configuration Server, Outbound Contact Server (OCS), and Stat Server). Genesys Web Services (GWS) is a prerequisite.

CX Contact supports Horizontal Pod Autoscaler (HPA) for Compliance Manager and Dial Manager.

The diagram below shows the CX Contact service architecture. For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page:



Included services

The table below provides a description of the CX Contact includes services:

Included service	Description
List Builder	<ul style="list-style-type: none"> Manages operations related to lists. Creates contact lists and suppression lists in Configuration Manager. Reads Compliance data from a compliance data provider. Copies files from FTP to NFS for List Builder

Included service	Description
	consumption.
List Manager	<ul style="list-style-type: none"> • Manages operations related to lists. • Creates contact lists and suppression lists in Configuration Manager. • Reads Compliance data from a compliance data provider. • Copies files from FTP to NFS for List Builder consumption.
Compliance Manager	<ul style="list-style-type: none"> • Manages dynamic compliance rules validation. • Reads suppression entries from Redis and responds to OCS pre-validation requests.
Campaign Manager	<ul style="list-style-type: none"> • Manages operations related to campaigns. • Executes preloading of campaigns. Processing is done in Outbound Database.
Job Scheduler	Creates and invokes jobs at the right time, providing for automation of tasks.
Dial Manager	Manages SMS and email interactions with Genesys Message Aggregation.
API Aggregator	This is the entry point of APIs to CX Contact. Ensures APIs stay invariant when internal implementation changes.
User Interface (UI)	A set of static HTML5 pages served by Nginx.

High availability and disaster recovery

Find out how this service provides disaster recovery in the event the service goes down.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Service	High Availability	Disaster Recovery	Where can you host this service?
CX Contact	N = N (N+1)		

This information is under development: Flagged items aren't yet confirmed or have info coming soon; Checked items are valid.

See High Availability information for all services: High availability and disaster recovery

CX Contact does not support Disaster Recovery or any kind of cross-regional deployment.

In most scenarios, CX Contact is deployed in the primary region only. If deployed in supplementary regions, each deployment is completely independent from the other, and pods in different regions do not communicate with each other.

Before you begin

Contents

- **1 Limitations and assumptions**
 - 1.1 Prerequisites
 - 1.2 Optional tasks
- **2 Download the Helm charts**
- **3 Third-party prerequisites**
- **4 Storage requirements**
- **5 Network requirements**
 - 5.1 Single namespace
 - 5.2 External connections
 - 5.3 Ingress
 - 5.4 Ingress SSL
 - 5.5 Logging
 - 5.6 Monitoring
- **6 Browser requirements**
- **7 Genesys dependencies**
- **8 GDPR support**

Find out what to do before deploying CX Contact.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Limitations and assumptions

There are no limitations. Before you begin deploying the CX Contact service, it is assumed that the following prerequisites and optional task, if needed, are completed:

Prerequisites

- A Kubernetes or OpenShift cluster is ready for deployment of CX Contact.
- The Kubectl and Helm command line tools are on your computer.
- You have connectivity to target cluster, the proper kubectl context to work with the cluster, and your user has administrative permissions to deploy CX Contact to the defined namespace.

Optional tasks

- **SFTP Server**—Install an SFTP Server with basic authentication for optional input and output data. SFTP Server is used when automation capabilities are required.
- **CDP NG access credentials**—As of CX Contact 9.0.025, Compliance Data Provider Next Generation (CDP NG) is used as a CDP by default. Before attempting to connect to CDP NG, obtain the necessary access credentials (ID and Secret) from Genesys Customer Care.
- **Bitnami repository**—If you choose to deploy dedicated Redis and Elasticsearch for CX Contact, add the Bitnami repository to install Redis and Elasticsearch using the following command:
`helm repo add bitnami https://charts.bitnami.com/bitnami`

After you've completed the mandatory tasks, check the Third-party prerequisites.

Download the Helm charts

For information about how to download the Helm charts, see [Downloading your Genesys Multicloud CX containers](#).

See [Helm charts and containers for CX Contact](#) for the Helm chart version you must download for your release.

CX Contact is the only service that runs in the CX Contact Docker container. The Helm charts included with the CX Contact release provision CX Contact and any Kubernetes infrastructure necessary for CX Contact to run.

Third-party prerequisites

Set up Elasticsearch and Redis services as standalone services or installed in a single OpenShift cluster. See [Install third-party prerequisites in OpenShift](#). You can also install them as shared services, deployed in an "infra" namespace in OpenShift.

For information about setting up your Genesys Multicloud CX private edition platform, see [Software requirements](#).

Third-party services

Name	Version	OpenShift	GKE	Purpose	Shared service?	Notes
Elasticsearch	7.x	Elasticsearch (ECK) Operator	OpenSearch Helm chart	Used for text searching and indexing. Deployed per service that needs Elasticsearch during runtime.	Yes	CX Contact supports Elasticsearch 6.3 and later releases.
Redis	6.x	Redis Enterprise Operator	Redis Helm chart	Used for caching. Only distributions of Redis that support Redis cluster mode are supported, however, some services may not support cluster mode.	No	CX Contact supports Redis 4.0 (5.0 and later releases recommended), clustered with persistence in Production.

Storage requirements

CX Contact requires shared persistent storage and an associated storage class created by the cluster administrator. The Helm chart creates the ReadWriteMany (RWX) Persistent Volume Claim (PVC) that is used to store and share data with multiple CX Contact components.

The minimal recommended PVC size is 100GB.

Network requirements

This topic describes network requirements and recommendations for CX Contact in private edition deployments:

Single namespace

Deploy CX Contact in a single namespace to prevent ingress/egress traffic from going through additional hops, due to firewalls, load balancers, or other network layers that introduce network latencies and overhead. Do not hardcode the namespace. You can override it by using the Helm file/values (provided during the Helm install command **standard --namespace= argument**), if necessary.

External connections

For information about external connections from the Kubernetes cluster to other systems, see Architecture. External connections also include:

- Compliance Data Provider (AWS)
- SFTP Servers

Ingress

The CX Contact UI requires Session Stickiness. Use **ingress-nginx** as the ingress controller (see github.com).

Important

The CX Contact Helm chart contains default annotations for session stickiness only for **ingress-nginx**. If you are using a different ingress controller, refer to its documentation for session stickiness configuration.

Ingress SSL

If you are using Chrome 80 or later, the **SameSite** cookie must have the **Secure** flag (see Chromium Blog). Therefore, Genesys recommends that you configure a valid SSL certificate on ingress.

Logging

Log rotation is required so that logs do not consume all of the available storage on the node.

Kubernetes is currently not responsible for rotating logs. Log rotation can be handled by the **docker json-file log driver** by setting the **max-file** and **max-size** options.

For effective troubleshooting, the engineering team should provide **stdout logs** of the pods (using the command **kubectl logs**). As a result, log retention is not very aggressive (see JSON file logging driver). For example:

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m",
    "max-file": "3"
  }
}
```

For on-site debugging purposes, CX Contact logs can be collected and stored in Elasticsearch. (For example, EFK stack. See [medium.com](#)).

For more information about logging, see [\[\[PEC-OU/Current/CXCPEGuide/Logging\]\]](#).

Monitoring

CX Contact provides metrics that can be consumed by Prometheus and Grafana. It is recommended to have the **Prometheus Operator** (see [github.com](#)) installed in the cluster. CX Contact Helm chart supports the creation of **CustomResourceDefinitions** that can be consumed by the Prometheus Operator.

For more information about monitoring, see [Observability in Outbound \(CX Contact\)](#).

Browser requirements

Browsers

Name	Version	Notes
Chrome	Current release or one version previous	Chrome updates itself automatically. Versions of Chrome are only an issue if your IT department restricts automatic updates. The latest version of Chrome must be used as the CX Contact UI browser.
Microsoft Edge (Legacy)	Current release	Starting from CX Contact release 9.0.026.04, Edge Chromium browser 2020

Genesys dependencies

CX Contact components operate with Genesys core services (v8.5 or v8.1) in the back end. All voice-processing components (Voice Microservice and shared services, such as GVP), and the GWS and Genesys Authentication services (mentioned below) must be deployed and running before deploying the CX Contact service. See [Order of services deployment](#).

The following Genesys services and components are required:

- GWS
- Genesys Authentication Service
- Tenant Service
- Voice Microservice
- Multi-tenant Configuration Server

Nexus is optional.

GDPR support

CX Contact does not support GDPR.

Configure CX Contact

Contents

- [1 Override Helm chart values](#)
- [2 Configure Kubernetes](#)
- [3 Configure security](#)
 - [3.1 Security Context](#)
 - [3.2 Arbitrary UIDs in OpenShift](#)

Learn how to configure CX Contact.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Override Helm chart values

You can specify parameters for the deployment by overriding Helm chart values in the **values.yaml** file. See the Parameters table for a full list of overridable values.

For more information about Helm chart values, see *Overriding Helm chart values* in the suite-level *Private Edition Guide*.

If you want to use arbitrary UIDs in your OpenShift deployment, you must override the **securityContext** settings in the **values.yaml** file, so that no user or group IDs are specified. For details, see *Security Context*, below.

Parameter	Description
configserver.user_name, user_password	Defines the system username and password for CX Contact.
redis.nodes	Provides a valid URI to Redis.
redis.password	Provides a valid auth password for Redis.
elasticsearch.host	Provides a valid URI to Elasticsearch.
gws.client_id	The name of the GWS service client that will be created (if it doesn't exist) and the secret that will be placed in the k8s secrets repository.
gws.client_secret	The client that will be created with this secret string. If a GWS client with this name already exists, you'll need to enter the secret here.
gws.frontend_host, frontend_port	The SSO GAuth URI where CX Contact redirects during log in.
core.auth, environment	The internal URI to core services that is required for

Parameter	Description
	further provisioning. You can see, in our example GAuth is installed in namespace "gauth"
platform.ocs, configuration, .. etc.	The internal URI to the platform's GWS services. You can see, in our example GWS is installed in namespace "gws"
ingress.cxc_frontend	Creates a URI that is used by Ingress to route external incoming requests to CX Contact (Web UI and API).
internal_ingress.cxc_backend	Creates the URI that is used by Ingress to route internal incoming requests to CX Contact (API for OCS, GWS, Designer, etc)
storage.size	Defines the appropriate size for the permanent storage, depending on the daily volume of interactions, etc.
storage.storageClassName	Picks the existing Storage Class, which is described in this document earlier.

Configure Kubernetes

Preconfiguring Kubernetes ConfigMaps is not a requirement for CX Contact in OpenShift. However, you do create the default secret when you are preparing the cluster resources. See [Create the pull secret](#).

Configure security

When configuring CX Contact, you must set the connectivity to the Compliance Data Provider (CDP).

Tip

Before attempting to connect to CDP Next Generation (NG), you'll need the access ID and Secret. To obtain these credentials, contact Genesys Customer Care.

As of 9.0.025.xx, CX Contact uses CDP NG 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, but do not have default values.

You can use the following parameters 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"
```

Security Context

The security context settings define the privilege and access control settings for pods and containers.

By default, the user and group IDs are set in the **values.yaml** file as 500:500:500, meaning the **genesys** user. For example:

```
securityContext:
  runAsNonRoot: true
  runAsUser: 500
  runAsGroup: 500
  fsGroup: 500
```

Arbitrary UIDs in OpenShift

If you want to use arbitrary UIDs in your OpenShift deployment, you must override the **securityContext** settings in the **values.yaml** file, so that you do not define any specific IDs. For example:

```
securityContext:
  runAsNonRoot: true
  runAsUser: null
  runAsGroup: 0
  fsGroup: null
```

Deploy CX Contact

Contents

- 1 Prepare cluster resources
 - 1.1 Create the Storage Class
- 2 Deploy CX Contact
- 3 Validate the deployment in OpenShift
- 4 Configure monitoring and logging

Learn how to deploy CX Contact.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Include a link to the "suite-level" documentation that describes the particular cloud platform - for the initial Genesys Multicloud CX private edition release, this is OpenShift: [\[\[PrivateEdition/Current/PEGuide/GetStarted\]\]](#)

Important

Review Before you begin for the full list of prerequisites required to deploy CX Contact.

CX Contact is a shared service and is deployed in each region, as required. After deployment, it will be fully functional only in the tenant's primary region.

Prepare cluster resources

To prepare your cluster resources, create a storage class.

Tip

Creating the storage class is optional. If you have an existing storage class, you can use it and a different provisioner, but the storage class must have ReadWriteMany (RWX) capabilities.

Create the Storage Class

1. In the Outbound Contact (OC) Command Line Interface (CLI), log in to the Cluster UI.
2. Go to **Storage**, and click **Storage Claim > Create Storage Class**.
3. Click **Edit YAML**.
4. Update the values in the template using the example below that works with the setup of your environment.
 - If your cluster is on-premises (NFS-based storage):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cxc-storage
provisioner: cluster.local/nfs-vce-c00ds-vol1-nfs-subdir-external-provisioner
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

You might want to use Kubernetes to provision the NFS storage in your OpenShift cluster. In GitHub, see [Kubernetes NFS Subdir External Provisioner](#). Ask your cluster administrator for assistance with the configuration.

- If your cluster is in the cloud (Azure files-based storage):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cxc-storage
provisioner: kubernetes.io/azure-file
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

5. Click **Create**.

Deploy CX Contact

Complete the following procedure using Kubectl, the Helm tools, and if you deploy to OpenShift, the OC CLI:

1. Create a new project.
 - For Kubernetes, enter

```
kubectl create namespace cxc
```
 - For OpenShift, enter

```
oc new-project cxcontact
```
2. Create the pull secret.
Use the following code snippets as examples of how to create the default pull secret for Kubernetes and OpenShift
 - For generic Kubernetes:


```
oc create secret docker-registry mycred --docker-server=pureengage.jfrog.io --docker-username= --docker-password=
```

- For OpenShift:

```
oc secrets link default mycred --for=pull
```

3. Download latest version of the CX Contact installation Helm Charts from the artifactory. See the JFrog Platform Artifactory.
4. Extract parameters from chart to see multiple (default) values used to fine-tune the installation.

```
helm install cxc ./cxcontact-.tgz > values.yaml
```

You can apply multiple override values to customize your setup. However, Genesys recommends using minimal overriding values in the installation: For example, `override_values.yaml`

```
configserver:
  user_name: cloudcon
  user_password: cloudcon

cxcontact:

  replicas: 2
  log:
    level: info

  compliance_data:
    cdp_url: false
    cdp_ng:
      gcloud_id: false
      gcloud_secret: false

# override:                                #if connecting to Nexus. Otherwise Dial Manager is off
#   dial-manager:
#     enabled: false
#     nexus:
#       host: ..
#       port: ..
#       api_key: ..                         #required

monitoring:
  enabled: true
  dashboards: false
  alarms: false
  pagerduty: false

redis:
  enabled: true
  cluster: true
  # can be comma-delimited list of redis nodes, for e.g.
  # nodes: redis://redis-nodel:6379,redis://redis-node2:6379,redis://redis-node3:6379
  nodes: redis://infra-redis-redis-cluster.infra.svc.cluster.local:6379
  #use_tls: false
  requirepass: true
  password:

elasticsearch:
  enabled: true
  host: http://elastic-es-http.infra.svc.cluster.local
  port: 9200
```

```
gws:
  # secret in plain text
  client_id: cx_contact
  client_secret: cx_contact

  # GWS Ingress URL
  frontend_host: https://gauth.apps.
  frontend_port: 443

# Services. Will be used for connection to GWS if GWS Internal Ingress URL is disabled
(empty values)
core:
  auth:
    host: http://gauth-auth.gauth
    port: 80
  environment:
    host: http://gauth-environment.gauth
    port: 80
platform:
  ocs:
    host: http://gws-service-proxy.gws
    port: 80
  configuration:
    host: http://gws-service-proxy.gws
    port: 80
  statistics:
    host: http://gws-service-proxy.gws
    port: 80
  setting:
    host: http://gws-service-proxy.gws
    port: 80
  voice:
    host: http://gws-service-proxy.gws
    port: 80

ingress:
  enabled: true
  #tls_enabled: false
  cxc_frontend: cxc.apps.
  annotations:
#   !!!Ingress Session Stickiness is required.
#   Default annotations:
  nginx.ingress.kubernetes.io/affinity: cookie
  nginx.ingress.kubernetes.io/session-cookie-samesite: "Lax"
  nginx.ingress.kubernetes.io/session-cookie-name: "cxc-session-cookie"
  nginx.ingress.kubernetes.io/proxy-body-size: "0"
  tls: []
# - hosts:
#   - chart-example.local
#   secretName: chart-example-tls

# Additional ingress to expose internal backend endpoints.
# If disabled - all endpoints will be exposed on ingress.cxc_frontend
internal_ingress:
  enabled: true
  tls_enabled: false
  cxc_backend: cxc-int.apps.
  annotations:
#   Default annotations:
  nginx.ingress.kubernetes.io/proxy-body-size: "0"
```

```
    nginx.ingress.kubernetes.io/ssl-redirect: 'false'
  tls: []
  # - hosts:
  #   - chart-example.local
  #   secretName: chart-example-tls

storage:
  # Persistent Volumes Claim Configuration
  pvc:
    enabled: true
    create: true
  # Instructs Helm to skip deleting PVC when a helm operation (such as helm uninstall,
  # helm upgrade or helm rollback)
  # would result in its deletion. However, this resource becomes orphaned. Helm will no
  # longer manage it in any way.
  # https://helm.sh/docs/howto/charts_tips_and_tricks/#tell-helm-not-to-uninstall-a-
  # resource
  # If PVC is already orphaned and you want to re-use it - set `storage.pvc.create` to
  # `false`.
  keepAfterDeletion: false
  size: 10Gi
  name: cxc-claim
  storageClassName: cxc-storage
```

5. Validate the Helm chart and provided values, enter:

```
$ helm template cxc ./cxcontact-.tgz -f override_values.yaml
```

6. Install the CX Contact chart, using the override values file that you prepared in step, enter:

```
$ helm install cxc ./cxcontact-.tgz -f override_values.yaml
```

7. If errors occur, verify the input values, YAML files syntax, and your Kubernetes context. enter:

```
$ kubectl config get-contexts
$ oc status
```

8. If troubleshooting is necessary, try adding the **--dry-run** command line parameter in **helm install ..** for verbose error output.

Tip

- To see the full set of available parameters, extract the default helm values from helm package:

```
$ helm show values cxcontact-.tgz > values.yaml
```

- For persistent volume claims in production, Genesys recommends 100-200 GB.

This completes the CX Contact shared service installation.

Next steps:

- Provision tenant for CX Contact/outbound. See Provision CX Contact.

- Validate the deployment.

Validate the deployment in OpenShift

1. Watch the helm output at the end of installation. It provides the status and additional information about where to log in to the CX Contact UI. See the following sample output:

```
Release "cxc" has been upgraded. Happy Helming!  
NAME: cxc  
LAST DEPLOYED: Tue Jul 13 10:18:07 2021  
NAMESPACE: cxc  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None  
NOTES:  
Please be patient while CXContact is being deployed  
  
ENDPOINTS:  
UI: http://cxc.apps./ui/cxcontact/#!/campaign/list  
API basepath: http://cxc.apps./cx-contact/v3/  
API swagger doc: http://cxc.apps./cx-contact/v3/explore  
CXC backend basepath: http://cxc-int.apps./
```

2. Check the status of the Events and Pods in the OpenShift Administration Console. Pods should be up and running—8 CX Contact components in total (7, if you have not defined Nexus for digital outbound).
3. Check the **amark-app** and other pods logs, primarily for errors related to connectivity to Redis and Elasticsearch.
4. If you have provisioned tenants for CX Contact, log in to the CX Contact UI (use URL from the Helm output above) using the tenant's administrator credentials. If you can log in successfully, that confirms that CX Contact works with the Redis cluster:
 - Check the CX Contact **About > Versions** page for the health status of the CX Contact components. They should be all green.
 - Check the **Analytics** page. It should show your successful log in, and confirm that CX Contact works with Elasticsearch.
 - Try to create a test Contact list. If you succeed, CX Contact will display a success confirmation message.

Configure monitoring and logging

CX Contact monitoring is enabled by default.

See monitoring details and how to configure logging parameters in Observability in Outbound (CX Contact).

Provision CX Contact

Contents

- [1 Prerequisites](#)
- [2 Tenant provisioning](#)
 - [2.1 Validate tenant provisioning](#)

- Administrator

Learn how to provision CX Contact.

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Prerequisites

Before you begin to provision tenants in CX Contact, ensure the following prerequisites are met:

- Ensure CX Contact is deployed. See [Deploy CX Contact](#).
- Ensure the tenant exists in the GWS environment. For example, on the local machine, enter:

`$ curl -u https:///environment/v3/environments`
Here's a sample output:

```
{
  "status": {
    "code": 0
  },
  "data": {
    "genesysEnvironments": [
      {
        "id": "9350e2fc-a1dd-4c65-8d40-1f75a2e080dd",
        "tenant": "Environment",
        "appName": "Cloud",
        "username": "default",
        "password": "password",
        "connectionProtocol": "addp",
        "localTimeout": 5,
        "remoteTimeout": 7,
        "traceMode": "CFGTMBoth",
        "tlsEnabled": false,
        "configServers": [
          {
            "primaryAddress": "tenant-9350e2fc-a1dd-4c65-8d40-1f75a2e080dd.voice.svc.cluster.local",
            "primaryPort": 8888,
            "readOnly": false,
            "locations": "/USW1",
            "readFromDb": false,
            "useConfigExporter": false,
            "initDb": false
          }
        ],
        "proxyPort": 0
      }
    ]
  }
}
```

- Ensure the "cxc" Helm release is deployed (during installation of CX Contact). For example:

```
$ helm ls
NAME      NAMESPACE   REVISION   UPDATED           STATUS      CHART              APP VERSION
cxc       cxc          2          2021-07-15 00:44:22.190262 -0700 PDT  deployed   cxcontact-026.03.242  9.0.026.03
```

Tenant provisioning

To provision tenants, you'll use the same Helm Charts as you used when deploying CX Contact, adding one additional overriding values YAML file. You should still use the same base override values file (**override_values.yaml**) that you used when deploying CX Contact.

Important

The **primary_host** parameter represents the primary Configuration Server's domain

name within the cluster. It's important that this parameter is configured correctly, as it must match the **configServers.primaryAddress** parameter in the GWS environment (see Prerequisites) to ensure the Helm Chart uses the existing environment and environment ID.

1. Prepare the provisioning_values.yaml file, as follows:

```

provisioning_values.yaml

# CXContact Tenant Provisioning configuration
tenant_provisioning:
  enabled: true
  # Basic Authentication for GWS Services. Required if `create_auth_client: true` or `create_environment: true`
  # Should be plain text
  gws_basic_auth_user: ops
  gws_basic_auth_pass: ops
  # Tenants list, that should be configured by CXC Tenant Provisioning. May contain multiple tenants
  tenants:
    # Tenant 0
    - configserver:
      # if set to 'true' - will create environment if it doesn't exist. Else will re-use existing.
      # if set to 'false' - will NOT create environment if it doesn't exist. Will use existing.
      create_environment: true
      # should be unique
      primary_host:

      primary_port: 8888
      backup_host:
      backup_port:
      # Username and Password that will be used for creation of environment. Should exist.
      username: default
      password: password
      # Configserver location e.g /USW1
      # corresponds to gws_configuration ENV GWS_CONFIGURATION_COMMON_LOCATION
      location: /USW1
      # GWS Server application name. Standard name is `CloudCluster`
      server_app_name: CloudCluster
      # GWS Client application name for GWS Connection. Standard name is `Cloud`
      client_app_name: Cloud
      # Outbound Contact Server Application Name
      ocs_name: OCS
      # Database Access Point Application Name
      ocs_dap_name: OCSDAP
      # CXContact requires set of options to be configured for OCS and CloudCluster applications.
      # Will not update app options if set to false.
      update_app_options: true
      # The short tenant name (for example 22-06), should be unique
      short_tenant_name: ten100
      # The customer name (for example cxc), should be unique
      customer_name: Tenant100
      # Domain, will be used for login, should be unique
      domain: t100

```

2. Validate the Helm Chart and values. Enter:


```
$ helm template cxc ./cxcontact-.tgz -f override_values.yaml -f provisioning_values.yaml
```
3. Upgrade the existing CX Contact Helm deployment with provisioning using the values file that you've just prepared. Enter:


```
$ helm upgrade cxc ./cxcontact-.tgz -f override_values.yaml -f provisioning_values.yaml
```
4. If you encounter errors, verify the input values, YAML files syntax, and your Kubernetes context.

Tip

As long as there are no changes to the override values, you can rerun the provisioning multiple times for the same tenant. It will not affect the CX Contact deployment or corrupt tenant's configuration.

Validate tenant provisioning

At the end of the installation, be sure to check the Helm Chart output. It will provide the status and other information about where to log in to the CX Contact UI. In addition to the standard CX Contact installation output you will see the following provisioning information:

```
Following tenants were provisioned:

0) tenant-9350e2fc-a1dd-4c65-8d40-1f75a2e080dd.voice.svc.cluster.local
Domain for login to this tenant: t100
Test Username: t100\cxc_genesys@Tenant100.com
Provisioning logs can be accessed via `kubectl logs` command:
  $ kubectl -n cxc logs -f -l job-name=cxc-provisioning-0 -c cxc-provi
sioning --tail 9999

* Password is configured in helm cxc overrides, see variable configserver.user_password
```

Recommendations

Finally, note the following recommendations:

- In addition to the above validation, you can also use the OpenShift Management Console to view the provisioning pod logs.
- Use the OpenShift Administration Console to check the statuses of Events and Pods. The Pods should be up and running.
- Log in to the CX Contact UI using the URL from the Helm Chart output above and the provisioned tenant's Administrator credentials.
- Check the CX Contact **About > Versions** page, which contains the health statuses of the CX Contact components. They should all be green.
- Check CX Contact **Analytics** page, which should show your successful log in.
- Try to create a test Contact List. If you do it right, CX Contact will display a confirmation message that you were successful.

Upgrade, rollback, or uninstall CX Contact

Contents

- [1 Upgrade CX Contact](#)
- [2 Rollback CX Contact](#)
- [3 Uninstall CX Contact](#)

Learn how to upgrade, rollback or uninstall CX Contact.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Upgrade CX Contact

To upgrade the CX Contact service, in the local machine's Command Line Interface (CLI), enter:

```
helm upgrade cxc cxcontact-.tgz -f overrides.yaml
```

Rollback CX Contact

To rollback the CX Contact service, in the local machine's CLI, enter:

```
helm rollback cxc
```

Uninstall CX Contact

To uninstall the CX Contact service, in the local machine's CLI, enter:

```
helm uninstall cxc
```

Observability in Outbound (CX Contact)

Contents

- **1 Monitoring**
 - **1.1 Enable monitoring**
 - **1.2 Configure metrics**
- **2 Alerting**
- **3 Logging**
 - **3.1 Setting the logging parameters**

Learn about the logs, metrics, and alerts you should monitor for Outbound (CX Contact).

Related documentation:

-
-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Monitoring

Private edition services expose metrics that can be scraped by Prometheus, to support monitoring operations and alerting.

- As described on Monitoring overview and approach, you can use a tool like Grafana to create dashboards that query the Prometheus metrics to visualize operational status.
- As described on Customizing Alertmanager configuration, you can configure Alertmanager to send notifications to notification providers such as PagerDuty, to notify you when an alert is triggered because a metric has exceeded a defined threshold.

The services expose a number of Genesys-defined and third-party metrics. The metrics that are defined in third-party software used by private edition services are available for you to use as long as the third-party provider still supports them. For descriptions of available Outbound (CX Contact) metrics, see:

- CX Contact API Aggregator metrics
- CX Contact Campaign Manager metrics
- CX Contact Compliance Manager metrics
- CX Contact Dial Manager metrics
- CX Contact Job Scheduler metrics
- CX Contact List Builder metrics
- CX Contact List Manager metrics

See also System metrics.

Enable monitoring

CX Contact monitoring is enabled by default.

The following Kubernetes objects are created, based on default parameter settings in the Helm chart:

- **ServiceMonitor**—Prometheus operator uses this object to auto-discover endpoints for metrics scraping.
- **PrometheusRule**—AlertManager uses this object to import alert rules.
- Several **ConfigMaps**—Grafana uses these objects to import dashboards.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
CX Contact API Aggregator	ServiceMonitor	9102	/metrics	15 seconds
CX Contact Campaign Manager	ServiceMonitor	3106	/metrics	15 seconds
CX Contact Compliance Manager	ServiceMonitor	3107	/metrics	15 seconds
CX Contact Dial Manager	ServiceMonitor	3109	/metrics	15 seconds
CX Contact Job Scheduler	ServiceMonitor	3108	/metrics	15 seconds
CX Contact List Builder	ServiceMonitor	3104	/metrics	15 seconds
CX Contact List Manager	ServiceMonitor	3105	/metrics	15 seconds

Configure metrics

The metrics that are exposed by the CX Contact services are available by default. No further configuration is required in order to define or expose these metrics. You cannot define your own custom metrics.

The Metrics pages linked to above show the metrics the CX Contact services expose. You can also query Prometheus directly or via a dashboard to see all the metrics available from the CX Contact services.

Alerting

No alerts are defined for Outbound (CX Contact).

Logging

Setting the logging parameters

Set/override the logging-related parameters to change the default Helm chart values as follows, if required:

```
cxcontact:
  log:
    level: info
    # logs can be saved to .log files on log volume.
    log_to_file: false
    # Log rotation.
    # false - will store logs indefinitely, needs rotation configured on the volume level.
    # true - will keep last 10 files with size up to 100mb
    log_rotation: true
    # Log volume configuration, If using persistentVolumeClaim - it should be created outside
of helm chart.
    log_volume_config:
      hostPath:
        # path on k8s nodes, that will be mounted to the pods
        # IMPORTANT! Should allow Write access to user with uid:guid 500:500!
        path: /mnt/log/cxc
```

CX Contact logs to stdout, by default. You can override this setting by changing the following parameter value to true:

```
log_to_file=true
```


API Aggregator metrics and alerts

Find the metrics APIA exposes and the alerts defined for APIA.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
API Aggregator	ServiceMonitor	9102	/metrics	15 seconds

See details about:

- API Aggregator metrics
- API Aggregator alerts

Metrics

Add some introductory text... TBD.

Metric and description	Metric details	Indicator of
cxc_api_aggregator_schedules_created_total Total schedules created.	Unit: Type: Counter Label: "'ccid', 'tenant_name'" Sample value: 42	
cxc_api_aggregator_schedules_removed_total Total schedules removed.	Unit: Type: Counter Label: "'ccid', 'tenant_name'" Sample value: 42	
cxc_api_aggregator_campaign_template_created_total Total campaign templates created.	Unit: Type: Counter Label: "'ccid', 'tenant_name'" Sample value: 42	
cxc_api_aggregator_campaign_template_removed_total Total campaign templates removed.	Unit: Type: Counter Label: Sample value: 42	
cxc_api_aggregator_users_logged_in_total Total logged in users.	Unit: Type: Gauge Label: "'ccid', 'tenant_name'" Sample value: 4.2	
cxc_api_aggregator_users_logged_out_total Total logged out users.	Unit: Type: Gauge Label: "'ccid', 'tenant_name', 'service_name'" Sample value: 4.2	
cxc_api_aggregator_api_requests_total	Unit:	

Metric and description	Metric details	Indicator of
Total count of requests.	Type: Counter Label: "'ccid', 'tenant_name'" Sample value: 42	
cxc_api_healthy_instance Healthy instance.	Unit: Type: Gauge Label: "'ccid', 'tenant_name'" Sample value: 4.2	
cxc_api_aggregator_api_requests_processed_success Total count of success requests.	Unit: Type: Counter Label: "'ccid', 'tenant_name'" Sample value: 42	
cxc_api_aggregator_top_api_requests Top api requests.	Unit: Type: Counter Label: "'path', 'method', 'id', 'name', 'ccid', 'tenant_name', 'code'" Sample value: 42	
cxc_api_aggregator_redis_connection_failed Failed Redis connection.	Unit: Type: Gauge Label: "'ccid', 'tenant_name'" Sample value: 4.2	
cxc_api_aggregator_request_count Total requests by verb and code.	Unit: Type: Counter Label: "'method', 'path', 'code'" Sample value: 42	
cxc_api_aggregator_request_latencies_ms Request latencies histogram by verb, in milliseconds.	Unit: Type: Histogram Label: "'method', 'path', 'code'" Sample value: [1, 2, 3]	
cxc_api_aggregator_request_out_count Total out requests by verb, destination and code.	Unit: Type: Counter Label: "'method', 'destination', 'code'" Sample value: 42	
cxc_api_aggregator_request_out_latencies_ms Out Request latencies histogram by verb, destination and code, in milliseconds.	Unit: Type: Histogram Label: "'method', 'destination', 'code'" Sample value: [1, 2, 3]	
cxc_api_aggregator_elasticsearch_service_latencies_ms Elasticsearch Request latencies histogram by verb, destination and code, in milliseconds.	Unit: Type: Histogram Label: "'method', 'destination', 'code'" Sample value: [1, 2, 3]	

Alerts

Campaign Manager metrics and alerts

Find the metrics CPGM exposes and the alerts defined for CPGM.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Campaign Manager	ServiceMonitor	3106	/metrics	15 seconds

See details about:

- Campaign Manager metrics
- Campaign Manager alerts

Metrics

Metric and description	Metric details	Indicator of
cxc_cm_campaign_group_created_total Total campaign groups created.	Unit: Type: Counter Label: ""ccid", 'tenant_name"" Sample value: 42	
cxc_cm_campaign_group_removed_total Total campaign groups removed.	Unit: Type: Counter Label: ""ccid", 'tenant_name"" Sample value: 42	
cxc_cm_campaign_group_running_total Total campaign groups are running.	Unit: Type: Gauge Label: ""ccid", 'tenant_name"" Sample value: 4.2	
cxc_cm_healthy_instance Healthy instance.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_cm_campaign_group_active_total Campaign group active total.	Unit: Type: Gauge Label: ""ccid", 'tenant_name"" Sample value: 4.2	
cxc_cm_schedule_item_running_total Campaign group with schedule running total.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_cm_schedule_item_active_total Campaign group with schedule active total.	Unit: Type: Gauge Label: n/a Sample value: 4.2	

Metric and description	Metric details	Indicator of
<p>cxc_cm_request_count</p> <p>Total requests by verb and code.</p>	<p>Unit:</p> <p>Type: Counter Label: ""method', 'path', 'code" Sample value: 42</p>	
<p>cxc_cm_request_latencies_ms</p> <p>Request latencies histogram by verb in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""method', 'path', 'code" Sample value: [1, 2, 3]</p>	
<p>cxc_cm_request_out_count</p> <p>Total out requests by verb destination and code.</p>	<p>Unit:</p> <p>Type: Counter Label: ""method', 'destination', 'code" Sample value: 42</p>	
<p>cxc_cm_request_out_latencies_ms</p> <p>Out Request latencies histogram by verb destination and code, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""method', 'destination', 'code" Sample value: [1, 2, 3]</p>	
<p>cxc_cm_elasticsearch_service_latencies_ms</p> <p>Elasticsearch Request latencies histogram by verb destination and code, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""method', 'destination', 'code" Sample value: [1, 2, 3]</p>	

Alerts

Compliance Manager metrics and alerts

Find the metrics CPLM exposes and the alerts defined for CPLM.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Compliance Manager	ServiceMonitor	3107	/metrics	15 seconds

See details about:

- Compliance Manager metrics
- Compliance Manager alerts

Metrics

Metric and description	Metric details	Indicator of
compliance_api_history_requests_total Total number of history API calls.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name'" Sample value: 42	
compliance_validation_under_processing_total Total number validation requests are under processing.	Unit: Type: Gauge Label: ""type', 'ccid', 'tenant_name'" Sample value: 4.2	
compliance_validation_complete_total Total number of completed validation calls.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name'" Sample value: 42	
compliance_validation_success_total Number of validated requests with Success status.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name'" Sample value: 42	
compliance_validation_failed_total Number of validation requests with Failed status.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name'" Sample value: 42	
compliance_validation_success_by_tenant Number of validation requests by Tenant with Success result.	Unit: Type: Counter Label: ""type', 'tenant_name'" Sample value: 42	
compliance_validation_failed_by_tenant Number of validation requests by Tenant with Fail result.	Unit: Type: Counter Label: ""type', 'tenant_name'" Sample value: 4.2	

Metric and description	Metric details	Indicator of
cxc_compliance_healthy_instance Healthy instance.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_compliance_request_latencies_ms The latencies of all HTTP requests distributed by method, plus path and HTTP response code.	Unit: Type: Histogram Label: "'method', 'path', 'code'" Sample value: [1, 2, 3]	
cxc_compliance_request_count The number of all HTTP requests distributed by method, plus path and HTTP response code.	Unit: Type: Counter Label: "'method', 'path', 'code'" Sample value: 42	
compliance_redis_connections_made Total number of Redis connections made.	Unit: Type: Counter Label: n/a Sample value: 42	
compliance_redis_connections_closed Total number of Redis connections closed. Current can be calculated with the help of compliance_redis_connections_made.	Unit: Type: Counter Label: n/a Sample value: 42	
compliance_redis_access_errors Total number of reported REDIS errors.	Unit: Type: Counter Label: n/a Sample value: 42	
compliance_ocs_calls_placed Total number of calls placed by OCS broken by GSW_CALL_RESULT.	Unit: Type: Counter Label: 'GSW_CALL_RESULT' Sample value: 42	
cxc_compliance_request_out_count Total Out Requests by verb, destination, and code.	Unit: Type: Counter Label: "'method', 'path', 'code'" Sample value: 42	
cxc_compliance_request_out_latencies_ms Out Request latencies histogram by verb, destination, and code, in milliseconds.	Unit: Type: Histogram Label: "'method', 'path', 'code'" Sample value: [1, 2, 3]	
cxc_dm_elasticsearch_service_latencies_ms Elasticsearch Request latencies histogram by verb, destination, and code, in milliseconds.	Unit: Type: Histogram Label: n/a Sample value: [1, 2, 3]	

Metric and description	Metric details	Indicator of
<p>cxc_compliance_validation_rate_limit_reached</p> <p>Total number of validation requests rejected due to rate limit exceeded, broken by customer (tenant) and a limit reason {device, customerId, overall}.</p>	<p>Unit:</p> <p>Type: Counter Label: "ccid", "reason" Sample value: 42</p>	

Alerts

Dial Manager metrics and alerts

Find the metrics DM exposes and the alerts defined for DM.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Dial Manager	ServiceMonitor	3109	/metrics	15 seconds

See details about:

- Dial Manager metrics
- Dial Manager alerts

Metrics

Metric and description	Metric details	Indicator of
cxc_dm_healthy_instance Healthy instance.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_dm_processed_batches_total Total processed batches.	Unit: Type: Counter Label: ""media', 'ccid', 'tenant_name" Sample value: 42	
cxc_dm_processed_messages_total Total processed messages.	Unit: Type: Counter Label: ""media', 'ccid', 'tenant_name" Sample value: 42	
cxc_dm_opt_out_messages_total Total opt out messages.	Unit: Type: Counter Label: ""media', 'ccid', 'tenant_name" Sample value: 42	
cxc_dm_failed_processed_messages_total Total failed messages.	Unit: Type: Counter Label: ""media', 'ccid', 'tenant_name" Sample value: 42	
cxc_dm_batch_size Batch size histogram.	Unit: Type: Histogram Label: ""media', 'ccid', 'tenant_name" Sample value: [1, 2, 3]	
cxc_dm_process_message_duration_seconds Processing message duration histogram.	Unit: Type: Histogram Label: ""media', 'ccid', 'tenant_name" Sample value: [1, 2, 3]	

Metric and description	Metric details	Indicator of
<p>cxc_dm_delivery_buffer_size</p> <p>Delivery buffer size.</p>	<p>Unit:</p> <p>Type: Gauge Label: 'media' Sample value: 4.2</p>	
<p>cxc_dm_test_messages_total</p> <p>Total test messages.</p>	<p>Unit:</p> <p>Type: Counter Label: "'media', 'ccid', 'tenant_name'" Sample value: 42</p>	
<p>cxc_dm_failed_test_messages_total</p> <p>Total failed test messages.</p>	<p>Unit:</p> <p>Type: Counter Label: "'media', 'ccid', 'tenant_name'" Sample value: 42</p>	
<p>cxc_dm_nexus_service_status</p> <p>The current status of the connection to the Nexus service.</p>	<p>Unit:</p> <p>Type: Gauge Label: "'ccid', 'tenant_name'" Sample value: 4.2</p>	
<p>cxc_dm_request_count</p> <p>Total requests made to Nexus via websocket.</p>	<p>Unit:</p> <p>Type: Counter Label: "'media', 'ccid', 'tenant_name', 'code'" Sample value: 42</p>	
<p>cxc_dm_request_latencies_ms</p> <p>Request latencies histogram by tenant, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: Sample value: [1, 2, 3]</p>	
<p>cxc_dm_request_out_count</p> <p>Total out requests by verb, destination, and code.</p>	<p>Unit:</p> <p>Type: Counter Label: "'method', 'destination', 'code'" Sample value: 42</p>	
<p>cxc_dm_request_out_latencies_ms</p> <p>Out Request latencies histogram by verb, destination, and code, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: Sample value: [1, 2, 3]</p>	
<p>cxc_dm_elasticsearch_service_latencies_ms</p> <p>Elasticsearch Request latencies histogram by verb, destination, and code, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: "'method', 'destination', 'code'" Sample value: [1, 2, 3]</p>	

Alerts

Job Scheduler metrics and alerts

Find the metrics JS exposes and the alerts defined for JS.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Job Scheduler	ServiceMonitor	3108	/metrics	15 seconds

See details about:

- Job Scheduler metrics
- Job Scheduler alerts

Metrics

Metric and description	Metric details	Indicator of
cxc_js_jobs_executed_total Total jobs executed.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name" Sample value: 42	
cxc_js_jobs_failed_total Total failed jobs.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name" Sample value: 42	
cxc_js_jobs_success_total Total successful jobs.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name" Sample value: 42	
cxc_js_jobs_nothing_to_do_total Total jobs with Nothing TO DO result.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name" Sample value: 42	
cxc_js_jobs_run_now_total Total jobs that were started manually.	Unit: Type: Counter Label: ""ccid', 'tenant_name" Sample value: 42	
cxc_js_files_imported_total Total files imported.	Unit: Type: Counter Label: ""action', 'ccid', 'tenant_name" Sample value: 42	
cxc_js_jobs_ttl_exceeded_total Total ttl exceeded jobs.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name" Sample value: 42	

Metric and description	Metric details	Indicator of
<p>cxc_js_jobs_running_total</p> <p>Number of currently active jobs.</p>	<p>Unit:</p> <p>Type: Gauge Label: ""type', 'ccid', 'tenant_name"" Sample value: 4.2</p>	
<p>cxc_js_redis_connections</p> <p>Count of active connections to Redis server.</p>	<p>Unit:</p> <p>Type: Gauge Label: n/a Sample value: 4.2</p>	
<p>cxc_js_job_duration_seconds</p> <p>Job duration histogram.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""type', 'ccid', 'tenant_name"" Sample value: [1, 2, 3]</p>	
<p>cxc_js_job_import_file_size_megabytes</p> <p>Job import file size histogram.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""action', 'ccid', 'tenant_name"" Sample value: [1, 2, 3]</p>	
<p>cxc_js_healthy_instance</p> <p>Healthy instance.</p>	<p>Unit:</p> <p>Type: Gauge Label: n/a Sample value: 4.2</p>	
<p>cxc_js_request_count</p> <p>Total requests by verb and code.</p>	<p>Unit:</p> <p>Type: Counter Label: ""method', 'path', 'code"" Sample value: 42</p>	
<p>cxc_js_request_latencies_ms</p> <p>Request latencies histogram by verb, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""method', 'path', 'code"" Sample value: [1, 2, 3]</p>	
<p>cxc_js_request_out_count</p> <p>Total out requests by verb, destination, and code.</p>	<p>Unit:</p> <p>Type: Counter Label: ""method', 'destination', 'code"" Sample value: 42</p>	
<p>cxc_js_request_out_latencies_ms</p> <p>Out Request latencies histogram by verb, destination, and code, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""method', 'destination', 'code"" Sample value: [1, 2, 3]</p>	
<p>cxc_js_healthy_tenants</p> <p>Healthy tenants.</p>	<p>Unit:</p> <p>Type: Gauge Label: ""ccid', 'tenant_name"" Sample value: 4.2</p>	

Alerts

List Builder metrics and alerts

Find the metrics LB exposes and the alerts defined for LB.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
List Builder	ServiceMonitor	3104	/metrics	15 seconds

See details about:

- List Builder metrics
- List Builder alerts

Metrics

Metric and description	Metric details	Indicator of
cxc_lb_jobs_running_total Number of currently active jobs.	Unit: Type: Counter Label: ""type', 'ccid', 'tenant_name'" Sample value: [1, 2, 3]	
cxc_lb_contacts_imported_total Total contacts imported.	Unit: Type: Counter Label: ""ccid', 'tenant_name'" Sample value: 42	
cxc_lb_devices_imported_total Total device imported.	Unit: Type: Counter Label: ""ccid', 'tenant_name'" Sample value: 42	
cxc_lb_rejected_contact_lines_total Total of rejected lines in input contact list files.	Unit: Type: Counter Label: ""ccid', 'tenant_name'" Sample value: 42	
cxc_lb_healthy_instance Healthy instance.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_lb_request_count Total requests by verb and code.	Unit: Type: Counter Label: ""method', 'path', 'code'" Sample value: 42	
cxc_lb_request_latencies_ms Request latencies histogram by verb, in milliseconds.	Unit: Type: Histogram Label: ""method', 'path', 'code'" Sample value: [1, 2, 3]	

Metric and description	Metric details	Indicator of
<p>cxc_lb_job_count</p> <p>Total jobs.</p>	<p>Unit:</p> <p>Type: Counter Label: ""type', 'result', 'ccid', 'tenant_name'" Sample value: 42</p>	
<p>cxc_lb_job_duration_seconds</p> <p>Jobs duration histogram in seconds</p>	<p>Unit:</p> <p>Type: Histogram Label: ""type', 'ccid', 'tenant_name'" Sample value: [1, 2, 3]</p>	
<p>cxc_lb_request_out_count</p> <p>Total out requests by verb, destination, and code.</p>	<p>Unit:</p> <p>Type: Counter Label: ""method', 'destination', 'code'" Sample value: 42</p>	
<p>cxc_lb_request_out_latencies_ms</p> <p>Out Request latencies histogram by verb, destination, and code, in milliseconds.</p>	<p>Unit:</p> <p>Type: Histogram Label: ""method', 'destination', 'code'" Sample value: [1, 2, 3]</p>	

Alerts

List Manager metrics and alerts

Find the metrics LM exposes and the alerts defined for LM.

Related documentation:

-

Early Adopter Program

Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

Contents

- [1 Metrics](#)
- [2 Alerts](#)

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
List Manager	ServiceMonitor	3105	/metrics	15 seconds

See details about:

- List Manager metrics
- List Manager alerts

Metrics

Metric and description	Metric details	Indicator of
cxc_list_manager_executed_jobs_count Total executed jobs count.	Unit: Type: Counter Label: n/a Sample value: 42	
cxc_list_manager_running_jobs_count Running jobs count.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_list_manager_rejected_jobs_count Rejected jobs count.	Unit: Type: Counter Label: n/a Sample value: 42	
cxc_list_manager_jobs_duration Job duration, in milliseconds.	Unit: Type: Histogram Label: n/a Sample value: [1, 2, 3]	
cxc_list_manager_responses_summary Response time, in milliseconds.	Unit: Type: Summary Label: "'method', 'path', 'status'" Sample value: 42	
cxc_list_manager_healthy_instance Healthy instance.	Unit: Type: Gauge Label: n/a Sample value: 4.2	
cxc_list_manager_downloaded_compliance_files_count Count of downloaded compliance files.	Unit: Type: Counter Label: n/a Sample value: 42	

Metric and description	Metric details	Indicator of
cxc_list_manager_contacts_lists_created_count Count of created Contacts Lists.	Unit: created_count Type: Counter Label: ""ccid','tenant_name"" Sample value: 42	
cxc_list_manager_import_contacts_requests_processed_count Count of created Contacts Lists.	Unit: requests_processed_count Type: Counter Label: ""ccid','tenant_name"" Sample value: 42	
cxc_list_manager_import_contacts_requests_failed_count Count of created Contacts Lists.	Unit: requests_failed_count Type: Counter Label: ""ccid','tenant_name"" Sample value: 42	

Alerts