



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.

## Digital Channels Private Edition Guide

# Table of Contents

<b>Overview</b>	
About Digital Channels	6
Architecture	8
High availability and disaster recovery	10
<b>Configure and deploy</b>	
Before you begin	11
Configure Digital Channels	15
Deploy Digital Channels	22
Upgrade, rollback, or uninstall Digital Channels	27
<b>Integrate and provision</b>	
Pre-configure tenant objects	29
Enable a tenant for Digital Channels	38
Provision API keys	50
Provision SMS	57
<b>Observability</b>	
Observability in Digital Channels	74
Digital Channels metrics and alerts	78

---

## Contents

- [1 Overview](#)
- [2 Configure and deploy](#)
- [3 Integrate and provision](#)
- [4 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.

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

## Overview

Learn more about Digital Channels, its architecture, and how to support high availability and disaster recovery.

- About Digital Channels
- Architecture
- High availability and disaster recovery

---

## Configure and deploy

Find out how to configure and deploy Digital Channels.

- Before you begin
- Configure Digital Channels
- Deploy Digital Channels
- Upgrade, rollback, or uninstall Digital Channels

---

## Integrate and provision

Learn how to integrate with the tenant and provision API keys and SMS.

- Pre-configure tenant objects
- Enable a tenant for Digital Channels
- Provision API keys
- Provision SMS

---

## Observability

Learn how to monitor Digital Channels with metrics and logging.

- Observability in Digital Channels
- Digital Channels metrics and alerts

# About Digital Channels

## Contents

- [1 Supported Kubernetes platforms](#)

Learn about Digital Channels 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.

Digital Channels powers your customer interactions across the chat and SMS channels. It provides a platform that enables you to grow sales, create more targeted marketing campaigns, and deliver exceptional customer service. The Digital Channels service processes, manages and archives customer and agent interactions across media.

Chats and SMS are treated just like regular Genesys interactions. When customers communicate with your company on one of these channels, Genesys matches them against customers already in the contact database. If there's a match, the agent handling the interaction has access to all previous interactions with the contact. Until the interaction is marked Done, agents can also return to the chat conversation at any time in the future — for example, they might need to take time to find additional information for the contact or initiate a business process in your company.

## Supported Kubernetes platforms

Digital Channels is supported on the following cloud platforms:

- OpenShift Container Platform (OpenShift)

See the Digital Channels Release Notes for information about when support was introduced.

# Architecture

Learn about Digital Channels architecture.

**Related documentation:**

- 

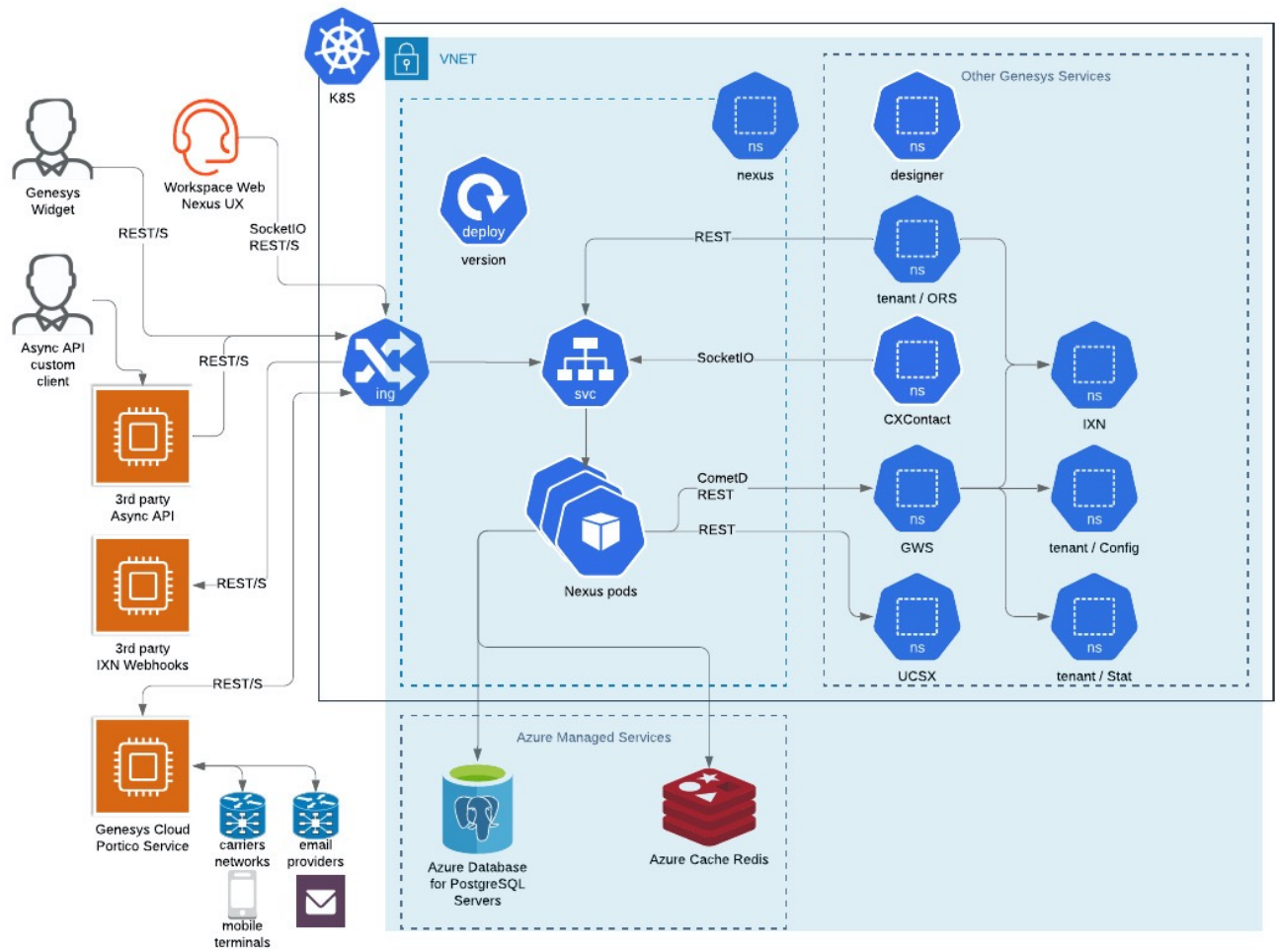
**Early Adopter Program**

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.

The following diagram shows the high-level architecture of Digital Channels (labelled as "nexus").

For more information about the overall architecture of Genesys Multicloud CX private edition, see the [high-level Architecture page](#).





# 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?
Digital Channels	N = N (N+1)	Not supported	Primary unit only

*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

# Before you begin

## Contents

- [1 Limitations and assumptions](#)
- [2 Download the Helm charts](#)
- [3 Third-party prerequisites](#)
- [4 Storage requirements](#)
- [5 Network requirements](#)
- [6 Genesys dependencies](#)

Find out what to do before deploying Digital Channels.

**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

Digital Channels for private edition has the following limitations:

- Supports only a single-region model of deployment.
- Social media requires additional components that are not included in Digital Channels.

## Download the Helm charts

Digital Channels in Genesys Multicloud CX private edition includes the following containers:

- nexus
- hubpp
- tenant\_deployment

The service also includes a Helm chart, which you must deploy to install all the containers for Digital Channels:

- nexus

See Helm charts and containers for Digital Channels for the Helm chart version you must download for your release.

To download the Helm chart, navigate to the **nexus** folder in the JFrog repository. For information about how to download the Helm charts, see [Downloading your Genesys Multicloud CX containers](#).

## Third-party prerequisites

Install the prerequisite dependencies listed in the **Third-party services** table before you deploy Digital Channels.

Third-party services

Name	Version	OpenShift	GKE	Purpose	Shared service?	Notes
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	Digital Channels supports Redis deployed in either cluster (TLS) or non-cluster (non-TLS) mode.
PostgreSQL	11.x			Relational database.	Optional	No support for enforced SSL.

## Storage requirements

Digital Channels uses PostgreSQL and Redis to store all data.

## Network requirements

For general network requirements, review the information on the suite-level Network settings page.

## Genesys dependencies

Digital Channels has dependencies on the following Genesys services:

## Before you begin

---

- Genesys Authentication
- Web Services and Applications
- Tenant Microservice
- Universal Contact Service
- Designer

For detailed information about the correct order of services deployment, see [Order of services deployment](#).

# Configure Digital Channels

## Contents

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

Learn how to configure Digital Channels.

**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.

Complete the steps on this page to configure your Digital Channels deployment.

## 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 how to override Helm chart values, see *Overriding Helm chart values in the Setting up Genesys Multicloud CX 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 *Configure security* below.

Parameters

Parameter	Description	Valid values	Default
global.imageRegistry	The Docker registry from which Kubernetes pulls images.	A valid registry URL	nil
global.imagePullSecrets	An array of global docker-registry secret names.	An array of secret names	[] (does not add image pull secrets to deployed pods)
global.storageClass	The global storage class used for dynamic provisioning.	A valid storage class	nil
image.registry	The Nexus image registry.	A valid registry URL	TBD
image.repository	The Nexus image name.	A valid image name	nexus/nexus



Parameter	Description	Valid values	Default
image.tag	The Nexus image tag.	A valid image tag	{TAG_NAME}
image.pullPolicy	Specifies when Kubernetes pulls images from the registry on start up.	IfNotPresent or Always	IfNotPresent
image.pullSecrets	An array of docker-registry secret names.	An array of secret names	[] (does not add image pull secrets to deployed pods)
nameOverride	A string to partially override the nexus.fullname template. This string is prepended to the release name.	String	nil
fullnameOverride	A string to fully override the nexus.fullname template.	String	nil
nexus.redirectProtocol	Defines the Web Services and Applications to Nexus redirect protocol (HTTP or HTTPS).	A valid protocol	http://
nexus.redis.enabled	Specifies whether to use Redis. You must not changes this from the default value of true.	true	true
nexus.redis.nodes	A comma-separated list of Redis nodes to connect.	A valid URL	redis://nexus-redis-master.default.svc.cluster.local:6379
nexus.redis.useCluster	Specifies whether to deploy Redis as a cluster.	true or false	false
nexus.redis.enableTls	Specifies whether to use TLS on the Redis connection.	true or false	false
nexus.redis.password	The password for Redis authentication.	A valid password	""
nexus.db.host	The Postgres service URL.	A valid URL	nexus-postgres-postgresql.default.svc.cluster.local
nexus.db.port	The Postgres service port.	A valid port	5432
nexus.db.user	The user assigned for the Nexus application to access Postgres.	A valid user	nexus
nexus.db.password	The password assigned for the Nexus application to access Postgres.	A valid password	nexus

Parameter	Description	Valid values	Default
podSecurityContext.runAsNonRoot	Specifies whether the container must run as a non-root user.	true or false	true
podSecurityContext.runAsUser	The user ID to run the entry point of the container process. In OpenShift, if your deployment uses arbitrary UIDs set this value to null.	A valid user ID	500
podSecurityContext.runAsGroup	The group ID to run the entry point of the container process. In OpenShift, if your deployment uses arbitrary UIDs set this value to 0.	A valid group ID	500
podSecurityContext.fsGroup	A supplemental group ID that applies to all containers in a pod. In OpenShift, if your deployment uses arbitrary UIDs set this value to null.	A valid group ID	500
resources	The requests and limits for CPU and memory usage in Kubernetes. See the Kubernetes documentation for details.		requests: { cpu: "300m", memory: "512Mi" }
affinity	Specifies the affinity and anti-affinity for Digital Channels pods. See the Kubernetes documentation for details.	Object	{}
nodeSelector	The labels Kubernetes uses to assign pods to nodes. See the Kubernetes documentation for details.	Object	{}
tolerations	The tolerations Kubernetes uses for advanced pod scheduling. See the Kubernetes documentation for details.	Object	[]
priorityClassName	The class name Kubernetes uses to determine the priority of	A valid priority class name	""

Parameter	Description	Valid values	Default
	a pod relative to other pods. See the Kubernetes documentation for details.		
monitoring.enabled	Specifies whether to deploy Custom Resource Definitions (CRD) for ServiceMonitors to determine which services should be monitored.	true or false	false
service.type	The Kubernetes service type.	See the Kubernetes documentation for details.	LoadBalancer
service.port	The Kubernetes service HTTP port.	A valid port	80
service.httpsPort	The Kubernetes service HTTPS port.	A valid port	443
service.nodePorts.http	The Kubernetes service HTTP node port.	A valid port	""
service.nodePorts.https	The Kubernetes service HTTPS node port.	A valid port	""
service.externalTrafficPolicy	Enables client source IP preservation. See the Kubernetes documentation for details.	Cluster or Local	Cluster
service.loadBalancerIP	The IP address of the load balancer service.	A valid IP address	""
ingress.enabled	Enables the ingress controller resource.	true or false	false
ingress.annotations	The ingress annotations.	A valid set of annotations as "name: value"	[]
ingress.certManager	Add annotations for cert-manager.	true or false	false
ingress.hosts[0].name	The hostname of your Nexus installation.	A valid hostname	nexus.local
ingress.hosts[0].path	The path (within the URL structure) to your Nexus installation.	A valid path	/
ingress.hosts[0].tls	Specifies whether to use TLS backend in ingress.	true or false	false
ingress.hosts[0].tlsHosts	An array of TLS hosts for ingress record. If nil, this value defaults to the	Valid hosts	nil

Parameter	Description	Valid values	Default
	value of ingress.hosts[0].name.		
ingress.hosts[0].tlsSecret	The TLS secret (certificates).	A valid secret	nexus.local-tls-secret
ingress.secrets[0].name	The TLS secret name.	A valid name	nil
ingress.secrets[0].certificate	The TLS secret certificate.	A valid certificate	nil
ingress.secrets[0].key	The TLS secret key.	A valid key	nil
podAnnotations	Custom annotations for each pod.	A valid set of labels as "name: value"	{}

## Configure Kubernetes

Digital Channels stores passwords for Redis and the default database as Kubernetes secrets. In the **nexus** namespace, create a secret called "deployment-secrets" using the following YAML:

```
kind: Secret
apiVersion: v1
metadata:
  name: deployment-secrets
  namespace: nexus
stringData:
  nexus_db_password:
  nexus_redis_password:
type: Opaque
```

## Configure security

To learn more about how security is configured for private edition, be sure to read the Permissions and OpenShift security settings topics in the *Setting up Genesys Multicloud CX Private Edition* guide.

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.

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

## 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.

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

For details about these parameters and possible values, see **podSecurityContext.\*** in the Parameters table above.

# Deploy Digital Channels

## Contents

- [1 Prepare your environment](#)
  - [1.1 Configure a secret to access JFrog](#)
- [2 Deploy in OpenShift](#)
- [3 Validate the deployment in OpenShift](#)
- [4 Next steps](#)

Learn how to deploy Digital Channels.

### 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.

#### Important

Make sure to review [Before you begin](#) for the full list of prerequisites required to deploy Digital Channels.

## Prepare your environment

To prepare your environment for the deployment, first confirm the cluster is running:

```
oc get clusterversion
```

Create a new namespace (also called a project) for Digital Channels:

```
oc new-project nexus
```

### Configure a secret to access JFrog

If you haven't done so already, create a secret for accessing the JFrog registry:

```
kubectl create secret docker-registry --docker-server= --docker-username= --docker-password=
```

Now map the secret to the default service account (for Openshift):

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

### Deploy in OpenShift

To deploy Digital Channels in OpenShift, you'll need the Helm package and override files you downloaded in a previous step. Copy **values.yaml** and the Helm package (**nexus.tgz**) to the installation location.

The following example shows how your **values.yaml** file might look:

```
deploymentType: Deployment
replicaCount: 1
image:
  registry: pureengage-docker-staging.jfrog.io
  repository: nexus/nexus
  pullPolicy: IfNotPresent
imagePullSecrets: [ mycred ]
nameOverride: ""
fullNameOverride: ""
existingSecret:
existingConfig:
nexus:
  fqdn: "http://digital."
  redirectProtocol: "http://"
  redis:
    enabled: true
    nodes: "redis://"
    useCluster: true
    enableTls: false
    password: $nexus_redis_password
  db:
    host: ""
    port:
    user: ""
    password: nexus_db_password
    enableSsl: false
  social:
    apikey: ""
    retryTimeout: 10000
podSecurityContext:
  runAsUser: null
  runAsGroup: 0
  fsGroup: null
  runAsNonRoot: true
service:
  enabled: true
  type: ClusterIP
ingress:
  tls:
    - hosts:
      - digital.
      secretName: letsencrypt
  enabled: true
hosts:
  - host: digital.
    paths:
      - path: '/chat/v3/'
        port: http
      - path: '/nexus/v3/'
        port: http
      - path: '/ux/'
        port: http
      - path: '/admin/'
```



```
    port: http
  - path: '/auth/'
    port: http
monitoring:
  enabled: true
  alarms: true
```

Run the following command to install Digital Channels:

```
helm install nexus ./nexus-.tgz --set version= -f values.yaml
```

### Validate the deployment in OpenShift

To validate the deployment, send the following GET request:

```
$nexusURL/health/detail
```

Where **\$nexusURL** is the fully qualified domain name (FQDN) for Digital Channels.

The response should look like this:

```
{
  "buildInfo": {
    "@genesys/nexus-admin-ux": "^1.0.21",
    "@genesys/nexus-ux": "^2.2.46",
    "version": "9.0.001.01.95292",
    "changeset": "8c3a2b34888d41b318d949a4bda2903368cf3bda",
    "timestamp": "Thu Oct 21 13:55:13 UTC 2021"
  },
  "startTime": "2021-10-22T10:40:27.456Z",
  "os": {
    "upTime": 1142897,
    "freemem": 105664512,
    "loadavg": [1.32, 0.68, 0.43],
    "totalmem": 2052542464
  },
  "upTime": 279363374,
  "memoryUsage": {
    "rss": 306659328,
    "heapTotal": 196685824,
    "heapUsed": 162114568,
    "external": 2490373,
    "arrayBuffers": 818214
  },
  "cache": {
    "ready": true,
    "state": "READY"
  },
  "db": {
    "ready": true
  },
  "state": "green"
}
```

The deployment was successful if `state="green"`. You can also confirm that

`db.ready=true` and `cache.ready=true`.

## Next steps

Complete the steps in the "Integrate and provision" chapter to finish deploying Digital Channels.

- Pre-configure tenant objects
- Enable a tenant for Digital Channels
- Provision API keys
- Provision SMS

# Upgrade, rollback, or uninstall Digital Channels

## Contents

- [1 Upgrade Digital Channels](#)
- [2 Rollback Digital Channels](#)
- [3 Uninstall Digital Channels](#)

Learn how to upgrade, rollback or uninstall Digital Channels.

**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 Digital Channels

Use **helm upgrade** to upgrade to a new revision.

## Rollback Digital Channels

Use **helm rollback** to rollback to the previous revision.

## Uninstall Digital Channels

Use **helm delete** to uninstall the deployment.

# Pre-configure tenant objects

## Contents

- [1 Prerequisites](#)
- [2 Create a user](#)
- [3 Configure the Genesys Web Services application](#)
  - [3.1 Enable Nexus UX](#)
- [4 Configure the Universal Contact Server application](#)
  - [4.1 Enable Nexus UX](#)
- [5 Update the Environment tenant](#)
- [6 Create transactions](#)
- [7 Create scripts](#)
  - [7.1 asynchold queue](#)
  - [7.2 asynchold View script](#)
  - [7.3 asynchold application script](#)
  - [7.4 asynchold Submitter script](#)
  - [7.5 undelivered queue](#)
  - [7.6 undelivered View script](#)
  - [7.7 undelivered application script](#)
  - [7.8 undelivered Submitter script](#)
- [8 Create an SMS interaction subtype](#)

Learn how to configure your tenant resources for Digital Channels.

**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.

Complete the steps on this page to configure your tenant resources for Digital Channels.

**Important**

“Nexus” is the simplified name we use for the digital channels application and nodes, so you’ll see that name referenced throughout this document.

## Prerequisites

Review the **Prerequisites** table and make sure you have all the listed information before you get started. The values in this table are referenced later by the name in the Variable column.

Prerequisites

Parameter	Variable	Type	Example	Notes
Digital Channels FQDN URL	\$nexusURL	HTTPS URL string	https://nexus.mydomain.com	The fully qualified domain name (FQDN) for Digital Channels.
Contact center ID	\$ccId	string	578ec98e-f07c-46ad-9675-f36c20e150	The contact center ID provisioned in Web Services and Applications. If you

Parameter	Variable	Type	Example	Notes
				don't have this ID, see Get contact center ID from GWS.
Tenant name	\$tenantName	string	premise_tenant	The tenant name.

## Create a user

Use Agent Setup to create a user with these properties:

- Make sure **Agent** is not checked.
- Set **Name, First Name, Last Name** and **Employee ID** to nexus.
- Set the password.
- Set the System Administrator and Superuser.
- Add the user to the System Administrator and Superuser access groups in the **Member Of** tab.

## Configure the Genesys Web Services application

Contact your Genesys representative to add the following configuration options for Web Services and Applications:

Enable Nexus UX

Section	Option name	Value
[NexusCommunication]	label	Communication
	url	https://\$nexusURL/ux/comm?customername=\$tenantName&ccid=\$ccId
[NexusConversation]	label	Conversation
	url	https://\$nexusURL/ux/conv?iid=\$Interaction.Id&customername=\$tenantName&ccid=\$ccId
[interaction-workspace]	workspace.web-content	NexusCommunication
	interaction.web-content	NexusConversation
	service-client-api.accepted-web-content-origins	*
	service-client-api.allow-full-api	true
	privilege.chat.can-place-on-hold-async	false



### Important

When enabling the Conversation tab you must also add the Communication tab to ensure proper functionality of the solution. If you want to disable the Communication tab for your agents, you must hide it instead of removing it. To hide a tab, add the **mode** option with the value set to **HIDDEN** on either the NexusCommunication section or the NexusConversation section.

## Configure the Universal Contact Server application

Contact your Genesys representative to add the following configuration options for Universal Contact Service:

### Enable Nexus UX

Section	Option name	Value
[index]	enabled	true
[index.contact]	enabled	true
	storage-path	

## Update the Environment tenant

Contact your Genesys representative to add the following option for the Environment tenant:

Section	Option name	Value
[nexus]	url	\$nexusURL

## Create transactions

Use Agent Setup to create the following transactions in the Environment tenant:

- List 'NexusEndpoints'
- List 'NexusServices'

## Create scripts

Contact your Genesys representative to create the following scripts in the Environment tenant:

### asynchold queue

Property	Value
Name	asynchold
Type	Interaction Queue
Tenant	Environment
State enabled	checked

### asynchold View script

#### General Tab

Property	Value
Name	asynchold/scheduled_view
Type	Interaction Queue View
Tenant	Environment
State enabled	checked

#### Annex Tab

Section	Property	Value
[Namespace]	Name	scheduled_view
[View]	Condition	
	Order	
	Queue	asynchold
	scheduling-mode	scheduled-and-unscheduled
	freeze-interval	30

### asynchold application script

#### General Tab

Property	Value
Name	asynchold.ER
Type	Enhanced Routing
Tenant	Environment
State enabled	checked

---

**Annex Tab**

Section	Property	Value
[Application]	url	\$nexusURL/scxml/redirect_queue.scxml

asynchold Submitter script

**General Tab**

Property	Value
Name	asynchold.IS
Type	Interaction Submitter
Tenant	Environment
State enabled	checked

**Annex Tab**

Section	Property	Value
[Submitter]	View	asynchold/scheduled_view
	Strategy	asynchold.ER

undelivered queue

Property	Value
Name	undelivered
Type	Interaction Queue
Tenant	Environment
State enabled	checked

undelivered View script

**General Tab**

Property	Value
Name	undelivered/scheduled_view
Type	Interaction Queue View
Tenant	Environment
State enabled	checked

**Annex Tab**

---

Section	Property	Value
[Namespace]	Name	scheduled_view
[View]	Condition	
	Order	
	Queue	undelivered
	scheduling-mode	scheduled-and-unscheduled

### undelivered application script

#### General Tab

Property	Value
Name	undelivered.ER
Type	Enhanced Routing
Tenant	Environment
State enabled	checked

#### Annex Tab

Section	Property	Value
[Application]	url	\$nexusURL/scxml/undelivered.scxml

### undelivered Submitter script

#### General Tab

Property	Value
Name	undelivered.IS
Type	Interaction Submitter
Tenant	Environment
State enabled	checked

#### Annex Tab

Section	Property	Value
[Submitter]	View	undelivered/scheduled_view
	Strategy	undelivered.ER

## Create an SMS interaction subtype

Contact your Genesys representative to create the following Business Attribute Value in the "Interaction Subtype" Business Attribute:

Name	Display name	Description
SMS	SMS	The SMS text message. This interaction subtype is only required if you are using SMS.

# Enable a tenant for Digital Channels

## Contents

- [1 Prerequisites](#)
- [2 Get contact center ID from GWS](#)
- [3 Add GWS to the list of available GWS services for Nexus](#)
  - [3.1 Create the authentication client](#)
  - [3.2 Verify client](#)
  - [3.3 Add GWS to the nex\\_gapis table for Nexus](#)
- [4 Provision Digital Channels in GWS](#)
- [5 Get the authentication token](#)
- [6 Provision the Universal Contact Service connection](#)
- [7 Provision the tenant in Digital Channels](#)
- [8 Enable routing using Designer Applications](#)

Learn how to enable your tenant for Digital Channels.

### **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.

#### **Important**

“Nexus” is the simplified name we use for the Digital Channels application and nodes, so you’ll see that name referenced throughout this document.

Complete the steps on this page to provision your tenant and set up Digital Channels to work with Web Services and Applications (GWS).

## Prerequisites

Review the **Prerequisites** table and make sure you have all the listed information before you get started. The values in this table are referenced later by the name in the Variable column.

Prerequisites

Parameter	Variable	Type	Example	Notes
GWS URL	\$gwsURL	HTTPS URL string	https://gws.mydomain.com	Host of the Web Services and Applications load balancer.
Auth URL	\$authURL	HTTPS URL string	https://auth.mydomain.com	Host of the Auth Services load balancer.
GWS admin username	\$gwsAdminName	string	ops	The GWS ops credentials. See Provision Genesys Web Services and Applications.
GWS admin password	\$gwsAdminPass	string	ops	The GWS ops credentials. See Provision Genesys Web Services and Applications.
Tenant User	\$tenantUser	string	nexus	The Person username from your tenant configuration. See Create a Person object.
Tenant User Password	\$tenantUserPassword	string	nexus_password	The Person password from your tenant configuration. See Create a Person object.
Contact Center ID	\$ccld	string	578ec98e-f07c-46ad-9675-f36c22f3ba9f	The contact center ID provisioned in Web Services and Applications. If you don't have this ID, see Get contact center ID from GWS below.
Tenant Name	\$tenantName	string	premise_tenant	The tenant name you used as in the Enable Nexus UX step.
Digital Channels FQDN URL	\$nexusURL	HTTPS URL string	https://nexus.mydomain.com	The fully qualified domain



## Enable a tenant for Digital Channels

---

Parameter	Variable	Type	Example	Notes
				name (FQDN) for Digital Channels.
UCS URL	<code>\$ucsURL</code>	HTTPS URL string	<code>https://ucs.mydomain.com</code>	A URL pointing on the Universal Contact Service (UCS) entry point.

### Important

Use a REST client or curl utility to make the requests explained on this page. Make sure to substitute the variables - prefixed with '\$' - with their values.

## Get contact center ID from GWS

If you don't have the contact center ID (as specified in the **Prerequisites** table above), you can get it with the following request:

```
curl --user $gwsAdminName:$gwsAdminPass --request GET '$authURL/environment/v3/contact-centers'
```

The expected response:

```
{
  "data": {
    "contactCenters": [
      {
        "id": "",
        "environmentId": "",
        "domains": [
        ],
        "auth": "configServer"
      }
    ]
  }
}
```

## Add GWS to the list of available GWS services for Nexus

Complete the steps in this section to add GWS to the list of available GWS services for Nexus.

### Create the authentication client

To create the authentication client, send a POST request to GWS that includes a body parameter called **data** in JSON format. Give **data** the following properties:

Property	Value
clientType	CONFIDENTIAL
internalClient	true
authorizedGrantTypes	refresh_token, password, client_credentials, authorization_code
redirectURIs	\$nexusURL

## Enable a tenant for Digital Channels

---

Property	Value
authorities	ROLE_INTERNAL_CLIENT
scope	*
description	nexus_client
name	nexus_client
client_id	nexus_client
client_secret	- randomly generated and saved for further use
accessTokenExpirationTimeout	43200
refreshTokenExpirationTimeout	2592000

### Sample request

```
curl --user $gwsAdminName:$gwsAdminPass --request POST '$authURL/auth/v3/ops/clients' \
--header 'Content-Type: application/json' \
--data '{"data": {
  "internalClient": true,
  "name": "nexus_client",
  "clientType": "CONFIDENTIAL",
  "client_id": "nexus_client",
  "client_secret": "",
  "authorities": ["ROLE_INTERNAL_CLIENT"],
  "scope": ["*"],
  "authorizedGrantTypes": ["client_credentials", "authorization_code", "refresh_token", "implicit", "password"],
  "redirectURIs": ["$nexusURL"],
  "accessTokenExpirationTimeout": 43200,
  "refreshTokenExpirationTimeout": 2592000
}
}'
```

Enable a tenant for Digital Channels

---

Expected response

The expected response is **200 OK**.

### Verify client

To verify that authentication was successful, send a POST request to GWS:

```
curl --user nexus_client: \
--request POST '$authURL/auth/v3/oauth/token?grant_type=client_credentials>ope=&client_id=nexus_client&client_secret='
```

This is the response:

```
{"access_token":"","token_type":"bearer","expires_in":43199,"scope":"*"}
```

### Add GWS to the nex\_gapis table for Nexus

Execute the following query in the PostgreSQL command line interface:

```
INSERT INTO nex_gapis (url, clientid, apikey, clientsecret, created, region, authinturl, authexturl) VALUES ('$gwsURL', 'nexus_client', 'NA', '', now(), NULL, '$authURL', '$authURL');
```

## Provision Digital Channels in GWS

To provision Digital Channels in GWS, send a POST request to GWS:

```
curl --user $gwsAdminName:$gwsAdminPass --request POST '$authURL/environment/v3/contact-centers/$ccId/settings' \
--header 'Content-Type: application/json' \
--data '{
  "data":
  {
    "name": "chatservice-uri",
    "location": "/",
    "value": "$nexusURL",
    "shared": false
  }
}
```

## Get the authentication token

To get the authentication token, send a POST request to GWS:

```
curl --user nexus_client: --request POST '$authURL/auth/v3/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'username=$ccId\$tenantUser' \
--data-urlencode 'client_id=nexus_client' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'password=$tenantUserPassword'
```

The expected response:

```
{"access_token": "", "token_type": "bearer", "refresh_token": "", "expires_in": 43199, "scope": "*"}
```

As an output of this step, you will have the GWS access token:

Parameter	Variable	Type	Example	Notes
Nexus API Key	\$outGWSAccessToken	string	9b7682b7-cbce-422f-90bb-ecda85e61695	The access_token from the response example.

## Provision the Universal Contact Service connection

You must complete this step if you are using Universal Contact Service in your environment.

Make the following POST request to create the PlatformUCS service:

---

## Enable a tenant for Digital Channels

---

```
curl -X POST \  
  $nexusURL/nexus/v3/provisioning/services/$ccid/PlatformUCS \  
  -H 'Content-Type: application/json' \  
  -H 'x-api-key: $apiKey' \  
  -H 'x-ccid: $ccid' \  
  -d '{  
    "url" : "$ucsURL",  
    "secret": {},  
    "data" : {}  
  }'
```

## Provision the tenant in Digital Channels

To provision the tenant in Digital Channels, make the following POST request:

```
curl --request POST '$nexusURL/nexus/v3/provisioning/tenants' \  
--header 'Authorization: Bearer $outGWSAccessToken' \  
--header 'Content-Type: application/json' \  
--data '{  
  "id": "$ccId",  
  "name": "$tenantName",  
  "type": "PureEngage",  
  "backendurl": "$gwsURL",  
  "username": "$tenantUser",  
  "token": "$tenantUserPassword",  
  "genesystemenantid": 1  
}'
```

The expected response:

```
{  
  "status": {  
    "code": 0  
  },  
  "data": {  
    "xapikey": "$nexusApiKey"  
  },  
  "operationId": "ec90f3d2-f4b5-47fd-9004-2c309398ab38"  
}
```

As an output of this step, you will have the specific tenant API key:

Parameter	Variable	Type	Example	Notes
Nexus API Key	\$nexusApiKey	UUID string	9b7682b7-cbce-422f-9bb0-ecda85e61695	The <b>xapikey</b> from the response example. For creating chat.

## Enable routing using Designer Applications

Use Agent Setup to add the following values to the "DesignerEnv" transaction in the Environment tenant. If the transaction doesn't exist, create it with a type of List and "DesignerEnv" as Name and Alias.

---



## Enable a tenant for Digital Channels

---

Property	Value
baseurl	/nexus/v1
password	
url	/nexus/v3

# Provision API keys

## Contents

- [1 Prerequisites](#)
- [2 Tenant API keys](#)
- [3 Cluster API keys](#)
- [4 Designer API key](#)

Learn how to provision API keys for Digital Channels.

**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.

**Important**

“Nexus” is the simplified name we use for the Digital Channels application and nodes, so you’ll see that name referenced throughout this document.

Complete the steps on this page to provision API keys for Digital Channels.

## Prerequisites

Review the **Prerequisites** table and make sure you have all the listed information before you get started. The values in this table are referenced later by the name in the Variable column.

Prerequisites

Parameter	Variable	Type	Example	Notes
Auth URL	\$authURL	HTTPS URL string	https://auth.mydomain.com	Host of the Auth Services load balancer.
Contact Center ID	\$ccId	string	578ec98e-f07c-46ad-9675-f36c22f3ba9f	The contact center ID provisioned in Web Services and Applications. If you don't have this ID, see Get contact center ID from GWS below.
GWS admin username	\$gwsAdminName	string	ops	The GWS ops credentials. See Provision Genesys Web Services and Applications.
GWS admin password	\$gwsAdminPass	string	ops	The GWS ops credentials. See Provision Genesys Web Services and Applications.
Digital Channels FQDN URL	\$nexusURL	HTTPS URL string	https://nexus.mydomain.com	The fully qualified domain name (FQDN) for Digital Channels.
Tenant Name	\$tenantName	string	premise_tenant	The tenant name you used as in the Enable Nexus UX step.
Tenant User	\$tenantUser	string	nexus	The Person username from your tenant configuration. See Create a Person object.
Tenant User Password	\$tenantUserPassword	string	nexus_password	The Person password from your tenant configuration. See Create a Person object.

## Tenant API keys

### Important

Use a REST client or curl utility to make the requests explained on this page. Make sure to substitute the variables - prefixed with '\$' - with their values.

Each API key you provision has a tenant or a set of tenants to which it is provisioned. The API key is also associated with a set of permissions. In the example below, we create a tenant API key with consumer chat API key permissions. Complete the steps below to provision an API key. First, get the authentication token by sending a POST request to GWS:

```
curl --user nexus_client: --request POST '$authURL/auth/v3/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'username=$ccId\\$tenantUser' \
--data-urlencode 'client_id=nexus_client' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'password=$tenantUserPassword'
```

The expected response:

```
{"access_token": "", "token_type": "bearer", "refresh_token": "", "expires_in": 43199, "scope": ""}
```

As an output of this step, you will have the GWS access token for the tenant:

Parameter	Variable	Type	Example	Notes
Tenant GWS access token	\$outGWSAccessToken	string	9b7682b7-cbce-422f-9bbb-ecda85e61695	The access_token from the response example.

To provision the API key, send a POST request to Digital Channels:

```
curl --request POST '$nexusURL/nexus/v3/apikeys' \
--header 'Authorization: $outGWSAccessToken' \
--header 'Content-Type: application/json' \
--data '{
  "enabled": true,
  "tenant": "$ccId",
  "name": "API key for $tenantName",
  "permissions": [
    "nexus:consumer:chat"
  ]
}'
```

The expected response:

```
{
  "id": "",
```

## Provision API keys

---

```
"tenant": "$ccId",
"name": "API key for $tenantName",
"permissions": [
  "nexus:consumer:chat"
],
"enabled": true,
"created": "",
"revoked": false,
"expires": false,
"createdby": "",
"capacity": 120,
"frequency": 60
}
```

As an output of this step, you will have the tenant API key:

Parameter	Variable	Type	Example	Notes
Tenant API Key	\$apikey	UUID string	9b7682b7-cbce-422f-9b0b-ecda85e61695	The ID from the response example

## Cluster API keys

Complete the following steps to provision a cluster (multi-tenant) API key.

First, get the cluster authentication token by sending a POST request to GWS:

```
curl --user nexus_client: --request POST '$authURL/auth/v3/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'username=$gwsAdminName' \
--data-urlencode 'client_id=nexus_client' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'password=$gwsAdminPass'
```

The expected response:

```
{"access_token": "", "token_type": "bearer", "refresh_token": "", "expires_in": 43199, "scope": "*"}
```

As an output of this step, you will have the cluster GWS access token:

Parameter	Variable	Type	Example	Notes
Cluster GWS access token	\$outClusterGWSAccessToken	UUID string	9b7682b7-cbce-422f-9b0b-ecda85e61695	The access_token from the response example.

To provision the API key, send a POST request to Digital Channels:

```
curl --request POST '$nexusURL/nexus/v3/apikeys' \
--header 'Authorization: $outClusterGWSAccessToken' \
--header 'Content-Type: application/json' \
--data '{
  "enabled": true,
  "tenant": "*",
  "name": "Cluster API key",
  "permissions": [
    "nexus:cluster:*"
  ]
}
```

```
}'
```

The expected response:

```
{
  "id": "",
  "tenant": "*",
  "name": "Cluster API key",
  "permissions": [
    "nexus:cluster:*"
  ],
  "enabled": true,
  "created": "",
  "revoked": false,
  "expires": false,
  "createdby": "",
  "capacity": 120,
  "frequency": 60
}
```

As an output of this step, you will have the cluster API key:

Parameter	Variable	Type	Example	Notes
Cluster API Key	<code>\$clusterApikey</code>	UUID string	9b7682b7-cbce-422f-91ef-ecda85e61695	Obtained from the response example

## Designer API key

First, follow the instructions above to create a dedicated Designer API key. Set permissions as:

```
"permissions": [ "nexus:designer:chat" ]
```

Genesys recommends creating a separate key from the consumer chat API key so that you can update your consumer API key without impacting Designer applications.

```
curl --request POST '$nexusURL/nexus/v3/apikeys' \
--header 'Authorization: $outGWSAccessToken' \
--header 'Content-Type: application/json' \
--data '{
  "enabled": true,
  "tenant": "$ccId",
  "name": "Designer API key",
  "permissions": [
    "nexus:designer:chat"
  ]
}'
```

The expected response:

```
{
  "id": "",
  "tenant": "$ccId",
  "name": "Designer API key",
  "permissions": [
```

## Provision API keys

---

```
    "nexus:designer:chat"  
  ],  
  "enabled": true,  
  "created": "",  
  "revoked": false,  
  "expires": false,  
  "createdby": "",  
  "capacity": 120,  
  "frequency": 60  
}
```

As an output of this step, you will have the specific tenant API key:

Parameter	Variable	Type	Example	Notes
Designer API Key	\$designerApikey	UUID string	9b7682b7-cbce-422f-91e1-ecda85e61695	Taken from the response example



# Provision SMS

## Contents

- **1 Set up Digital Channels to use Genesys Messaging Aggregation**
  - 1.1 Prerequisites
  - 1.2 Request a Genesys Cloud CX organization
  - 1.3 Create the Digital Channels integration in Genesys Cloud CX
  - 1.4 Create a Digital Channels API key for Genesys Cloud CX
  - 1.5 Create Digital Channels services definitions
  - 1.6 Manage phone numbers and email domains
- **2 Set up Digital Channels to use a custom gateway**
  - 2.1 Prerequisites
  - 2.2 Create Digital Channels services definitions
  - 2.3 Manage phone numbers and email domains
- **3 Provision phone numbers in Digital Channels**
  - 3.1 Add custom HTTP headers
- **4 Provision email domains in Digital Channels**

- Administrator
- Developer

Learn how to enable SMS media and outbound email campaigns for Digital Channels.

**Related documentation:**

- 
- 

**Early Adopter Program**

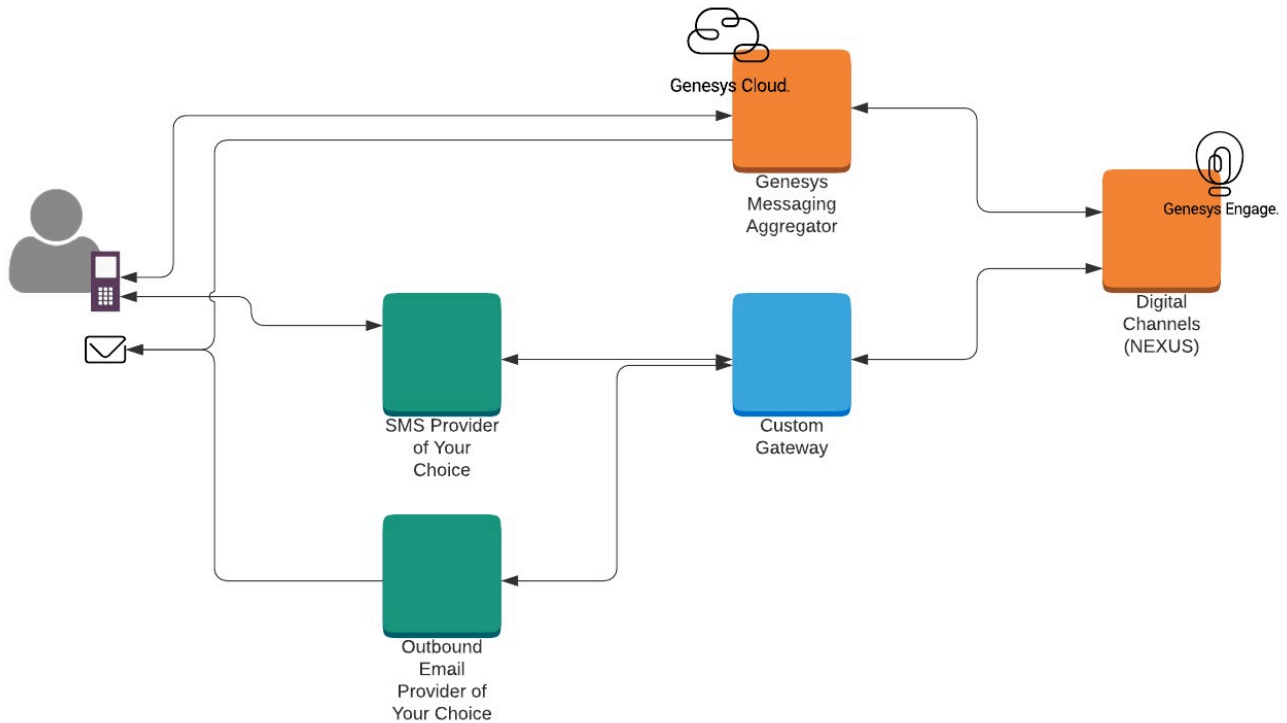
Genesys Multicloud CX private edition is being released to pre-approved customers as part of the Early Adopter Program. Please note that the documentation and the product are subject to change. For more details about the program, please contact your Genesys representative.

**Important**

“Nexus” is the simplified name we use for the Digital Channels application and nodes, so you’ll see that name referenced throughout this document.

Digital Channels can handle SMS media and outbound email campaigns using either Genesys Messaging Aggregation from Genesys Cloud CX or a third-party aggregator through a custom gateway.

See the network diagram below for details.



After completing the setup steps on this page, you will be able to:

- Receive SMS messages from your consumers on your corporate numbers and allow a Designer application or agent to respond to them.
- Use outgoing SMS and email messages in your outbound campaigns (through CX Contact).

To get started, complete the configuration steps for **one** of the following scenarios:

- Set up Digital Channels to use Genesys Messaging Aggregation  
OR
- Set up Digital Channels to use a custom gateway

## Set up Digital Channels to use Genesys Messaging Aggregation

Complete the steps in this section to set up Digital Channels to use Genesys Messaging Aggregation as its SMS gateway. Here's a breakdown of the tasks:

1. Review the prerequisites table and make sure you have the information listed.
2. Contact your Genesys representative to create a Genesys Cloud CX organization and get administrator user credentials. Your Genesys representative also must add the Genesys Messaging Aggregation product to your organization.
3. Create the Digital Channels integration in Genesys Cloud CX. This will give you a `clientId` and `clientSecret` to authenticate API calls with the Digital Channels provisioning API.

4. Create a Digital Channels API key for Genesys Cloud CX.
5. Use the Digital Channels provisioning API to create Digital Channels services definition.
6. Manage phone numbers and email domains.

## Prerequisites

Review the **Prerequisites** table and make sure you have all the listed information before you get started. The values in this table are referenced later by the name in the Variable column.

Prerequisites

Parameter	Variable	Type	Example	Notes
GWS tenant Contact Center ID	\$ccid	UUID string	45acae06-6b7c-4f97-9c76-471c4b25bf71	This value comes from your Web Services and Applications deployment.

## Request a Genesys Cloud CX organization

Contact your Genesys representative to create a Genesys Cloud CX organization and get administrator user credentials. Your Genesys representative also must add the Genesys Messaging Aggregation product to your organization.

Make sure your Genesys representative provides you with the details in the **Genesys Cloud CX information** table.

Genesys Cloud CX information

Parameter	Variable	Type	Example	Notes
Genesys Cloud CX Organization ID	\$orgId	UUID string	47d8329d-1c28-4c86-9374-5596bd0fee15	Your Genesys Cloud CX organization ID.
Genesys Cloud CX Organization admin user credentials	\$orgUsername \$orgPassword	string	admin-user / admin-password	The username and password for an account with administrative permissions for this organization.
Genesys Cloud CX Login URL	\$gcLoginURL	HTTPS URL string	https://login.mypurecloud.com	Your Genesys Cloud CX login URL (depends on your organization region).
Genesys Cloud CX API URL	\$gcAPIURL	HTTPS URL string	https://api.mypurecloud.com	Your Genesys Cloud CX login URL (depends on your

## Provision SMS

---

Parameter	Variable	Type	Example	Notes
				organization region).

## Create the Digital Channels integration in Genesys Cloud CX

Complete the steps in this section as an administrator user in Genesys Cloud CX to create the integration client credentials that will be used by Digital Channels to access Genesys Cloud CX APIs to send and receive messages. You're going to create a new role, assign it to your admin user, and create the access credentials.

First, create the new role:

1. Navigate to `$gcLoginURL` (for example, `https://login.mypurecloud.com`) and log in to Genesys Cloud CX with your `$orgUsername/$orgPassword`.
2. Go to **Admin**.
3. Under **People and Permissions**, click **Roles/Permissions**.
4. Click **Add Role** and give it a name. For example, `Nexus Messaging`.
5. Under **Permissions**, search for **messaging** and select **messaging > All Permissions** and **messagingProvisioning > All Permissions**. Save your changes.

Next, assign the role to your administrator user:

1. Click **Admin**.
2. Under **People and Permissions**, click **People**.
3. Search for your admin user.
4. Under **Roles**, switch the view to **All** and search for the name of your new role (`Nexus Messaging`). Click to enable the role and then save your changes.
5. Log out and log in again to enable the permissions.

Now create access credentials for the Digital Channels integration.

1. Click **Admin**.
2. Under **Integrations**, click **OAuth**.
3. Click **Add Client**.
4. Under **Client Details**, set **App Name** to `Nexus Messaging Integration` and select the Client Credentials **Grant Type**.
5. Click **Roles** and assign the `Nexus Messaging` role. Save your changes.
6. Go back to **Client Details** and copy the values for **clientId** and **clientSecret**.

As the output of this step, you will have the access credentials:

Parameter	Variable	Type	Example
Nexus Integration client ID	<code>\$clientId</code>	UUID string	4da40a9de-b113-4024-8ba9-c9dd89c91f67
Nexus Integration client secret	<code>\$clientSecret</code>	string	aKSXEgLO57cm6FqxD4hrjkcW- iuWiXhd0uF0WocZUm2



## Create a Digital Channels API key for Genesys Cloud CX

To create an API key that will be used by Genesys Cloud CX to send requests to Digital Channels, follow the steps in Provision API keys. Make sure to use the following parameters:

```
"tenant": "*"
"name": "Portico Cluster API Key"
"permissions" : ["nexus:cluster:*"]
```

As an output of this step, you will have the API key:

Parameter	Variable	Type	Example
Messaging Cluster API Key	\$apikey	UUID string	9b7682b7-cbce-422f-9bbb-ecda85e61695

## Create Digital Channels services definitions

In this step you will enable Digital Channels to use Genesys Messaging Aggregation as the SMS and outbound email provider. You must configure the following services in Digital Channels within your tenant:

- PurecloudIDP - Integrates the Genesys Multicloud CX tenant to the Genesys Cloud CX organization.
- SMS - Enables SMS media for the tenant and selects the provider.
- PorticoSMS - Enables the SMS service through Genesys Messaging Aggregation.
- PorticoEmail - Enables the outbound email service through Genesys Messaging Aggregation.

### Important

Use a REST client or curl utility to provision the following services in Digital Channels using the provisioning API. Make sure to substitute the variables - prefixed with '\$' - with their values. Note: You must create these services once per tenant.

### Create the **PurecloudIDP** service:

```
curl -X POST \
  $nexusURL/nexus/v3/provisioning/services/$ccid/PurecloudIDP \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: $apiKey' \
  -H 'x-ccid: $ccid' \
  -d '{
    "url" : "$gcLoginURL",
    "secret": {"clientId": "$clientId", "clientSecret": "$clientSecret"},
    "data" : {}
  }'
```

### Create the **SMS** service:

```
curl -X POST \
```

```
$nexusURL/nexus/v3/provisioning/services/$ccid/SMS \  
-H 'Content-Type: application/json' \  
-H 'x-api-key: $apiKey' \  
-H 'x-ccid: $ccid' \  
-d '{  
  "url" : "N/A",  
  "secret": {},  
  "data" : {"tokenProvider": "Purecloud"}  
'
```

### Create the **PorticoSMS** service:

```
curl -X POST \  
$nexusURL/nexus/v3/provisioning/services/$ccid/PorticoSMS \  
-H 'Content-Type: application/json' \  
-H 'x-api-key: $apiKey' \  
-H 'x-ccid: $ccid' \  
-d '{  
  "url" : "$gcAPIURL",  
  "secret": {},  
  "data": {}  
'
```

(Optional) - Create the **PorticoEmail** service if you use CX Contact email campaigns.

```
curl -X POST \  
$nexusURL/nexus/v3/provisioning/services/$ccid/PorticoEmail \  
-H 'Content-Type: application/json' \  
-H 'x-api-key: $apiKey' \  
-H 'x-ccid: $ccid' \  
-d '{  
  "url" : "$gcAPIURL",  
  "secret": {},  
  "data": {}  
'
```

## Manage phone numbers and email domains

You can purchase new SMS numbers or re-use your existing numbers or email domains.

If you need to register an existing (Bring-Your-Own-Number) number, email domain or new short code, contact your Genesys representative to complete this step. Otherwise, follow the steps below to use the Genesys Messaging Aggregation API from Genesys Cloud CX to purchase and register toll free numbers from the pool of available numbers. **Note:** Each purchased number will incur additional costs to your account.

### Retrieve the Genesys Cloud CX token

Any Genesys Cloud CX operation has to include a security token that remains valid for a configured amount of time. When the token expires, you must retrieve it again in order to send new requests.

To retrieve your token, use Basic Authentication where the username is **\$clientId** and the password is **\$clientSecret**.

```
curl -X POST \  
$gcLoginURL/oauth/token \  

```

```
-H 'Authorization: Basic YOUR_BASIC_AUTHENTICATION_SECRETS' \  
-H 'Content-Type: application/x-www-form-urlencoded' \  
-d 'grant_type=client_credentials'
```

The response contains:

- **access\_token** - You must include this in all subsequent requests.
- **expires\_in** - Indicates how long the token is valid.
- **token\_type** - Indicates how this token needs to be included in subsequent requests (bearer token).

List purchased and registered numbers

Run the following command to list your purchased and registered numbers:

```
curl -X GET \  
$gcAPIURL/api/v2/messaging/sms/provisioning/tollfreenumbers/ \  
-H 'Authorization: Bearer $access_token'
```

Search for available toll free number

To order a new number, first search for available numbers and then select one of the options.

```
curl -X GET \  
$gcAPIURL/api/v2/messaging/sms/provisioning/tollfreenumbers/available \  
-H 'Authorization: Bearer $access_token'
```

The response contains a few currently available numbers - choose the one you like. We'll use the variable **\$tfn** to represent this number.

Order a toll free number

When you send this request, use the **comment** and **emailAddress** fields to help Genesys Customer Care quickly identify the best contact person if there's an issue with the SMS service. For example, you can include your organization name in the **comment** field.

```
curl -X POST \  
$gcAPIURL/api/v2/messaging/sms/provisioning/tollfreenumbers/ \  
-H 'Authorization: Bearer $access_token' \  
-H 'Content-Type: application/json' \  
-d '{  
  {  
    "tollfreeNumber": "$tfn",  
    "comment": "Nexus Premise ACME Corp",  
    "moUrl": "$nexusURL/nexus/v3/sms/message",  
    "drUrl": "$nexusURL/nexus/v3/sms/receipt",  
    "webhookUsername": "$ccid",  
    "webhookPassword": "$GMAKey",  
    "emailAddress": "sms.admin@acme.test.com"  
  }  
}'
```

Finally, provision your new numbers in Digital Channels.

## Email domains

Contact your Genesys representative to complete this step.

## Set up Digital Channels to use a custom gateway

Complete the steps in this section to set up Digital Channels to use a custom SMS gateway or email provider. Here's a breakdown of the tasks:

1. Review the prerequisites table and make sure you have all the information listed.
2. Use the Digital Channels provisioning API to Create Digital Channels services definitions.
3. Manage phone numbers and email domains.

## Prerequisites

Review the **Prerequisites** table and make sure you have all the listed information before you get started. The values in this table are referenced later by the name in the Variable column.

Prerequisites

Parameter	Variable	Type	Example	Notes
Company's phone number	\$asyncPhoneNumber	string	16504661149	
Company's email domain	\$asyncEmailDomain	string	company.com	
Third-party Messaging Webhook URL	\$asyncWebhookURL	string	https://genesys-webhook.company.com	The FQDN of the third-party service implementing the Third-Party Messaging Webhook.
Third-Party Messaging API secret key	\$asyncAPISignatureKey	string	Secret	The key used by the third-party service to calculate the signature for calls to the Third-Party Messaging API.
Third-Party Messaging Webhook secret key	\$asyncWebhookSignatureKey	string	Secret	The key used by Digital Channels to calculate the signature for calls to the third-party service through the webhook.

## Create Digital Channels services definitions

In this step you will enable Digital Channels to use your custom gateway for the SMS/Email provider of your choice. You must configure the following services in Digital Channels within your tenant:

- Async - Integrates the Genesys Multicloud CX tenant to the custom gateway used to communicate with the SMS/Email provider of your choice.
- SMS - Enables SMS media for the tenant and selects the Async provider.
- Email - Enables outbound email service through the Async provider.

### Important

Use a REST client or curl utility to provision the following services in Digital Channels using the provisioning API. Make sure to substitute the variables - prefixed with '\$' - with their values. You must create these services once per tenant.

Use a REST client or curl utility to provision the following services in Digital Channels using the provisioning API. Make sure to substitute the variables - prefixed with '\$' - with their values. You must create these services once per tenant. Create the **Async** service:

```
curl -X POST \
  $nexusURL/nexus/v3/provisioning/services/$ccid/Async \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: $apiKey' \
  -H 'x-ccid: $ccid' \
  -d '{
    "data": {
      "channels": [
        {
          "channelId": "$asyncPhoneNumber",
          "webhook": { "url": "$asyncWebhookURL" }
        },
        {
          "channelId": "$asyncEmailDomain",
          "webhook": { "url": "$asyncWebhookURL" }
        }
      ]
    },
    "secret": {
      "channels": [
        {
          "channelId": "$asyncPhoneNumber",
          "webhook": { "secret": "$asyncWebhookSignatureKey" },
          "api": { "secret": "$asyncAPISignatureKey" }
        },
        {
          "channelId": "$asyncEmailDomain",
          "webhook": { "secret": "$asyncWebhookSignatureKey" },
          "api": { "secret": "$asyncAPISignatureKey" }
        }
      ]
    }
  }
```

```
    }  
  }'
```

### Create the **SMS** service:

```
curl -X POST \  
  $nexusURL/nexus/v3/provisioning/services/$ccid/SMS \  
  -H 'Content-Type: application/json' \  
  -H 'x-api-key: $apiKey' \  
  -H 'x-ccid: $ccid' \  
  -d '{  
    "url" : "N/A",  
    "data" : { "provider": "Async" },  
    "secret": {}  
  }'
```

### Create the **Email** service:

```
curl -X POST \  
  $nexusURL/nexus/v3/provisioning/services/$ccid/Email \  
  -H 'Content-Type: application/json' \  
  -H 'x-api-key: $apiKey' \  
  -H 'x-ccid: $ccid' \  
  -d '{  
    "url" : "N/A",  
    "data" : { "provider": "Async" },  
    "secret": {}  
  }'
```

## Manage phone numbers and email domains

Provision phone numbers and email domains (for integrations with CX Contact for email campaigns) in Digital Channels. **Note:** An SMS or email channel must have a "provider" property equal to "Async" to use the Third-Party Messaging API implementation.

## Provision phone numbers in Digital Channels

Complete the following steps for each phone number you want to use to send and receive SMS messages.

1. Log in to Designer and create an application to route SMS interactions. Create a chat endpoint and assign it to your application, making note of the name you use. We'll use the variable **\$designerEndpointName** to represent this value.
2. Log in to Platform Administration and go to **Environment > Transactions > NexusServices > Options**.
3. Create a section that starts with 'chat.' and includes some text that represents the purpose of number. For example, chat.SMS\_Main\_Corporate or chat.SMS\_CustomerSupport. Note: The 'chat.' prefix is required and the rest of the value can be made up of letters and underscore or dashes, but not spaces.
4. In this new section, create the following options:
  - channelId = **\$tfn**
  - channelType = sms

- endpoint = chat.**\$designerEndpointName**
- interactionSubtype = SMS
- interactionType = Inbound
- media = chat
- cxcOnly = false  
**Note:** Set `cxcOnly` to **true** for CX Contact integrations. If **true**, incoming SMS messages are not processed as chat messages and are not delivered to an agent. The default value is **false**.
- provider = Async (Only set this option if you use a custom gateway as the SMS aggregator.)

### Add custom HTTP headers

You can add custom HTTP headers with static values to the webhooks sent by Digital Channels to the third-party messaging aggregator. On the transaction **NexusServices > [your async provider]**, add a property that starts with the "header:" prefix and set it to your static value. For example, `header:custom-header-for-async-1 = 12345`. The webhook from Digital Channels will include this property name and value in the header:

```
HTTP
custom-header-for-async-1: 12345
X-Hub-Signature:
X-B3-TraceId:
Content-Type: application/json
```

```
{
  "messages": [
    {
      ... RichMedia message ...
    }
  ]
}
```

## Provision email domains in Digital Channels

Complete the steps in this section if you are integrating with CX Contact and plan to use email campaigns. These steps explain how to choose an email domain that you control (you should be able to update DNS record sets for this domain) and want to use in email campaigns as the "sent from" address.

If you are using Genesys Cloud CX as the provider, contact your Genesys representative to have them provision an email domain in Genesys Cloud CX for your organization. Once completed, you will receive a set of secrets you must use to update your domains records. After this update, contact your Genesys representative to validate the secrets and confirm domain ownership. Now you can provision an email service channel.

Complete the following for each domain only after your domain records have been updated, validated, and provisioned in Genesys Cloud CX.

1. In Agent Setup, navigate to the "NexusServices" transaction you created previously.
2. Create a section that starts with 'cxc.' and includes some text that represents the domain. For example,



`cxc.Corporate_Promotions`. Note: The 'cxc.' prefix is required and the rest of the value can be made up of letters and underscore or dashes, but not spaces.

3. In this new section, create the following options:

- `channelId = $emailDomain`
- `channelType = email`
- `provider = Async` (Only set this option if you use a custom gateway as the email aggregator.)

# Observability in Digital Channels

## Contents

- **1 Monitoring**
  - 1.1 Enable monitoring
  - 1.2 Configure metrics
- **2 Alerting**
  - 2.1 Configure alerts
- **3 Logging**

Learn about the logs, metrics, and alerts you should monitor for Digital Channels.

**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 Digital Channels metrics, see:

- Digital Channels metrics

See also System metrics.

## Enable monitoring

Digital Channels uses ServiceMonitor custom resource definitions (CRDs) and it also supports custom annotations for pods. To enable monitoring for Digital Channels, set the **monitoring.enabled** Helm chart parameter to `true`. To set custom annotations, use the **podAnnotations** Helm chart parameter. See Override Helm chart values for more information about these parameters.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Digital Channels	Both — ServiceMonitor and annotations	4004	nexus.nexus.svc.cluster.local/metrics	15 seconds

## Configure metrics

The metrics that are exposed by Digital Channels 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 some of the metrics Digital Channels exposes. You can also query Prometheus directly or via a dashboard to see all the metrics available from Digital Channels.

## Alerting

Private edition services define a number of alerts based on Prometheus metrics thresholds.

### Important

While you can use general third-party functionality to create rules to trigger alerts based on metrics values you specify, private edition does not enable you to create custom alerts, and Genesys does not provide support for custom alerting.

For descriptions of available Digital Channels alerts, see:

- Digital Channels alerts

## Configure alerts

Private edition services define a number of alerts by default (for Digital Channels, see the pages linked to above). No further configuration is required.

The alerts are defined as **PrometheusRule** objects in a **prometheus-rule.yaml** file in the Helm charts. As described above, Digital Channels does not support customizing the alerts or defining additional **PrometheusRule** objects to create alerts based on the service-provided metrics.

## Logging

Digital Channels outputs logs to stdout. You can extract these logs using log collectors such as logstash and Elasticsearch. For more information about logging in private edition, see the logging

overview in the Operations guide.

# Digital Channels metrics and alerts

Find the metrics Digital Channels exposes and the alerts defined for Digital Channels.

## **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
Digital Channels	Both — ServiceMonitor and annotations	4004	nexus.nexus.svc.cluster.local/metrics	15 seconds

See details about:

- Digital Channels metrics
- Digital Channels alerts

## Metrics

Digital Channels exposes many Genesys-defined metrics. You can query Prometheus directly to see all the available metrics. The metrics documented on this page are likely to be particularly useful. Genesys does not commit to maintain other currently available Digital Channels metrics not documented on this page.

Metric and description	Metric details	Indicator of
<b>nexus_errors_total</b> The total number of requests that resulted in an error.	<b>Unit:</b> <b>Type:</b> Number <b>Label:</b> <b>Sample value:</b> 100	
<b>nexus_request_total</b> The total number of requests.	<b>Unit:</b> <b>Type:</b> Number <b>Label:</b> <b>Sample value:</b> 1000	
<b>nexus_process_resident_memory_bytes</b> The total bytes of memory Digital Channels consumed.	<b>Unit:</b> <b>Type:</b> Number <b>Label:</b> <b>Sample value:</b> 100000	
<b>nexus_redis_connections_established</b> The current number of established Redis connections.	<b>Unit:</b> <b>Type:</b> Gauge <b>Label:</b> <b>Sample value:</b> 0	
<b>nexus_redis_connections_reconnecting</b> The current number of reconnecting Redis connections.	<b>Unit:</b> <b>Type:</b> Gauge <b>Label:</b> <b>Sample value:</b> 0	
<b>nexus_redis_connections_ready</b> The current number of ready Redis	<b>Unit:</b> <b>Type:</b> Gauge	

Metric and description	Metric details	Indicator of
connections.	<b>Label:</b> <b>Sample value:</b> 1	
<b>nexus_redis_duration_until_ready</b> The duration until Redis reaches the ready state.	<b>Unit:</b> <b>Type:</b> Histogram <b>Label:</b> 'le' <b>Sample value:</b> 0, 1, 39	
<b>nexus_redis_errors_total</b> The total number of Redis connection errors.	<b>Unit:</b> <b>Type:</b> Counter <b>Label:</b> <b>Sample value:</b> 0	
<b>nexus_db_connect_total</b> The total number of all database connection requests.	<b>Unit:</b> <b>Type:</b> Counter <b>Label:</b> 'db' <b>Sample value:</b> 1252424, 1457770	
<b>nexus_db_disconnect_total</b> The total number of all database disconnection requests.	<b>Unit:</b> <b>Type:</b> Counter <b>Label:</b> 'db' <b>Sample value:</b> 1252424, 1457770	
<b>nexus_db_request_total</b> The total number of all database requests sent.	<b>Unit:</b> <b>Type:</b> Counter <b>Label:</b> 'db' <b>Sample value:</b> 4850730, 5056452	
<b>nexus_db_success_total</b> The total number of all database requests executed successfully.	<b>Unit:</b> <b>Type:</b> Counter <b>Label:</b> 'db', 'command' <b>Sample value:</b> 2307896, 2126805, 1221394, 1450355	
<b>nexus_db_errors_total</b> The total number of all database errors.	<b>Unit:</b> <b>Type:</b> Counter <b>Label:</b> 'db', 'code' <b>Sample value:</b> 131, 5, 4	
<b>nexus_db_request_duration_milliseconds</b> The database transaction duration.	<b>Unit:</b> <b>Type:</b> histogram <b>Label:</b> 'le', 'db', 'method' <b>Sample value:</b> 2290844, 2306385, 2307241, 2307894	
<b>iwd_process_cpu_user_seconds_total</b> The total user CPU time spent, in seconds.	<b>Unit:</b> <b>Type:</b> counter <b>Label:</b> <b>Sample value:</b> 1634045655571	
<b>nexus_process_cpu_system_seconds_total</b>	<b>Unit:</b>	



Metric and description	Metric details	Indicator of
The total system CPU time spent, in seconds.	<b>Type:</b> counter <b>Label:</b> <b>Sample value:</b> 1634045655571	
<b>nexus_process_cpu_seconds_total</b>	<b>Unit:</b>	
The total user and system CPU time spent, in seconds.	<b>Type:</b> counter <b>Label:</b> <b>Sample value:</b> 1634045655571	
<b>nexus_process_start_time_seconds</b>	<b>Unit:</b>	
The start time of the process since the Unix epoch, in seconds.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1633992102	
<b>nexus_process_resident_memory_bytes</b>	<b>Unit:</b>	
The resident memory size, in bytes.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_process_virtual_memory_bytes</b>	<b>Unit:</b>	
The virtual memory size, in bytes.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_process_heap_bytes</b>	<b>Unit:</b>	
The process heap size, in bytes.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_process_open_fds</b>	<b>Unit:</b>	
The number of open file descriptors.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_process_max_fds</b>	<b>Unit:</b>	
The maximum number of open file descriptors.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 197176	
<b>iwd_nodejs_eventloop_lag_seconds</b>	<b>Unit:</b>	
The Node.js event loop lag, in seconds.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_nodejs_active_handles</b>	<b>Unit:</b>	
The number of active libuv handles, grouped by handle type. Every handle type is a C++ class name.	<b>Type:</b> gauge <b>Label:</b> 'type' <b>Sample value:</b> 17, 1, 69	
<b>nexus_nodejs_active_handles_total</b>	<b>Unit:</b>	

Metric and description	Metric details	Indicator of
The total number of active libuv handles.	<b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_nodejs_active_requests</b> The number of active libuv requests, grouped by request type. Every request type is a C++ class name.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> 'type' <b>Sample value:</b> 2	
<b>nexus_nodejs_active_requests_total</b> The total number of active libuv requests.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_nodejs_heap_size_total_bytes</b> The process heap size from Node.js, in bytes.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_nodejs_heap_size_used_bytes</b> The process heap size used from Node.js, in bytes.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_nodejs_external_memory_bytes</b> The Node.js external memory size, in bytes.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> <b>Sample value:</b> 1634045655572	
<b>nexus_nodejs_heap_space_size_total_bytes</b> The process heap space size total from Node.js, in bytes.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> 'space' <b>Sample value:</b> 262144, 16777216, 130428928, 6721536	
<b>nexus_nodejs_heap_space_size_used_bytes</b> The process heap space size used from Node.js in bytes.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> 'space' <b>Sample value:</b> 32808, 1479672, 92634792, 4852384	
<b>nexus_nodejs_heap_space_size_available_bytes</b> The process heap space size available from Node.js, in bytes.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> 'space' <b>Sample value:</b> 0, 6899976, 37040456, 1542496	
<b>nexus_nodejs_version_info</b> Node.js version information.	<b>Unit:</b> <b>Type:</b> gauge <b>Label:</b> 'version', 'major', 'minor', 'patch' <b>Sample value:</b> 1	

## Alerts

The following alerts are defined for Digital Channels.

Alert	Severity	Description	Based on	Threshold
Nexus error rate	Critical	Triggered when the error rate on this pod is greater than 20% for 15 minutes.	nexus_errors_total, nexus_request_total	For 15 minutes
Memory usage is above 3000 Mb	Critical	Triggered when the memory usage on this pod is above 3000 Mb for 15 minutes.	nexus_process_resident_memory_bytes	For 15 minutes