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.

Setting up Genesys Multicloud CX Private Edition

2/7/2026

Table of Contents

Overview

Requirements

Deployment overview

Configure your environment

Deploy private edition

Upgrade

Uninstall

Troubleshooting

References

Contents

This guide provides general information covering what you need to know about setting up Genesys Multicloud CX private edition in your environment.

Related documentation:

-
-

RSS:

- For private edition

Genesys Multicloud CX private edition is a microservices-based contact center offering that adopts containerization technology for all the components.

Overview

Learn about Genesys Multicloud CX private edition, its architecture, the private edition services, and an overview on private edition's networking, and security.

- About Genesys Multicloud CX private edition
- Architecture
- Private edition services
- High availability and disaster recovery
- Networking overview
- Security overview

Requirements

Learn about prerequisites, third-party dependencies, and responsibilities between your organization and Genesys in setting up Genesys Multicloud CX private edition.

- Software requirements
- Storage requirements
- Communication ports and protocols
- Understanding responsibilities

Deployment overview

Have a quick tour of the deployment steps and learn about setting up your infrastructure to run Genesys Multicloud CX private edition.

- Quick deployment tour
- Find all service-level deployment guides

Configure your environment

Learn about the topology, namespace recommendations, and security settings for configuring your Genesys Multicloud CX private edition environment.

- Network settings
- Creating namespaces
- Configuring logging
- Configuring monitoring

Deploy private edition

Learn about how to deploy Genesys Multicloud CX private edition.

- Order of services deployment
- Downloading your Genesys Multicloud CX containers
- Overriding Helm chart values
- Service priorities for Genesys Multicloud CX services
- Setting up a CD pipeline

Upgrade

Learn about different types of upgrades supported by Genesys Multicloud CX services.

- Upgrade overview
- Upgrade strategies

- Upgrade process
- Rollback

Uninstall

Learn about the steps involved in uninstalling Genesys Multicloud CX private edition.

- Uninstall instructions for services

References

- Public Repository Links

About Genesys Multicloud CX private edition

Contents

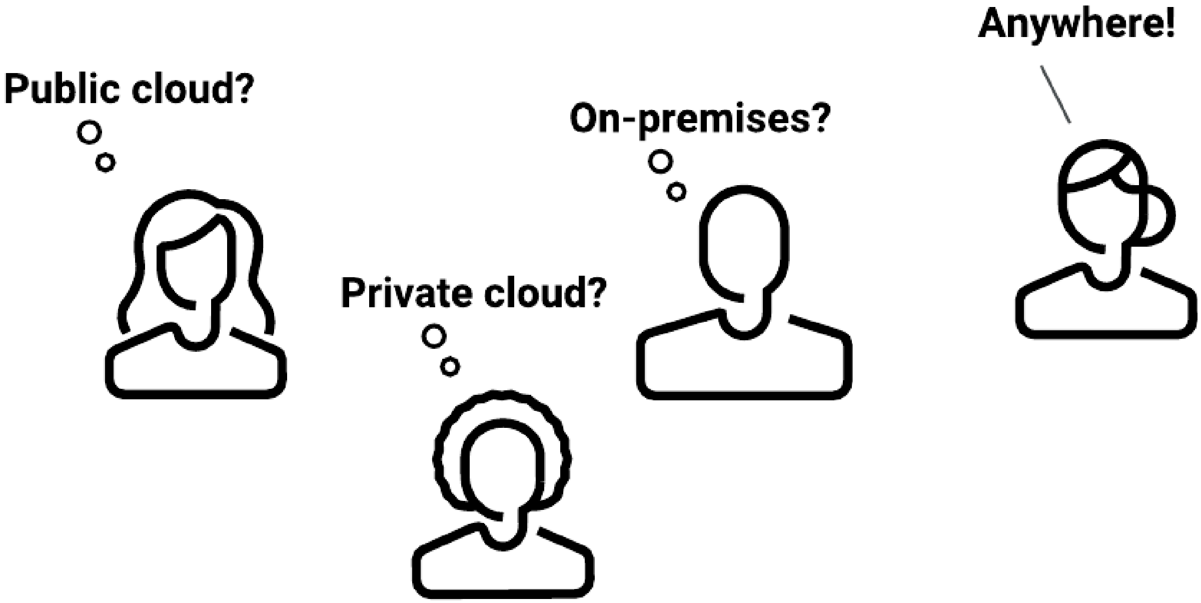
Learn about the Genesys Multicloud CX private edition offering and its key features.

Related documentation:

-
-

RSS:

- For private edition



Genesys Multicloud CX private edition is a microservices-based contact center offering that adopts containerization technology for all the components. Containerized Genesys Multicloud CX services are cloud-native and portable, meaning that the Genesys Multicloud CX private edition software offers the same set of features whether it is deployed on public or private clouds, on virtual machines, or on bare-metal servers on-premises.

Genesys Multicloud CX private edition has been designed to:

- Allow a customer or partner to deploy Genesys Multicloud CX on a number of Kubernetes platforms (whether on-premises, or in a public or private cloud)

- Improve deployment and monitoring
- Meet and exceed the scalability, security, and reliability requirements of the largest enterprise customers

With Genesys' support of Kubernetes and Helm, you can quickly set up your contact center with seamless automated deployments, get faster upgrades, monitor the services, and trigger alerts for faulty systems.

Key Features of Genesys Multicloud CX private edition

Benefits	Description
Cloud-native	Cloud-native architecture scales automatically to meet the customer demand without extra overhead costs.
Automated Deployment	Deployment artifacts and features allow customers to automate the deployment of the Genesys Multicloud platform to their choice of Infrastructure such as cloud provider and premise.
High Availability	The HA capabilities have expanded through a new functionality (for example, the change from active-passive model to N+1) and utilization of Kubernetes functionality such as auto-restart of pods. The platform can be deployed to take advantage of multiple availability zones and geo-diverse regions to increase the availability of the platform.
Autoscaling	Autoscaling is the ability to automatically create service copies to meet the demand. When demand decreases, microservices are chosen to be decommissioned to an appropriate level of readiness.
Monitoring	All services in the platform have a rich set of metrics to allow you to monitor the operational health of the platform as a whole and the individual services in order to detect potential problem areas sooner.

Supported Kubernetes platforms

Genesys supports the following Kubernetes platforms for its private edition offering:

- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)

For more information on the third-party dependencies required for the related Kubernetes platforms, see Software requirements.

Architecture

Contents

Understand the architecture and components of Genesys Multicloud CX private edition; the supported third-party back-end services; and how they all work together in both single- and multi-region deployments.

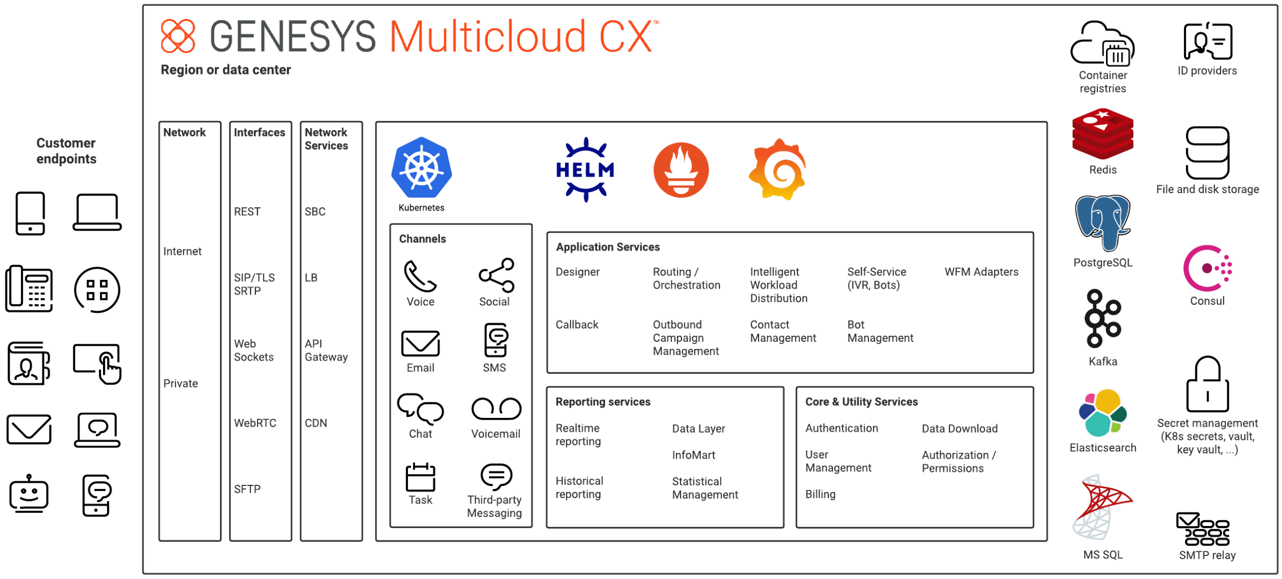
Related documentation:

-
-

RSS:

- For private edition

As mentioned in the About page, Genesys Multicloud CX private edition gives you the flexibility to deploy your contact center on a public cloud or a private one—and even on bare metal servers that reside within your corporate data center.

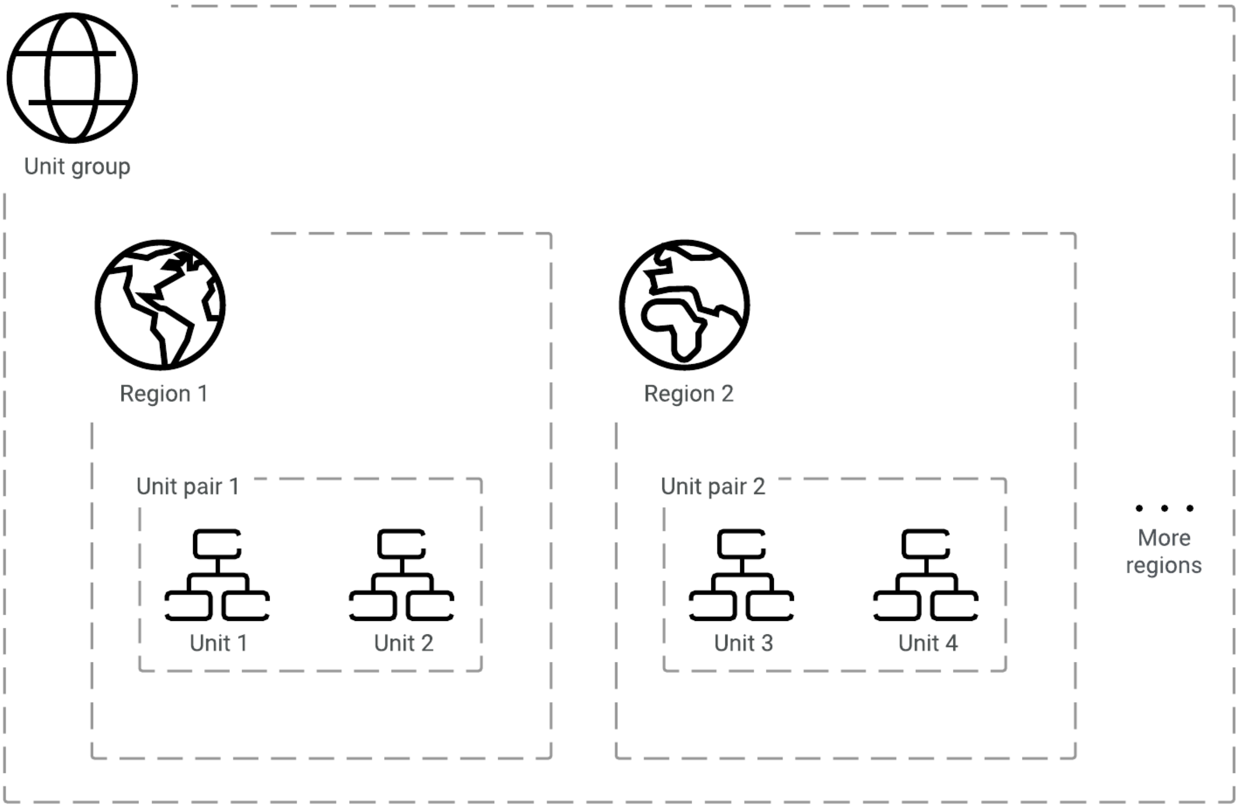


Platform and network

Platform

The basic architecture for private edition involves three levels:

- A **unit** consists of all of the Genesys Multicloud CX and third-party services and resources required to create a single instance of Genesys Multicloud CX private edition. This instance is hosted within a single region or data center.
- A **unit group** brings together a network of units to create a global platform for tenants that covers all geographical regions
- A **unit pair** consists of two units that are part of a unit group and that are both located within a specific geographical region



The following definitions describe important features of the private edition architecture:

- **Region**—A set of isolated and physically separated Availability Zones deployed within a latency-defined perimeter and connected through a dedicated low-latency network within a specific geographical area. **Note:** Regions as defined here are a feature of the cloud deployment architecture and are not supported in the private data center deployment architecture, which does not use Availability Zones.
- **Data center**—A building, a dedicated space within a building, or a group of buildings used to house computer systems and associated components, such as telecommunications and storage systems.
- **Availability Zone (AZ)**—A discrete location within a region that is designed to operate independently from the other Availability Zones in that region. Because of this separation, any given Availability Zone is unlikely to be affected by failures in other Availability Zones. **Note:** Availability Zones are a feature of the cloud deployment architecture and are not supported in the private data center deployment architecture.
  Note: Google Kubernetes Engine (GKE) uses the term "Zone" instead of "Availability Zone."

- **Tenant**—A business entity that has common goals and procedures, and occupies part or all of a contact center. Tenants that share a contact center could be different businesses, or different divisions within the same business.
- **Multi-tenancy**—The partitioning capacity for a platform to host and manage tenants. Each tenant is configured individually and separately.

The following sections provide a more in-depth description of the characteristics of the three levels of the private edition architecture.

Units

A unit can either be dedicated to a specific tenant or used for multiple tenants. The unit and its services and resources can be distributed across Availability Zones if the environment has them.

A unit is composed of the following:

- Network access services (load balancers, firewalls, SBC, and so on)
- A Kubernetes cluster with all of the private edition service pods
- Third-party services (Postgres, Redis, Consul, Kafka, and so on)

There are two main types of units:

- A **primary unit** centralizes certain services used by all regions for a specific tenant, such as Designer application creation, historical reporting, or UI. There is only one primary unit in a unit group. In the current architecture, digital channels are only supported by the primary unit.
- A **secondary unit** only supports voice-related services at this time. Digital channels are only supported by the primary unit.

Unit pairs

Unit pairs provide the following capabilities:

- Redundancy within a geographical region. This geo-redundancy is built into the private edition services.
- Tenants can be distributed across the two units to help reduce the blast area in case of a major failure

A unit pair can consist of a primary unit and a secondary unit, or of two secondary units. **Note:** Unit pairs are only supported by voice-related services at this time.

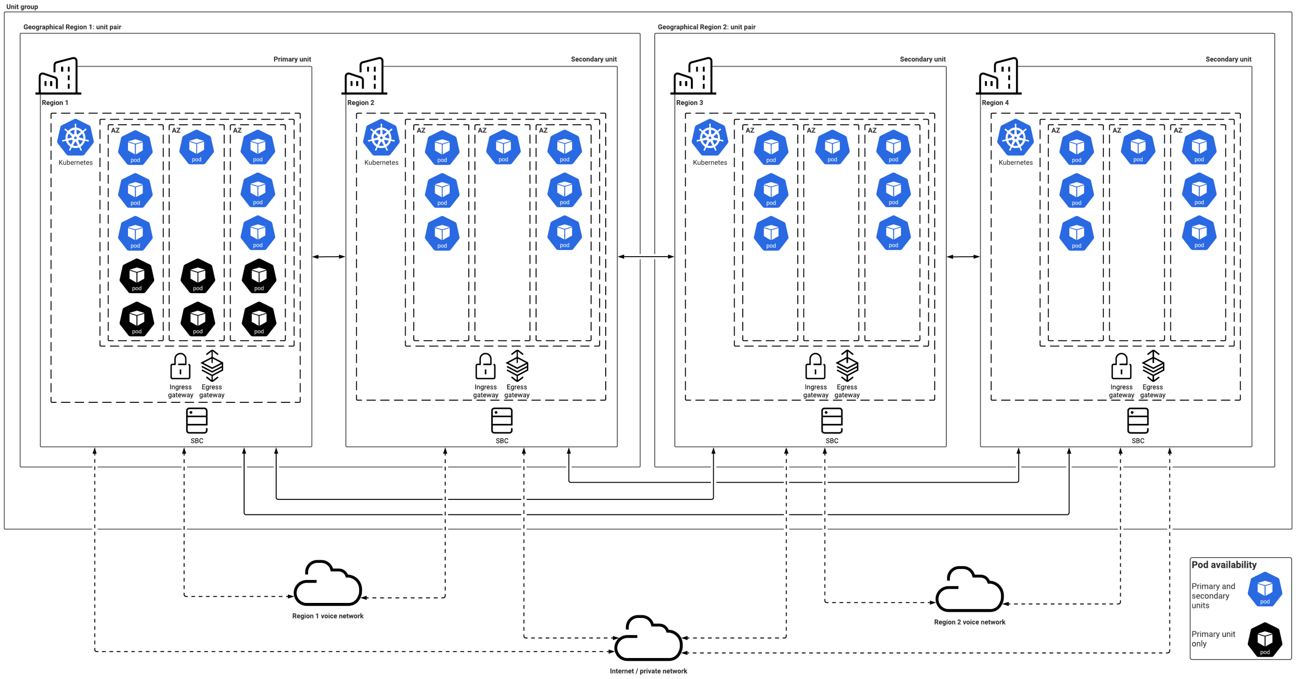
Unit groups

Unit groups interconnect their constituent units by means of a network peering solution, and all inter-region traffic uses either your network connectivity or the network connectivity of your cloud provider. Each group contains a primary unit in one region in the group. This primary regional unit hosts all of the private edition services, while the secondary regional unit hosts only a subset of private edition services. A unit group must contain at least one unit pair. If you add a new geographical region, then you must add a unit pair to the unit group in that geographical region.

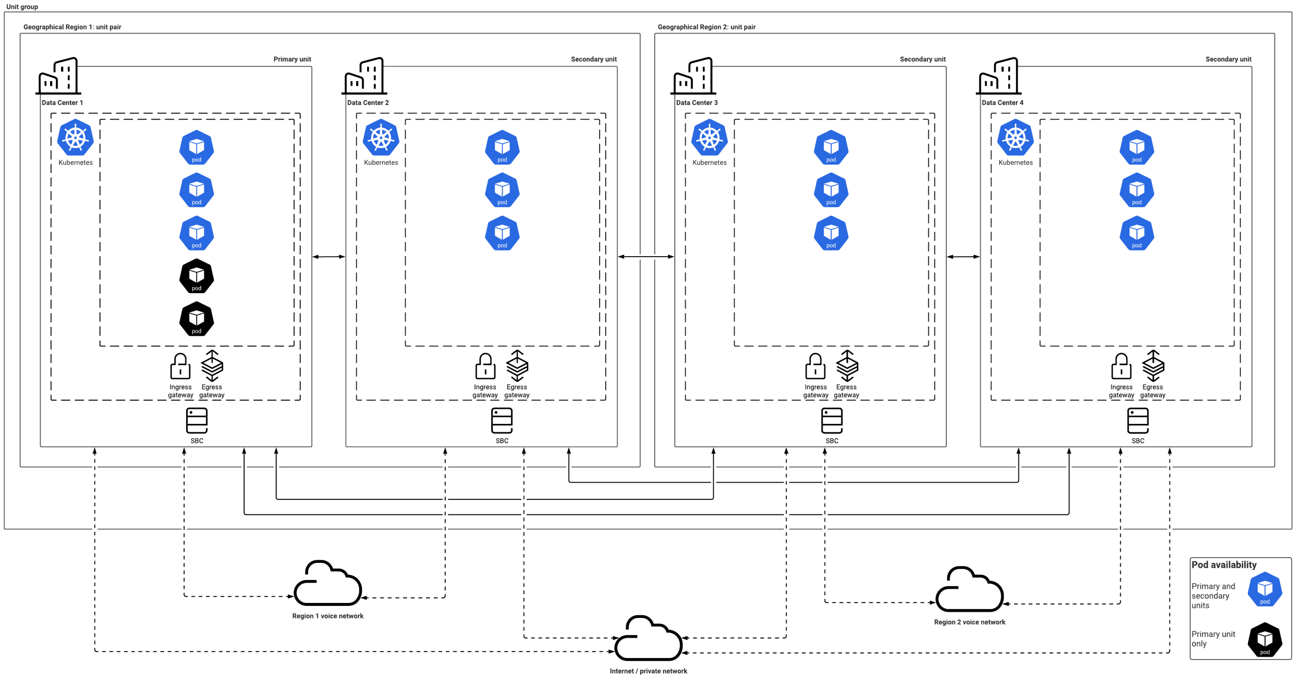
Deployment models

Genesys Multicloud CX private edition allows you to set up a highly available and resilient infrastructure whether you are using a cloud deployment or hosting it in a private data center, as shown in the following diagrams.

Cloud architecture



Private data center architecture



Multiple regions and data centers

The platform supports deployment across multiple regions and data centers. This capability provides extra availability for the voice-related services, with a global view.

- **Call routing and processing**—The ability to distribute call processing across regions. Also, to centrally create and distribute Designer applications across regions.
- **Agent availability**—The ability to have a call processed by agents from any region
- **Data sovereignty**—The ability to contain the data (recordings, and so on) and processing of the call within the region in which the call originated
- **Reporting (Real-time and Historical)**—The ability to provide a global view across all regions
- **Tenant provisioning**—The ability to centrally provision the contact center across multiple regions
- **Callback**—The ability to use a central service to provide in-queue callback across regions

Subnets

Subnets are your responsibility: you must create a subnet for the Kubernetes cluster to accommodate the Genesys Multicloud CX services.

Network access

For information about network access, see Networking Overview.

Supported services

Genesys Multicloud CX private edition supports the services listed on the Genesys Multicloud CX services list.

Software requirements

Genesys Multicloud CX private edition requires the software and versions listed on the software requirements page. Note that you are responsible for installing and deploying the appropriate third-party software in a way that best suits your requirements and the requirements of the Genesys Multicloud CX services.

Kubernetes clusters

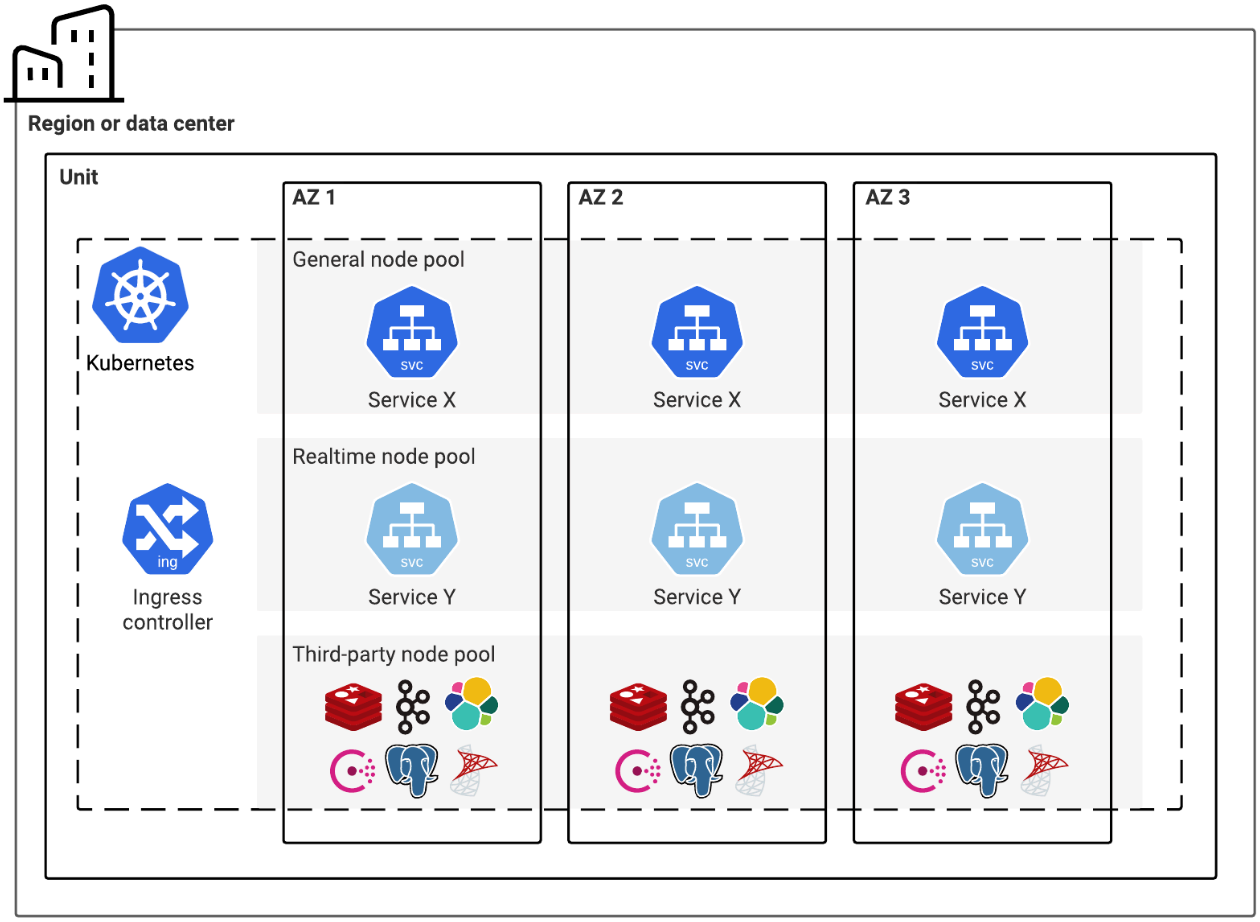
All Genesys Multicloud CX services must run in Kubernetes. Required third-party services can be managed either outside Kubernetes or within Kubernetes. Kubernetes is responsible for managing the running of services, such as monitoring them, restarting them, and so on.

Private edition does not currently support multiple instances of the platform in a single Kubernetes cluster. In other words, if you want to set up separate environments for testing, staging, production, and so on, you must deploy the private edition instances for the various environments in separate clusters.

Deployment

Genesys currently recommends that you use node pools to deploy Kubernetes for the Genesys Multicloud CX services that are hosted within each unit.

- **Node pools**—Genesys recommends that you use the following node pools. Our Helm charts include overrides for the nodeSelector attribute. Use these overrides to assign a service to the appropriate node pool.
    - **General node pool**—This node pool is where most of the Genesys Multicloud CX services are deployed. This type of pool uses general-purpose compute instances with Premium SSD, which provide a 4:1 ratio of memory GB to vCPU.
    - **Real-time node pool**—This node pool is for stateful voice services that require a drain time of 60 minutes or longer to maintain active voice sessions. It uses general-purpose nodes with Premium SSD.
    - **Third-party service node pool (optional)**—This node pool is only needed if you are going to deploy data stores and other third-party services in Kubernetes, such as Redis, Kafka, Postgres or Elasticsearch. These services generally need locally optimized storage and will use the storage-optimized nodes with directly attached NVMe and Premium SSD, which provide an 8:1 ratio of memory GB to vCPU.



Networking

- For information about private edition's general networking requirements and constraints, see Networking Overview
- For information about networking settings for Kubernetes clusters, see Network Settings

Service priorities

For more information, see Service Priorities.

Autoscaling

Most services scale their Kubernetes pods by using the Horizontal Pod Autoscaler. However, this tool can only use CPU or memory metrics from the Kubernetes Metric Server in the HorizontalPodAutoscaler Object. Private edition also works with the Kubernetes cluster scaler. Note that each service provides its own autoscaling rule, and that the autoscaling rule for a specific service is stored in the Helm charts for that service.

Genesys uses the third-party KEDA open-source autoscaler for Genesys Multicloud CX services that require custom metrics from Prometheus. Use the included Helm override attributes to adjust the defaults for each service.

You must perform your own scaling operations on the Kubernetes control plane. The operational requirements of this scaling depend on the size of your contact center. For large installations, you might need to deploy multiple clusters and distribute the Genesys Multicloud CX services across them.

ConfigMaps

Private edition uses ConfigMaps to pass variables and data to the deployed services. This allows each service to be separate from its configuration data, which is a factor in making each service immutable. Genesys provides Helm override attributes that you use to set the configuration values for each service. For more information, see the appropriate service guide.

Operators

You can use operators to deploy most third-party services into clusters. Note that Genesys does not provide operators to deploy Genesys Multicloud CX services.

GKE

> ### Important
> The Genesys implementation of the Google Kubernetes Engine (GKE) is only available on the Google Cloud Platform using the Cloud deployment model.

Notes on what is supported

- Genesys supports deployments to GKE on the Google Cloud Platform (GCP) public cloud
- Genesys recommends using the Container-Optimized OS with ContainerD for Linux nodes.
- Private edition has only been tested on public GKE clusters (public IP addresses for the control plane, worker nodes, and public load balancers) and only supports VPC-native clusters that use alias IP ranges to provide VPC-routable IPs for direct pod access, along with direct access to other Google cloud services. More specifically, private edition does not support routes-based clusters.
- In order to support VPC peering when creating the VPC-native GKE cluster, you must enable IP Alias by including the **--enable-ip-alias** flag, as mentioned in the VPC Peering Restrictions documentation. This eliminates the necessity of creating and exporting routes. By default, VPC network peering with GKE is supported when used with IP aliases.

Not supported

- Although GKE supports a hybrid model, with workloads running both on-premises and in the cloud, private edition does not currently recommend and has not validated splitting its services between different environments or across multiple clusters.

All private edition components must run in a single GKE cluster on the public Google Cloud and not in any private or government GKE instances, or in any GKE instances that are hosted on-premises.

- Private edition does not currently recommend and has not validated hybrid models, or on-premises deployment of GKE.
- GKE clusters in Autopilot mode are not supported.
- GKE Sandbox (gVisor virtualized kernel) and Customer-Managed Keys (CMKs) are not supported.
- Network policies are not provided or supported and all ingress and egress pod traffic must be allowed between all namespaces.
- The GKE Service Mesh add-on, which uses Istio, is not supported. Customers must deploy the Consul Service Mesh instead.
- GKE Ingress is not supported for the initial private edition GKE offering. Instead, private edition requires deploying ingress-nginx.
- Private edition does not support routes-based clusters.

Security

Genesys Multicloud CX private edition has been developed using industry-standard tools and best practices to identify and eliminate security vulnerabilities.

You are responsible for setting up security in the cluster.

For more information about security-related topics, see Security overview.

High-Availability

For more information, see High Availability and Disaster Recovery.

Data stores

Each service must have its own data store cluster or instances, which must not be shared in production environments unless they are under the same service group.

- All data stores must enable and deploy their high availability (HA) functionality
- All data stores must be distributed across Availability Zones, if they are available
- All data stores must support TLS connections and authentication, as appropriate

Here are the data stores used by each service:

Elasticsearch / OpenSearch

> **Important**
>
> Private edition does not currently support authentication for Elasticsearch or OpenSearch.

The Elasticsearch and OpenSearch services are shareable across tenants, but the tenant data is never shared.

Service	Type of Data	Cross region replication
Designer	Application Analytics data	No
IWD	Interaction and Queue Analytics data	No
TLM	Searchable telemetry data	No
UCS	Searchable contact and interaction history data	No
GWS	Searchable Statistics data	No
CXC	Campaign Analytics	No

Redis

The Redis service is shareable across tenants, but the tenant data is never shared.

Private edition requires the following features for Redis:

- Must support cluster mode
- TLS provisioning
- If you want secure connections to Redis, you must provision Access Control Lists (ACLs) for authentication
- A minimum of three nodes
- A minimum of one replica
- Memory size setting must be based on the services algorithm
- A minimum of two shards per DB
- Must support persistence for services that require it

Service	Type of Data	Cross region access
Pulse	runtime statistics	No
Tenant	stream of tenant data	Yes
CXC	runtime campaign and calling list status	No
Designer	config data	No
GES	runtime callback status and data	No
Nexus	runtime messaging session data	No
IWD	Historical reporting data	No
VMS (all of these services have separate keys (registrate, ORS, ORS stream, Callthread, Agent, Config, SIP, RQ))	runtime interaction, agent, registrations, config and routing request streams, scxml session data	Yes (not all)
GAuth	authentication session data	No
GWS	cached statistics, interaction and agent data	No

SQL databases

> ### Important
> All SQL databases except GVP must use Postgres. GVP only supports the use of MS SQL.

You can set up your private edition SQL database instances in either of the following two ways. You can also use the first scenario for some services, and the second scenario for other services:

- Use a separate SQL database instance for each service
- Use a single SQL database instance for a combination of services

However, in each of these scenarios:

- Each service creates its own databases
- Tenant data is never shared.

Service	Type of Data	Shared across tenants	Cross region replication
GCXI	metadata for reports	Yes	No
GVP RS - MS SQL	GVP reporting data	Yes	No
GVP CFG	config data	Yes	No
IXN	digital interaction data	No	No
Pulse Permissions	config data	No	No
Tenant	config and campaign data	No	Yes
GES	config data	Yes	No
GIM	Historical reporting data	No	No
IWD	IWD config data	No	No
Nexus	config data	Yes	No
UCS	config data	Yes	No
UCS	contact, transcriptions, emails, interaction history	No	No
Gauth	config data	Yes	Yes
GWS	config data	Yes	Yes

File and disk storage

For more information, see Storage Requirements.

Voice Connectivity

For more information, see Voice Connectivity.

Email

The following private edition services send emails as part of their service:

- Voicemail
- GCXI
- Pulse

These services use standard mail agents on the operating system over SMTP via ports 25 and 587.

To use email with a service, you must set up the appropriate SMTP relay to relay messages from that service to your email system or email service. **Note:** This must be done from the Kubernetes clusters.

Content delivery networks (CDNs)

The WWE service that runs within private edition delivers static content. You can host this content from a CDN or from NGINX running in the Kubernetes cluster.

Monitoring

Private edition provides appropriate interfaces for you to use your own monitoring tools. For the purposes of this software, monitoring encompasses:

- Metrics
- Logging
- Warnings
- Alerts

Monitoring (metrics)

Private edition provides a set of Prometheus-based metrics and defines an endpoint which the Prometheus platform can scrape. However, it does not provide a Grafana dashboard or Alert rule definitions.

GKE

Private edition uses the Google Cloud operations suite for GKE for system and workload monitoring. The Google Cloud Operations Suite also provides a GKE dashboard for metrics and alerts.

You can enable the GKE workload metrics in order to scrape application metrics based on the PodMonitor resource definition. If a service doesn't provide a PodMonitor resource, then you might need to deploy a Prometheus server with a Stackdriver collector in order to expose Genesys custom application metrics as external metrics in Cloud Monitoring, which does incur an additional cost.

AKS

You can use the Container insights feature in Azure Monitor for monitoring system and workloads in private edition.

Azure Monitor Metrics feature supports collecting metrics from monitored workloads and you can create alerts based on the collected metrics.

Azure Monitor Metrics also supports Prometheus metrics collected from Kubernetes clusters. For more information, refer Azure product documentation.

Logging

Private edition provides the vast majority of its log data via stdout and stderr. In some exceptional cases, data is logged to disk.

GKE

Private edition uses the Google Cloud operations suite for GKE for system and workload logging. The Google Cloud Operations Suite also provides a Logs Explorer for system and workload logs.

You can either:

- Send your logs to Stdout to be collected and exposed in the Logs Explorer as part of Cloud Logging
- Send them to an RWX/NFS-style log volume provided by a shared Cloud Filestore for legacy or high volume logging.
  Note: RWX/NFS logging will be deprecated in the near future.

AKS

You can use the Log Analytics workspace feature in Azure Monitor for collecting log data of system and workloads in private edition. You can create single or multiple log analytics workspaces based on your organizational needs.

For more information on configuring logs in Azure Monitor log workspaces, refer Azure product documentation.

Integrations

Private edition support integrations with a wide variety of systems to provide an enriched customer experience, including in the following areas:

- Bot platforms, such as Google Dialogflow and AWS Lex
- WFM platforms, such as Verint and Nice
- Email systems
- Identity providers
- Reporting platforms, including business intelligence tools
- Messaging and social platforms
- CRM and BPM systems
- Biometrics systems

Genesys Multicloud CX private edition services

Contents

- 27 Helm charts and containers
- 28 Guides
- 29 Release Notes
- 30 Helm charts and containers
- 31 Guides
- 32 Release Notes
- 33 Helm charts and containers
- 34 Guides
- 35 Release Notes
- 36 Helm charts and containers
- 37 Guides
- 38 Release Notes
- 39 Helm charts and containers
- 40 Guides
- 41 Release Notes
- 42 Guides
- 43 Release Notes
- 44 Helm charts and containers
- 45 Guides
- 46 Release Notes
- 47 Helm charts and containers
- 48 Guides
- 49 Release Notes
- 50 Helm charts and containers
- 51 Guides
- 52 Release Notes
- 53 Helm charts and containers
- 54 Guides
- 55 Release Notes
- 56 Helm charts and containers

List of private edition services and their microservices.

Related documentation:

-

RSS:

- For private edition

The following table presents the list of Genesys Multicloud CX private edition services and their microservices. These services do not require any technical licenses or activation files for deployment or operation in any environment. For more licensing information, see Licensing requirements.

Services	Included services	Service documentation
	• CX Contact API Aggregator • CX Contact Campaign Manager • CX Contact Compliance Manager • CX Contact Dial Manager • CX Contact Job Scheduler • CX Contact List Builder • CX Contact List Manager • CX Contact UI	Guides • Release Notes • Helm charts and containers •
	• Designer • Designer Application Server	Guides • Release Notes • Helm charts and containers •

Services	Included services	Service documentation
	• AI Connector	Guides • Release Notes • Helm charts and containers •
	Single microservice only	Guides • Release Notes • Helm charts and containers •
	• Authentication Service • Authentication UI • Environment Service	Guides • Release Notes • Helm charts and containers •
	• Genesys CX Insights	

Services	Included services	Service documentation
	• Reporting and Analytics Aggregates	Guides • • • Release Notes • Helm charts and containers • •
	Single microservice only	Guides • Release Notes • Helm charts and containers •
	• GIM • GIM Config Adapter • GIM Stream Processor	Guides • Release Notes • Helm charts and containers •

Services	Included services	Service documentation
	• Pulse Web Service • Tenant Data Collection Unit (DCU) • Tenant Load Distribution Server (LDS) • Tenant Permissions Service	Guides • Release Notes • Helm charts and containers •
	• Voice Platform Configuration Server • Voice Platform Media Control Platform • Voice Platform Reporting Server • Voice Platform Resource Manager • Voice Platform Service Discovery	Guides • Release Notes • Helm charts and containers •
	• Agent Setup • GWS Chat Service • GWS Configuration Service • GWS Data Collector Service • GWS Ingress • GWS Interaction Service • GWS OCS Service • GWS Provisioning Service • GWS Services • GWS Setting Service • GWS Statistics Service	Guides • Release Notes • Helm charts and containers •

Services	Included services	Service documentation
	• GWS UCS Service • GWS Voice Service • GWS Workspace Service	
	Single microservice only	
	Single microservice only	Guides • Release Notes • Helm charts and containers •
	Single microservice only	Guides • Release Notes • Helm charts and containers •
	Single microservice only	Guides • Release Notes •

Services	Included services	Service documentation
	Single microservice only	Guides • Release Notes • Helm charts and containers •
	Single microservice only	Guides • Release Notes • Helm charts and containers •
	• Agent State Service • Call State Service • Config Service • Dial Plan Service • FrontEnd Service • ORS • Voice Registrar Service • Voice RQ Service • Voice SIP Cluster Service • Voice SIP Proxy Service • Voicemail	Guides • • • Release Notes • • • • • • • • • •

Services	Included services	Service documentation
		• • Helm charts and containers •
	• WebRTC CoTurn Service • WebRTC Gateway Service	Guides • Release Notes • Helm charts and containers •
	Single microservice only	Guides • Release Notes • Helm charts and containers •

High availability and disaster recovery

Contents

High availability (HA) and disaster recovery (DR) are two important factors in establishing a resilient infrastructure. This article describes the two supported architecture types for HA and DR, as well as the HA and DR modes supported by the private edition services.

Related documentation:

-
-

RSS:

- For private edition

Modern software environments demand two major types of agility:

- The ability to autoscale—that is, to rapidly increase processing power to handle a growth in interaction volume
- **Resiliency**—that is, the ability to fail over after losing one or more services—or even a whole data center or region

The second type of agility—the ability to bounce back from a failure—is broadly divided into two types of activity, each with its own requirements:

- **High availability (HA)** is the use of built-in redundancy to handle the failure of a service within a single region or data center
- **Disaster recovery (DR)** is the ability to continue processing after losing a whole region or data center, by failing over to another region or data center

Genesys Multicloud CX private edition allows you to set up a highly available and resilient infrastructure whether you are using a cloud deployment or hosting it in a private data center.

Note, however, that these two types of deployments require somewhat different architectures, as discussed below.

> **Important**
>
> Before you continue, review the platform section of the private edition architecture page for an in-depth discussion of key components of the private edition architecture, such as **unit pairs** and **Availability Zones**.

Key architectural distinctions

Both the cloud and private data center architectures use multiple geographical regions that are hosted within a single unit group. And in both types of environment, all of the unit pairs in a deployment are fully meshed with each other.

But the cloud deployment's ability to use Availability Zones makes its redundancy features more robust, as shown in the following table:

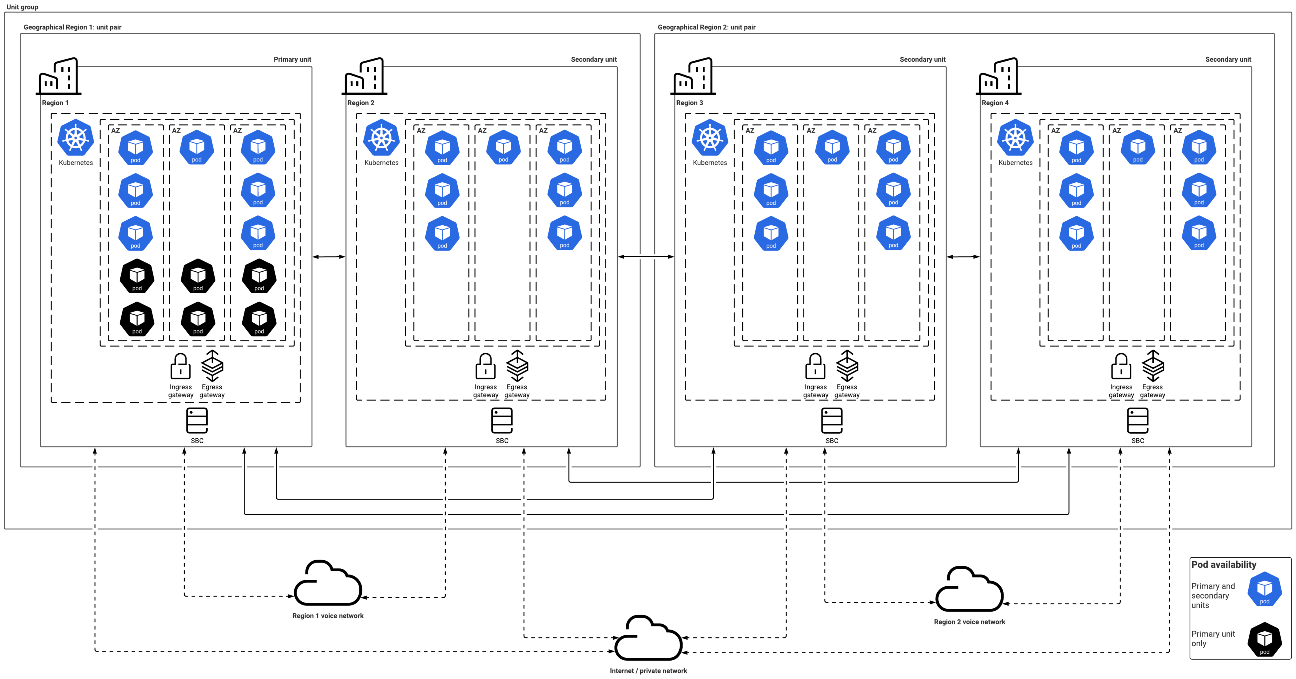
Deployment type	Redundancy type	Characteristics
Cloud	Availability Zones within regions	Multiple data centers in a small geographical area—can share a single Kubernetes cluster
Private data center	Physically discrete data centers	Data centers cannot share a Kubernetes cluster

Cloud architecture

One of the most important advantages of a cloud architecture is the enhanced redundancy through the use of Availability Zones (AZs). As discussed in the platform section of the private edition architecture page, an AZ is a discrete location within a region that is designed to operate independently from the other Availability Zones in that region. Because of this separation, any given Availability Zone is unlikely to be affected by failures in other Availability Zones.

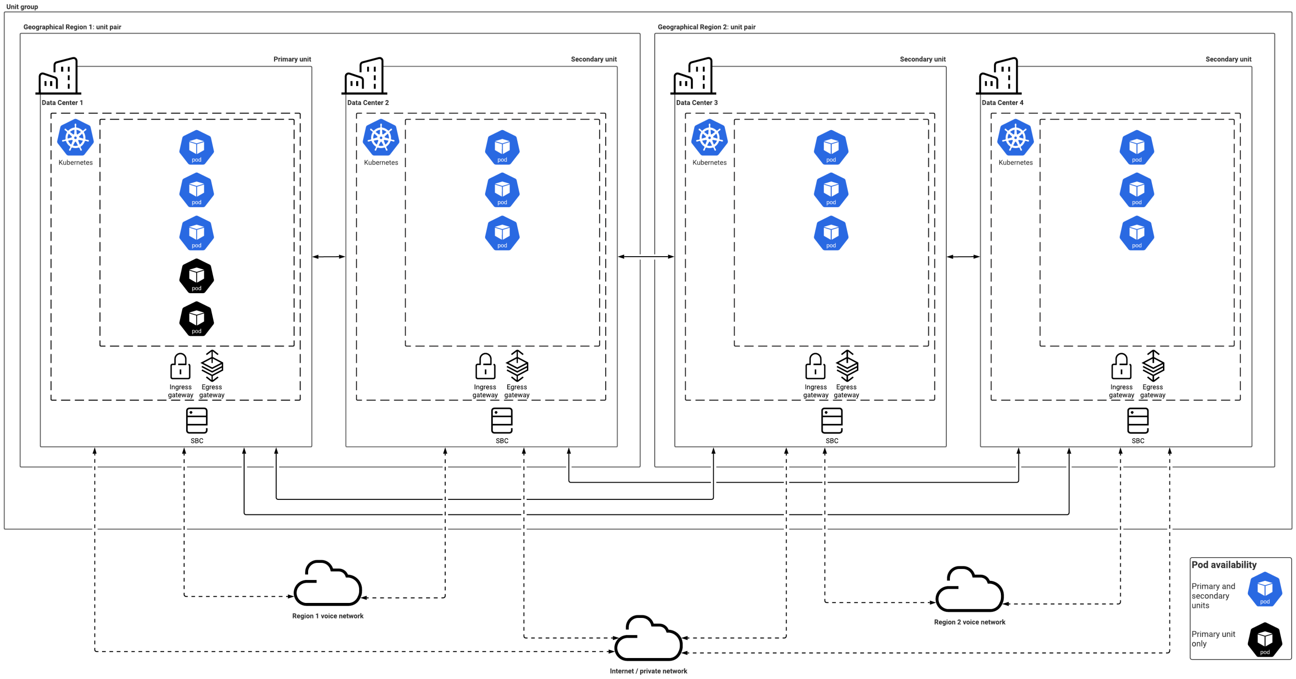
In the cloud architecture, high availability is achieved by deploying instances within different Availability Zones.

> **Important**
>
> Black pod icons indicate services that can only be hosted in the primary unit.



Private data center architecture

> **Important**
>
> Black pod icons indicate services that can only be hosted in the primary unit.



Planning for high availability

Private edition services scale automatically to meet demand. And when a service fails, private edition's high availability features enable an auto restart of that service.

For first-time deployments, you must plan:

- The number of nodes
- The number of pods that each node must run in your Kubernetes cluster

In order to reduce service disruptions, Genesys recommends that you run a minimum of three pod replicas for each service. Use the Sizing Calculator to determine the infrastructure requirements for achieving high availability in your contact center.

Resiliency modes of private edition services

High availability modes

Private edition services maintain high availability by using the following modes:

> **Important**
>
> Some services support more than one HA mode.

High availability modes

Mode	Description
N = 2 (active-active)	The service is running on two nodes simultaneously. If one fails, the other takes over.
N = 1 (singleton)	The service is running on a single node. If that node fails, a new node is started to take over processing for that service.
N = N (N+1)	The service normally runs on N nodes. If a node fails, a new node is started to replace the failing node.
Cron jobs	Some services run as cron jobs, meaning that normal HA is not applicable

Disaster recovery modes

Private edition services achieve disaster recovery by using the following modes:

Disaster recovery modes

Mode	Description
Active spare	A complete production replica is in place and serves traffic during normal operations
Limited active spare	A complete production replica is in place and serves traffic during normal operations, but the data is only used in case of disaster
Pilot light	The bare minimum configuration is in place to get the system back within a short time period. For example, there might be a read replica for a database. Application servers and web servers are deployed after the disaster.
Not supported	Disaster recovery is not supported for this service

Modes for each service

The following table displays the high availability and disaster recovery modes used by private edition services.

> **Important**
>
> Disaster recovery is not supported for services that are only available in the primary

unit.

Service & Included Services	High Availability	Disaster Recovery	Where can you host this service?
	N = N (N+1)	Active-spare	Primary or secondary unit

— Designer	N = N (N+1)
	Or
	N = 2 (active-active)
		Pilot light	Primary unit only

— Designer Application Server	N = N (N+1)
	Or
	N = 2 (active-active)
		Active-spare	Primary or secondary unit

— Voice Platform Configuration Server	N = 1 (singleton)	Active-spare	Primary or secondary unit
— Voice Platform Media Control Platform	N = N (N+1)	Active-spare	Primary or secondary unit
— Voice Platform Reporting Server	N = 1 (singleton)	Active-spare	Primary or secondary unit
— Voice Platform Resource Manager	N = 2 (active-active)	Active-spare	Primary or secondary unit
— Voice Platform Service Discovery	N = 1 (singleton)	Active-spare	Primary or secondary unit
	N = N (N+1)	Active-spare	Primary or secondary unit
	N = N (N+1)	Active-spare	Primary or secondary unit
	N = N (N+1)	Not supported	Primary unit only
	N = N (N+1)	Not supported	Primary unit only
	N = N (N+1)	Not supported	Primary unit only
	IWD Data Mart is a Cronjob that runs on a per-tenant basis, so High Availability (HA) is not applicable.		
	N = 1 (singleton)	Not supported	Primary unit only

Service & Included Services	High Availability	Disaster Recovery	Where can you host this service?
	N = N (N+1)	Not supported	Primary or secondary unit
— Genesys CX Insights	N = 2 (active-active)	Not supported	Primary unit only
— Reporting and Analytics Aggregates	N = 1 (singleton)	Limited active spare	Primary or secondary unit
	N = 1 (singleton)	Limited active spare	Primary or secondary unit
	N = 2 (active-active)	Pilot light	Primary unit only
	N = N (N+1)	Active-spare	Primary or secondary unit
	N = 1 (singleton)	Active-spare	Primary unit only
	N = N (N+1)	Active-spare	Primary or secondary unit
	N = N (N+1)	Not supported	Primary unit only
	N = N (N+1)	Active-spare	Primary or secondary unit
	N = N (N+1)	Active-spare	Primary or secondary unit

Networking overview

Contents

Learn about the network access types for voice and data traffic, and the network elements involved in their architecture. For Kubernetes cluster related network settings, see Network settings.

Related documentation:

-
-

RSS:

- For private edition

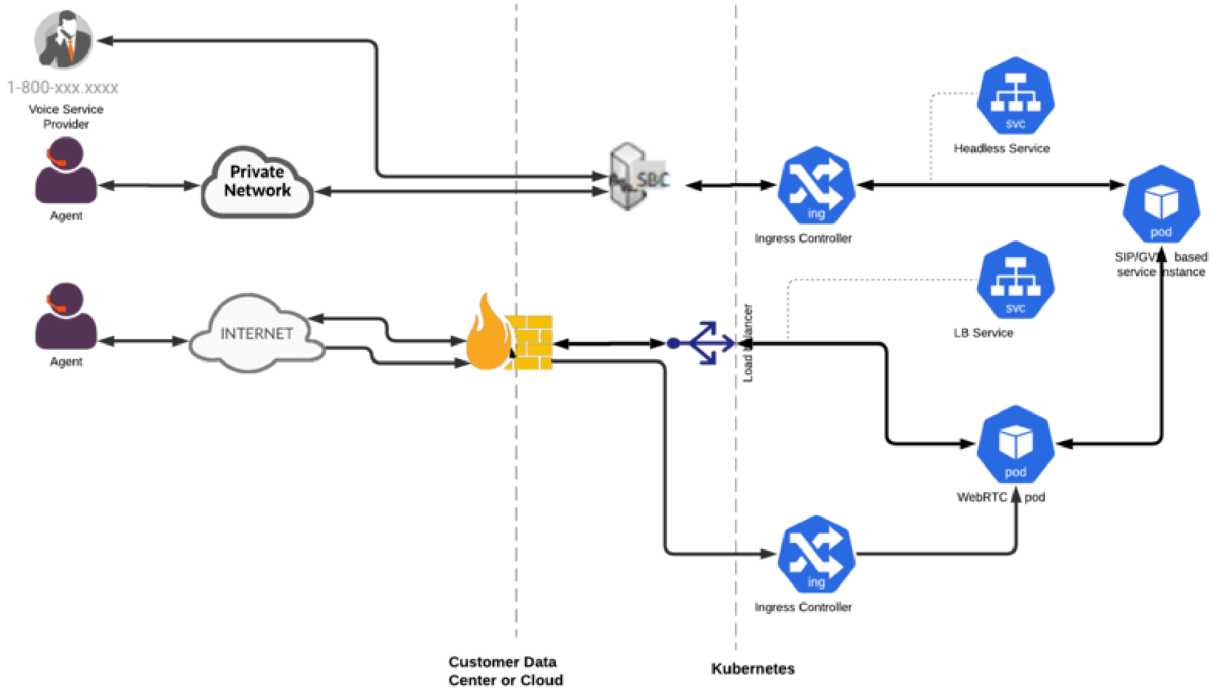
Network access types

There are two types of access to the platform from a tenant perspective:

- **Voice**—Voice (SIP/RTP) traffic
- **Data**—Data traffic.

Voice

The architecture supports SBC integration for both carrier and agent phones, and WebRTC phone access over a data network. The architecture of this voice network is up to you.



Ingress:

- **Firewall for non-HTTP traffic (TCP/UDP)**—Provides network access control (allowlisting, and so on) and a control point for monitoring the traffic.
- Requires VPC or virtual-network native addressing with direct access to the pods IP from SBC.

Data

Your network must include network elements to control the ingress and egress data traffic between the outside world and the Genesys Multicloud CX services running in Kubernetes. However, **you are responsible for determining how to manage access to the Genesys Multicloud CX services**.

The following items are optional, and are shown as examples of how you can control network access.

Ingress:

- **WAF for HTTP and** WebSocket—Provides DDOS protection and being able to terminate TLS at the edge of the network. It is also a control point for monitoring traffic.
- **Firewall for non-HTTP traffic (TCP/UDP)**—Provides network access control (allowlisting, and so on) and a control point for monitoring the traffic.
- **API Gateway**—Enables you to control application and system access to the Genesys Multicloud CX APIs from the standpoint of rate limiting and authorization

Egress:

Implementing Egress is optional and is up to you.

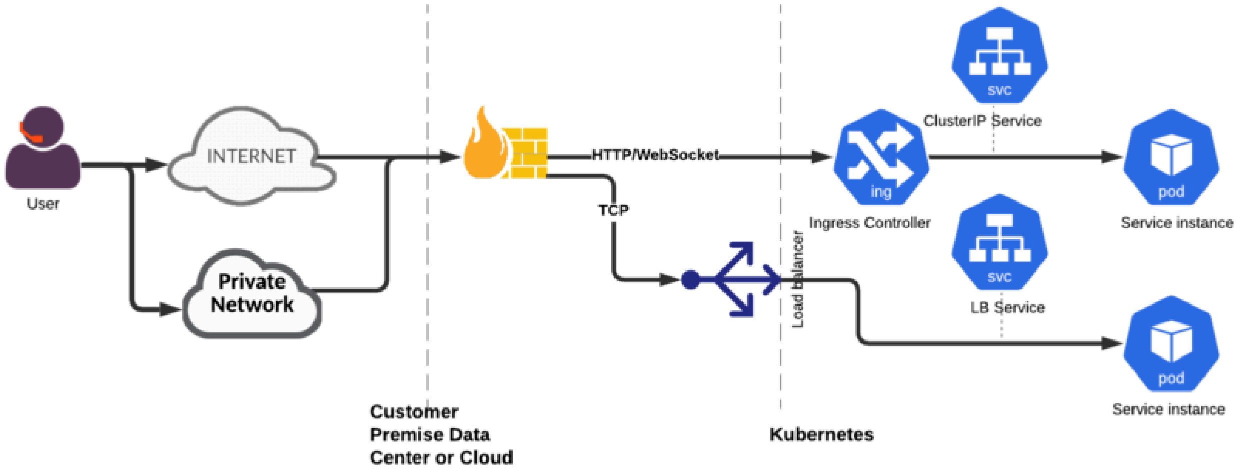
- **Firewall for all external traffic**— Provides network access control (allowlisting, and so on) and a control point for monitoring traffic, to support the security and compliance requirements of your business. All egress traffic to internet destinations must use virtual network-defined or subnet-defined UDR to route traffic through the network firewalls.

Ingress

This architecture uses the following data-related ingress connections:

- HTTP(S)
- WebSocket
- TCP

You must make sure that the right network infrastructure is in place to support your security needs. For more information about the ingress controller and load balancer configurations, see the appropriate service-level guides.

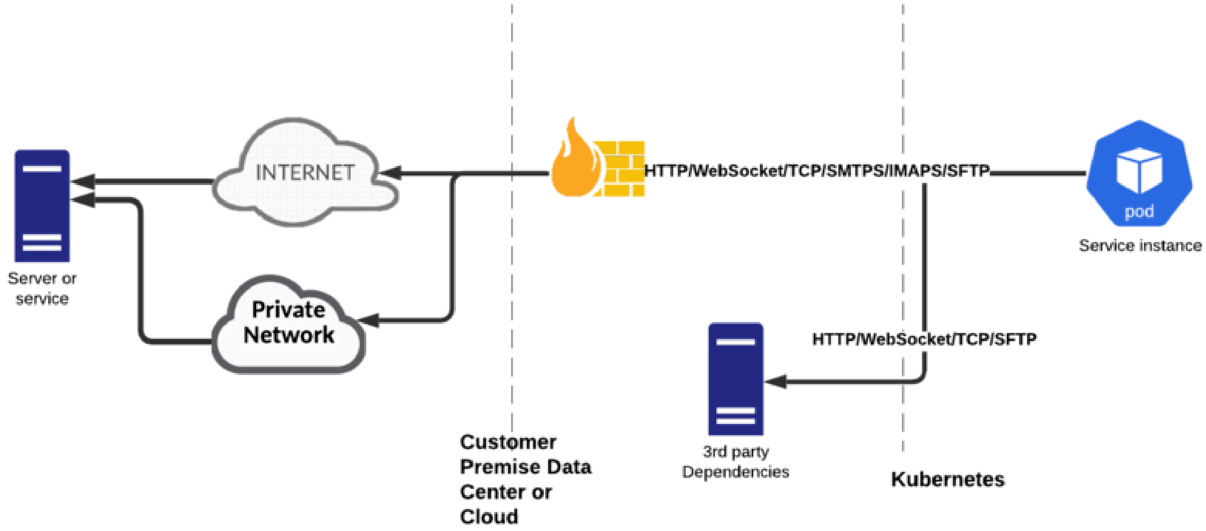


Egress

This architecture uses the following data-related external egress connections:

- HTTPS
- TCP
- SFTP
- IMAPS/SMTPS

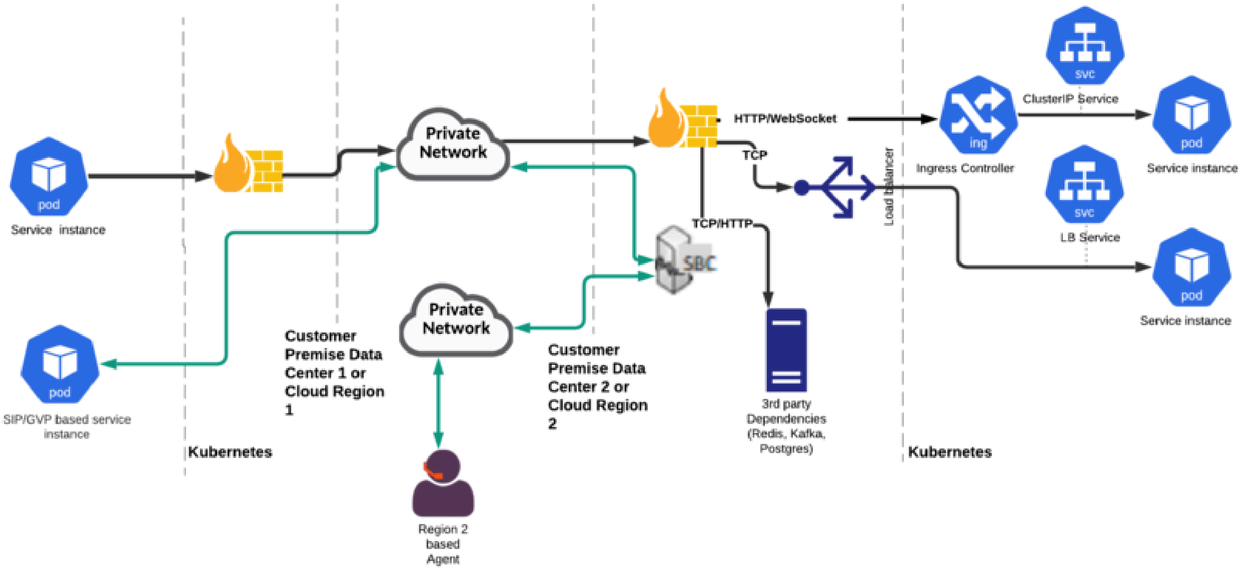
You must make sure that the right network infrastructure is in place to support your security needs.



Cross-Region traffic

This architecture uses the following data-related connections:

- HTTP
- TCP
- SIP/RTP



You must ensure that you have network infrastructure that allows communication between the following:

- **Regional SBCs**—For optimizing RTP connections when calls are crossing regions
- **Kubernetes clusters**—For Genesys Multicloud CX service-to-service communication
- **Third-party dependency clusters**—For Genesys Multicloud CX services to communicate with the clusters in other regions (such as Kafka, Redis, and Postgres)

The network infrastructure must have the following characteristics:

- **Low latency**—To allow for its use by voice traffic
- **Medium bandwidth**

Security overview

Contents

Learn about general security considerations involved in deploying Genesys Multicloud CX private edition.

Related documentation:

-
-

RSS:

- For private edition

Because security is a growing priority for today's enterprises, Genesys works hard to provide a full range of security-related features, such as authentication, role-based access control (RBAC), and many more.

Note, however, that you are responsible for maintaining the security of your private edition infrastructure, such as network security, firewalls, and so on.

Built-in security features

Genesys Multicloud CX private edition has been developed using industry-standard tools and best practices to identify and eliminate security vulnerabilities.

The services that ship with private edition are built with the following features, which provide a strong basis for you to create a secure, enterprise-grade solution:

- Containers are immutable and follow hardening best practices.
- Services run in least-privileged accounts based on the feature functionality needed by a given service.
- You can deploy your Kubernetes clusters into different network segments to partition software into security zones. To do this, you must create new Kubernetes Service Objects with load balancers, which expose the necessary connections between Kubernetes clusters.
- You can put security tool agents on your Kubernetes nodes to carry out the appropriate security tasks, such as host-based intrusion detection system (HIDS), file integrity management (FIM), user and entity behavior analytics (UEBA), and so on.
- If you need encryption in transit within the cluster, you can use a service mesh or various cloud-native solutions. You can also enable encryption in transit outside the cluster by using an ingress controller.
- Private edition services support encryption of their data at rest as well secure connections to datastores using Transport Layer Security (TLS) protocol.
- Appropriate Center for Internet Security (CIS) benchmarks are generally built into services' container images and are applied to how Kubernetes node resources are accessed. Please see specific service documentation for limited exceptions and specific requirements.

Overrides

Genesys recognizes that your own stringent security requirements can differ from those that are enabled by default in Genesys Multicloud CX services. You can customize many of these security requirements by overriding Helm chart values, in accordance with the information in the appropriate service guide.

In that context, here are additional security requirements for you to consider as you set up your environment.

Pod security policies

Private edition does not support pod security policies.

Secrets

Secrets are namespace objects that contain a small amount of sensitive data, such as a password, a token, or a key. Most of the Genesys Multicloud CX services require secrets at deployment time, for dependencies, such as Postgres, Redis, email server, Genesys Cloud CX, and so on.

The scope of a secret is the namespace in which the secret is created. Unless you are using a single namespace for all private edition services, in each namespace you must create secrets for the third-party dependencies that are required by the service(s) in that namespace. If a secret is shared by different services in different namespaces, you must duplicate the secret in all the respective namespaces. Depending on how complex you want to make management of credentials for shared datastores and other shared dependencies, you can either replicate the same secret across multiple namespaces, so that different services use the same credentials for a given datastore, or create different secrets in each namespace, so that individual services use their own credentials for a given datastore.

You must use only Kubernetes secrets at runtime, and they must support user-supplied values and secrets via Helm-value overrides.

Data encryption through TLS/SSL

Genesys Multicloud CX services support TLS protocol for connections into the cluster up to the ingress controller. Data is not encrypted beyond the ingress controller.

Genesys Multicloud CX services support TLS protocol for connections to third-party dependencies based on the details and capabilities of those dependencies. The credentials associated with each connection are managed through secrets associated with the relevant services.

Voice connectivity

Contents

Learn about the private edition services involved in handling SIP and RTP traffic, including their connections within and outside the private edition deployment.

Related documentation:

-
-

RSS:

- For private edition

Introduction

For the Genesys Multicloud CX private edition services to receive and process voice interactions, you must enable voice connectivity.

Voice connectivity in a private edition deployment covers the following:

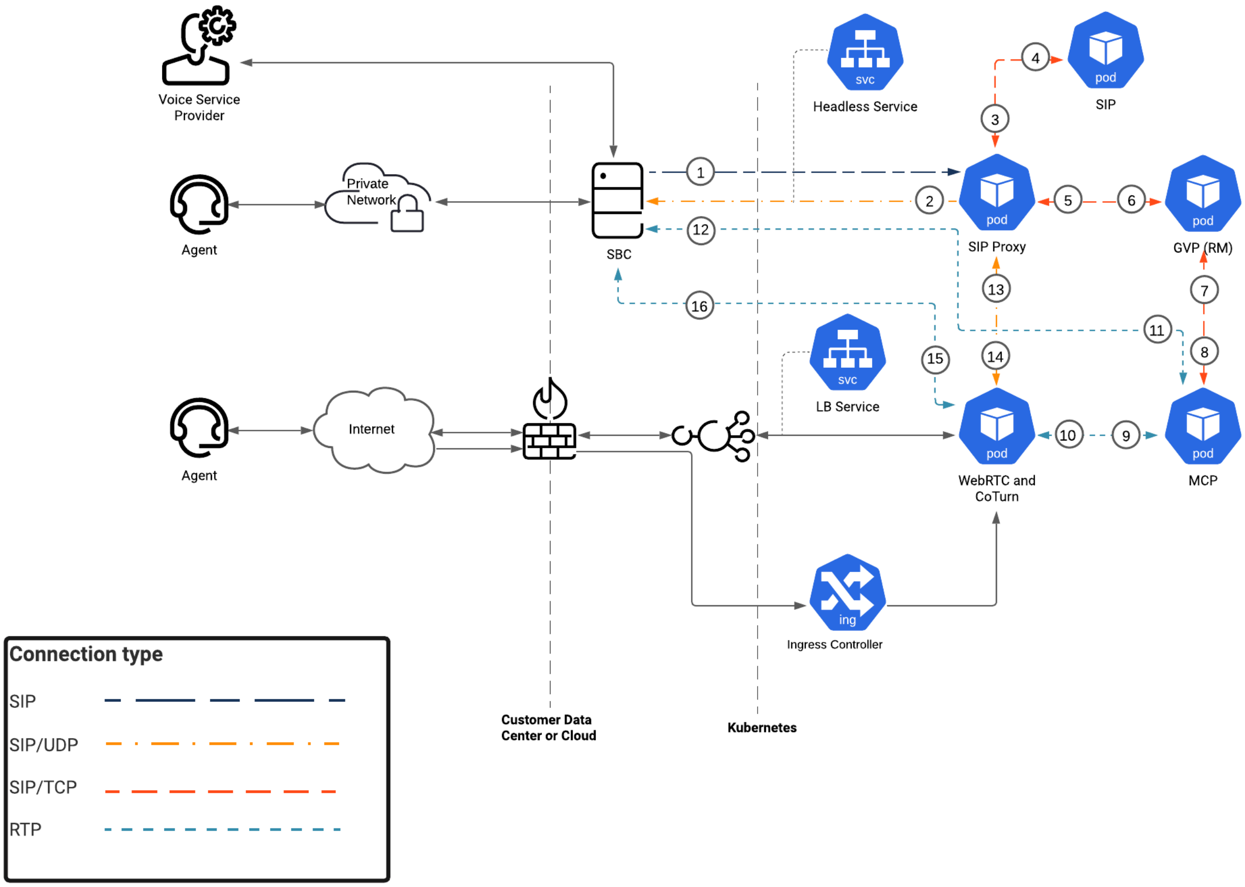
- Connectivity to and from Session Border Controller (SBC)
- Connectivity to and from agent-facing services (Agent Workspace, SIP phone, or Web phone)
- Connectivity to the private edition services involved in processing voice interactions:
  - Voice Microservices — in particular, Voice SIP Cluster Service and Voice SIP Proxy Service
  - Genesys Voice Platform (GVP) — in particular, GVP Media Control Platform (MCP)
  - Web Real-Time Communication (WebRTC) Media Service

For information about Genesys services' connections, see:

- Architecture for Voice Microservices in the *Voice Microservices Private Edition Guide*
- Architecture for Genesys Voice Platform in the *GVP Private Edition Guide*
- Architecture for WebRTC in the *WebRTC Private Edition Guide*

Connections

The following diagram shows the voice connections from the services running in Kubernetes to the other services.



The following table provides the network details of voice connections:

Connection	Client	Client network	Server	Server network	Protocol	Default port	Description
Not applicable	voice-sipproxy	Kubernetes Network	voice-config	Kubernetes Network	SIP/TCP	9100	Fetches tenant details
1	SBC	VNET Network	voice-sipproxy	Kubernetes Network	SIP	5080	SBC SIP signaling
2	voice-sipproxy	Kubernetes Network	SBC	VNET Network	SIP/UDP	5060	SBC SIP signaling
2	voice-sipproxy	Kubernetes Network	SBC (Cross-Region)	VNET Peering	SIP/UDP	5060	SBC SIP signaling
3	voice-sipproxy	Kubernetes Network	voice-sip	Kubernetes Network	SIP/TCP	5090	SIP signaling
4	voice-sip	Kubernetes Network	voice-sipproxy	Kubernetes Network	SIP/TCP	5080	SIP signaling
5	voice-	Kubernetes	gvp (RM)	Kubernetes	SIP/TCP	5060	IVR SIP

Connection	Client	Client network	Server	Server network	Protocol	Default port	Description
	sipproxy	Network		Network			signaling
6	gvp (RM)	Kubernetes Network	voice-sipproxy	Kubernetes Network	SIP/TCP	5080	GVP SIP signaling
7	gvp (RM)	Kubernetes Network	MCP	Kubernetes Network	SIP/TCP	5070	GVP SIP signaling
8	MCP	Kubernetes Network	gvp (RM)	Kubernetes Network	SIP/TCP	5080	GVP SIP signaling
9	MCP	Kubernetes Network	WebRTC/ CoTurn	Kubernetes Network	RTP	Negotiated	RTP Voice
10	WebRTC/ CoTurn	Kubernetes Network	MCP	Kubernetes Network	RTP	Negotiated	RTP Voice
11	MCP	Kubernetes Network	SBC	VNET Network	RTP	Negotiated	RTP Voice
12	SBC	VNET Network	MCP	Kubernetes Network	RTP	Negotiated	RTP Voice
13	voice-sipproxy	Kubernetes Network	WebRTC/ CoTurn	Kubernetes Network	SIP/UDP	5070	Agent SIP signaling
14	WebRTC/ CoTurn	Kubernetes Network	voice-sipproxy	Kubernetes Network	SIP/UDP	5080	Agent SIP signaling
15	WebRTC/ CoTurn	Kubernetes Network	SBC	VNET Network	RTP	Negotiated	RTP Voice
16	SBC	VNET Network	WebRTC/ CoTurn	Kubernetes Network	RTP	Negotiated	RTP Voice

SBC and private edition deployment integration

You must enable access to SBC to ensure the voice interactions pass through to the Genesys services.

Software requirements

Contents

Prerequisite software and third-party dependencies required for the Genesys Multicloud CX private edition environment.

Related documentation:

-
-

RSS:

- For private edition

This article covers the following sections:

- The prerequisites required for the private edition environment
- The third-party dependencies required for the Genesys Multicloud CX services.

You must first set up the private edition environment with the supported Kubernetes distribution, Helm, contact center components, and so on. In the Kubernetes clusters, deploy the third-party dependencies such as Consul, Redis, Kafka, and so on, that are necessary for the Genesys Multicloud CX services to function. Once you have the private edition environment with the required third-party dependencies deployed, you can proceed with deploying the Genesys Multicloud CX services.

Private edition general prerequisites

The private edition general prerequisites are:

- Domain Name System (DNS)
- Helm 3.0+
- Ingress Controller
    - NGINX Ingress Controller (Google Kubernetes Engine)
- JFrog Edge Artifactory account
- Kubernetes 1.25
- Kubernetes secrets
- Session Border Controller (SBC)
- Web Application Firewall (WAF) - optional, but recommended.

Licensing requirements

Genesys Multicloud CX private edition services (release 100.x and above) do not require any technical licenses or activation files for deployment or operation in any environment that contains only Genesys software. However, software or services provided by vendors other than Genesys might require licenses or activation files. Any licenses or activation files required for third-party software or services that are resold by Genesys and/or are embedded in Genesys services will be issued separately according to the terms outlined in your contract and services order. It is your responsibility to acquire licenses for software or services that you obtain from other vendors.

Third-party dependencies for Genesys Multicloud CX services

Genesys Multicloud CX services require specific third-party dependencies for its functioning, for example, Redis (an in-memory caching software). You can install these third-party dependencies in a different namespace or outside the cluster provided the namespace has direct network access to these services.

> **Important**
>
> Deploying and maintaining the third-party dependencies is your responsibility. For more information on your responsibilities and how Genesys supports the deployment process, see Understanding responsibilities.

See the table below for details about the Genesys supported third-party dependencies.

Name	Version	Purpose	Mandatory?	Private edition services
A container image registry and Helm chart repository		Used for downloading Genesys containers and Helm charts into the customer's repository to support a CI/CD pipeline. You can use any Docker OCI compliant registry.	Yes	All Genesys services
An SMTP relay		Facilitates email communications in an environment where GCXI reports or voicemails are sent as emails to contact center personnel. Genesys	No	• Genesys Customer Experience Insights • Voice Microservices

Name	Version	Purpose	Mandatory?	Private edition services
		recommends PostFix, but you can use any SMTP relay that supports standard mail libraries.		
Command Line Interface		The command line interface tools to log in and work with the Kubernetes clusters.	No	• Genesys Authentication • Genesys Web Services and Applications • AI Connector • Digital Channels • GIM Config Adapter • GIM • GIM Stream Processor • WebRTC Gateway Service
HTTPS certificates - cert-manager		Use with Let's Encrypt to provide free rotating TLS certificates for NGINX Ingress Controller.	Optional	• Genesys Authentication • Genesys Web Services and Applications • AI Connector • Digital Channels
HTTPS certificates - Let's Encrypt		Use with cert-manager to provide free rotating TLS certificates for NGINX Ingress Controller. **Note:** Let's Encrypt is a suite-wide requirement if you choose an Ingress Controller that needs it.	No	• Genesys Authentication • Genesys Web Services and Applications • AI Connector • Digital Channels
Ingress controller		HTTPS ingress controller.	Yes	• Genesys Authentication • Genesys Web Services and Applications • AI Connector

Name	Version	Purpose	Mandatory?	Private edition services
				• Digital Channels • Universal Contact Service
Load balancer		VPC ingress. For NGINX Ingress Controller, a single regional Google external network LB with a static IP and wildcard DNS entry will pass HTTPS traffic to NGINX Ingress Controller which will terminate SSL traffic and will be setup as part of the platform setup.	Yes	• Genesys Authentication • Designer • Genesys Web Services and Applications • AI Connector • Digital Channels • Intelligent Workload Distribution • CX Contact • Genesys Customer Experience Insights • Genesys Pulse • Universal Contact Service • WebRTC Gateway Service
Object storage		Persistent or shared data storage, such as Amazon S3, Azure Blob Storage, or Google Cloud Storage.	No	• GIM Config Adapter • GIM • GIM Stream Processor
Kafka	2.x	Message bus.	Yes	• Interaction Server • GIM Config Adapter • GIM • GIM Stream Processor • Tenant Service • Event Stream • Voice Microservices
Keda	2.0	Custom metrics for scaling. Use of Keda or HPA is configurable through Helm charts.	No	• WebRTC Gateway Service

Name	Version	Purpose	Mandatory?	Private edition services
Redis	6.x	Used for caching. Only distributions of Redis that support Redis cluster mode are supported, however, some services may not support cluster mode.	Yes	• Genesys Authentication • Designer • Genesys Web Services and Applications • Interaction Server • Genesys Engagement Service • AI Connector • Digital Channels • Email • Intelligent Workload Distribution • CX Contact • Genesys Pulse • Tenant Service • Event Stream • Voice Microservices
Consul	1.13.x	Service discovery, service mesh, and key/value store.	Yes	• Genesys Authentication • Genesys Web Services and Applications • Genesys Engagement Service • Tenant Service • Voice Microservices
Elasticsearch	7.x	Used for text searching and indexing. Deployed per service that needs Elasticsearch during runtime.	Yes	• Designer • Genesys Web Services and Applications • Intelligent Workload Distribution • CX Contact • Universal Contact Service
MS SQL Server	2016 or later	Relational database. Required only		• Genesys Voice Platform

Name	Version	Purpose	Mandatory?	Private edition services
		for GVP.		
PostgreSQL	11.x	Relational database.	Yes	• Genesys Authentication • Genesys Voice Platform • Genesys Web Services and Applications • Interaction Server • Genesys Engagement Service • AI Connector • Digital Channels • IWD Data Mart • Intelligent Workload Distribution • CX Contact • GIM • Genesys Pulse • Tenant Service • Universal Contact Service

For information on troubleshooting third-party services, refer to Troubleshooting Third-Party Services in our public repository.

Permissions

Security context parameters in the Helm charts specify the users authorized to access the pods and containers for the respective services. By default, the Helm charts specify the user, group, and file-service group IDs as `500:500:500`.

Consul

- Consul and Consul Service Mesh are required.
- Consul requires privileged containers; so the cluster-administrator must have permissions to install mutating hooks, configure kube-dns, and access Kubernetes APIs.

In an early implementation, private edition required the use of a custom SCC called **genesys-restricted** to control permissions associated with the **genesys** user (500) specified by the services. The **genesys-restricted** SCC has now been deprecated.

Arbitrary UIDs

To use arbitrary UIDs, override the Helm chart values so that no specific IDs are defined for users and groups.

Storage requirements

Contents

Provides information about different storage types required for Genesys Multicloud CX services.

Related documentation:

-
-

RSS:

- For private edition

Deciding storage includes a lot of factors such as the number of agents, call volumes, call recordings and archiving them, data security, accessibility, and so on. It also includes technical factors such as the Input Output Per Second (IOPS) or throughput, storage type, latency, and so on.

In Genesys Multicloud CX private edition, you will create storage for specific services, for example, Genesys Customer Experience Insights (GCXI), and Voice. The services that require storage elements, such as file and disk storage for processing its data, use the Kubernetes Persistence Volume subsystem (PV). The storage subsystem and Kubernetes StorageClass types requirements for different services for different Kubernetes platforms are given in the following tables:

- File and disk storage for Azure Kubernetes Service (AKS)
- File and disk storage for Google Kubernetes Engine (GKE)

You can create or select the storage subsystem for your service on a specific Kubernetes platform based on the information presented in the corresponding table. For the exact sizing of each storage subsystem or PVs, refer to the related service-level documentation.

Important

By default, the Kubernetes platform creates default file and disk storage classes. However, Genesys recommends not to use them but to create a custom file and disk storage for your service.

Tip

You can determine the storage requirements for your contact center yourself by either exploring the storage requirements of each service, by using the Sizing Calculator or by leveraging the Genesys Professional Services team's support.

File and disk storage for AKS

The following table provides the storage information for AKS:

AKS Storage Class Name#	Storage Type	Notes	Associated Services
disk-hdd (ephemeral)	Standard_HDD	Node disk mounted via HostPath.	• GCXI • Gplus WFM • GVP-MCP • GVP-RM • Interaction Server • Pulse • Tenant • Voice Services • WebRTC
disk-standard disk-premium	Azure Disk - Standard Azure Disk - Premium	Use single AZ disks to create an RWO volume that can be attached to a single pod.	• CX Contact • Designer • GVP • GWS • UCSX
files-standard	Azure Files - Standard Fileshare LRS	Local redundant storage (LRS) for RWX volumes that can be shared between multiple pod instances; replicated data in a single AZ. Lower throughput than premium and no IOPs guaranteed.	BDS
files-standard-redundant	Azure Files - Premium Fileshare ZRS	Zonal redundant storage (ZRS) for RWX volumes shared across multiple pods; replicated data across multiple AZs in a region. No IOPS guaranteed - similar to NFS.	• CX Contact • Designer • GCXI • Gplus WFM • GVP • GWS

AKS Storage Class Name#	Storage Type	Notes	Associated Services
			• Pulse • Tenant • UCSX • WebRTC
blob storage	Azure Blob Storage	Create Azure Blob Storage which is optimized for storing massive amounts of unstructured data across AZ and regions.	• Digital channels (image, files, upload) • GIM data feed/GSP • Recordings (GVP) • Telemetry • Voicemail

- *#The AKS storage class names are created by default. You can modify the storage class names based on your organizational needs.*

File and disk storage for GKE

The following table provides the storage information for GKE:

GKE Storage Class Name#	Storage Type	Notes	Associated Services
ephemeral (emptyDir)	Persistent disk	Node disk accessed through local ephemeral **emptyDir** volumes, provided there is no access to hostPath.	• GCXI • GVP-MCP • GVP-RM • Gplus-WFM • Interaction Server • Pulse • Tenant • Voice services • WebRTC
standard-rwo*	pd-balanced (SSD)	Persistent Disk (pd) - Default Zonal (single AZ) RWO	• CX Contact

GKE Storage Class Name#	Storage Type	Notes	Associated Services
premium-rwo*	pd-ssd (SSD)	StorageClasses provided by GKE with typical Block storage performance.	• Designer • GVP • GWS • UCSX
standard-rwx**	Filestore - Basic HDD	Local redundant storage for RWX volumes shared between pod instances; replicated data in a single AZ.	BDS
redundant-rwx**	Filestore - Enterprise	Regional redundant storage for RWX volumes shared between pod instances; replicated data to two zones in a region (Regional PD).	• CX Contact • Designer • GCXI • Gplus-WFM • GVP • GWS • Pulse • Tenant • UCSX • WebRTC
blob storage	Cloud Storage buckets	Create Google Cloud Storage which is optimized for storing massive amounts of unstructured data across AZ and regions.	• Digital channels (image, files, upload) • GIM data feed/GSP • Recordings (GVP) • Telemetry • Voicemail

- *#The GKE storage class names are created by default. You can modify the storage class names based on your organizational needs.*
- **RWO type storage is tested with the default CSI driver.*
- **RWX type storage is tested with the Filestore CSI driver. This storage driver is not enabled by default and it must be enabled in the GKE clusters. However, this configuration is available only in GKE 1.21.x releases. For more information on enabling Filestore CSI driver, see GKE documentation

Communication ports and protocols

Contents

Provides information on the ports required for different services. Also provides the communication protocols supported between Genesys Multicloud CX services and between Genesys Multicloud CX services and other external systems in the cloud private edition infrastructure.

Related documentation:

-
-

RSS:

- For private edition

Ports and protocols

Genesys Multicloud CX services require you to open specific ports in your cloud private edition environment. If your corporate network policy prevents access from external systems (or other clusters) to clusters that run Genesys Multicloud CX services, alter your network policy to allow appropriate access.

The following table presents the consolidated view of ports that different Genesys Multicloud CX services and third-party dependencies use. For more information about its configuration, see the related service-level guides.

List by service

View CSV: Download

Service	Protocol	Port
CX Contact	HTTP	3004-3008
	HTTP/HTTPS	443
	RTP/RTCP	
	TCP	20, 21, 22
		2049
		5050
		5432
		6379
		8888

Service	Protocol	Port
Designer	HTTP	443
		6380
		80
		8080
		8888
		9205
	HTTP/HTTPS	80
	HTTPS	443
		8095
Digital Channels	HTTP	80
	HTTP/CometD	
	HTTP/WS	
	HTTPS	443
	HTTPS/WSS	
Genesys Authentication	HTTP/HTTPS	80/443
	HTTPS	443
	TCP	5432
		6379 (non SSL) or 6380 (SSL)
		8888
		9200
Genesys Customer Experience Insights	HTTP	80
		8080
		9101
	HTTPS	443
	TCP	34952
		5432
		Logical connection only
Genesys Engagement Service	HTTP	3050
		5580
		8091
		8092
		8095
		9098
	HTTPS	443
	Postgres	5432
	Redis	6379
Genesys Info Mart	HTTP	443

Service	Protocol	Port
		8249
		9249
	HTTPS	443
	Kafka	9092
	SSL	5432
	TCP	
Genesys Pulse	HTTP	80
		8080
		8090
		9091
	HTTPS	443
	TCP	2060
		5432
		6380
		7120
		7122
		8000
		8888
Genesys Voice Platform	HTTP	11200
		443
		80
		8080
		8200
		8300
		9090
		9116
	HTTP/HTTPS	11200
		80
	RTP/RTCP	20000-45000
		20000-45000/14000-15999
	SIP/TCP	5060
		5070
		5090
	TCP	1433
		1705
		5432
		61616

Service	Protocol	Port
		61616 / 8080
		8500/8501
		8888
		9801
Genesys Web Services and Applications	HTTP	80
		8500
	HTTPS	443
	TCP	5432
		6379
		9200
Intelligent Workload Distribution	HTTP	80
	HTTPS	25, 443, 587, 993
		443
	TCP	10052
		4024
		5432
		6379
		80
		9200
Interaction Server	HTTP	13131
		13133
		13139
		8888
	TCP	2060
		7120
		7122
		8500
		8888
Telemetry Service	HTTP	80
		8107
		9107
	HTTPS	443
Tenant Service	HTTP	15000
		5580
	TCP	2060
		5050
		5432

Service	Protocol	Port
		6379
		7120
		8000
		8888
		9092/9093
Universal Contact Service	HTTP	443
		80
		8080
	TCP	10052
		443
		5432
		6432
		80
		8080
		9200
Workspace Web Edition	HTTP	8080
	HTTPS	443

List by protocol

View CSV: Download

Protocol	Port	Service
HTTP	11200	Genesys Voice Platform
	13131	Interaction Server
	13133	
	13139	
	15000	Tenant Service
	3004-3008	CX Contact
	3050	Genesys Engagement Service
	443	CX Contact
		Designer
		Genesys Info Mart
		Genesys Voice Platform
		Universal Contact Service
	5580	Genesys Engagement Service
		Tenant Service

Protocol	Port	Service
	6380	Designer
	80	Digital Channels
		Genesys Customer Experience Insights
		Genesys Pulse
		Genesys Voice Platform
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Telemetry Service
		Universal Contact Service
	8080	Designer
		Genesys Customer Experience Insights
		Genesys Pulse
		Genesys Voice Platform
		Universal Contact Service
		Workspace Web Edition
	8090	Genesys Pulse
	8091	Genesys Engagement Service
	8092	
	8095	
	8107	Telemetry Service
	8200	Genesys Voice Platform
	8249	Genesys Info Mart
	8300	Genesys Voice Platform
	8500	Genesys Web Services and Applications
	8888	Designer
		Interaction Server
	9090	Genesys Voice Platform
	9091	Genesys Pulse
	9098	Genesys Engagement Service
	9101	Genesys Customer Experience Insights
	9107	Telemetry Service
	9116	Genesys Voice Platform
	9205	Designer
	9249	Genesys Info Mart

Protocol	Port	Service
HTTP/CometD	80	Digital Channels
HTTP/HTTPS	11200	Genesys Voice Platform
	443	CX Contact
	80	Designer
		Genesys Voice Platform
	80/443	Genesys Authentication
HTTP/WS	80	Digital Channels
HTTPS	25, 443, 587, 993	Intelligent Workload Distribution
	443	Designer
		Digital Channels
		Genesys Authentication
		Genesys Customer Experience Insights
		Genesys Engagement Service
		Genesys Info Mart
		Genesys Pulse
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Telemetry Service
		Workspace Web Edition
	8095	Designer
HTTPS/WSS	443	Digital Channels
Kafka	9092	Genesys Info Mart
Postgres	5432	Genesys Engagement Service
Redis	6379	
RTP/RTCP	20000-45000	Genesys Voice Platform
	20000-45000/14000-15999	
	443	CX Contact
SIP/TCP	5060	Genesys Voice Platform
	5070	
	5090	
SSL	5432	Genesys Info Mart
TCP	10052	Intelligent Workload Distribution
		Universal Contact Service
	1433	Genesys Voice Platform
	1705	
	20, 21, 22	CX Contact

Protocol	Port	Service
	2049	
	2060	Genesys Pulse
		Interaction Server
		Tenant Service
	34952	Genesys Customer Experience Insights
	4024	Intelligent Workload Distribution
	443	Universal Contact Service
	5050	CX Contact
		Tenant Service
	5432	CX Contact
		Genesys Authentication
		Genesys Customer Experience Insights
		Genesys Info Mart
		Genesys Pulse
		Genesys Voice Platform
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Tenant Service
		Universal Contact Service
	61616	Genesys Voice Platform
	61616 / 8080	
	6379	CX Contact
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Tenant Service
	6379 (non SSL) or 6380 (SSL)	Genesys Authentication
	6380	Genesys Pulse
	6432	Universal Contact Service
	7120	Genesys Pulse
		Interaction Server
		Tenant Service
	7122	Genesys Pulse
		Interaction Server
	80	Intelligent Workload Distribution
		Universal Contact Service

Protocol	Port	Service
	8000	Genesys Pulse
		Tenant Service
	8080	Universal Contact Service
	8500	Interaction Server
	8500/8501	Genesys Voice Platform
	8888	CX Contact
		Genesys Authentication
		Genesys Pulse
		Genesys Voice Platform
		Interaction Server
	9092/9093	Tenant Service
	9200	Genesys Authentication
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Universal Contact Service
	9801	Genesys Voice Platform
	Logical connection only	Genesys Customer Experience Insights

List by port

View CSV: Download

Port	Protocol	Service
10052	TCP	Intelligent Workload Distribution
		Universal Contact Service
11200	HTTP	Genesys Voice Platform
	HTTP/HTTPS	
13131	HTTP	Interaction Server
13133		
13139		
1433	TCP	Genesys Voice Platform
15000	HTTP	Tenant Service
1705	TCP	Genesys Voice Platform
20, 21, 22		CX Contact
20000-45000	RTP/RTCP	Genesys Voice Platform

Port	Protocol	Service
20000-45000/14000-15999		
2049	TCP	CX Contact
2060		Genesys Pulse
		Interaction Server
		Tenant Service
25, 443, 587, 993	HTTPS	Intelligent Workload Distribution
3004-3008	HTTP	CX Contact
3050		Genesys Engagement Service
34952	TCP	Genesys Customer Experience Insights
4024		Intelligent Workload Distribution
443	HTTP	CX Contact
		Designer
		Genesys Info Mart
		Genesys Voice Platform
		Universal Contact Service
	HTTP/HTTPS	CX Contact
	HTTPS	Designer
		Digital Channels
		Genesys Authentication
		Genesys Customer Experience Insights
		Genesys Engagement Service
		Genesys Info Mart
		Genesys Pulse
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Telemetry Service
		Workspace Web Edition
	HTTPS/WSS	Digital Channels
	RTP/RTCP	CX Contact
	TCP	Universal Contact Service
5050		CX Contact
		Tenant Service
5060	SIP/TCP	Genesys Voice Platform
5070		
5090		

Port	Protocol	Service
5432	Postgres	Genesys Engagement Service
	SSL	Genesys Info Mart
	TCP	CX Contact
		Genesys Authentication
		Genesys Customer Experience Insights
		Genesys Info Mart
		Genesys Pulse
		Genesys Voice Platform
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Tenant Service
		Universal Contact Service
5580	HTTP	Genesys Engagement Service
		Tenant Service
61616	TCP	Genesys Voice Platform
61616 / 8080		
6379	Redis	Genesys Engagement Service
	TCP	CX Contact
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Tenant Service
6379 (non SSL) or 6380 (SSL)		Genesys Authentication
6380	HTTP	Designer
	TCP	Genesys Pulse
6432		Universal Contact Service
7120		Genesys Pulse
		Interaction Server
		Tenant Service
7122		Genesys Pulse
		Interaction Server
80	HTTP	Designer
		Digital Channels
		Genesys Customer Experience Insights
		Genesys Pulse
		Genesys Voice Platform

Port	Protocol	Service
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Telemetry Service
		Universal Contact Service
	HTTP/CometD	Digital Channels
	HTTP/HTTPS	Designer
		Genesys Voice Platform
	HTTP/WS	Digital Channels
	TCP	Intelligent Workload Distribution
		Universal Contact Service
80/443	HTTP/HTTPS	Genesys Authentication
8000	TCP	Genesys Pulse
		Tenant Service
8080	HTTP	Designer
		Genesys Customer Experience Insights
		Genesys Pulse
		Genesys Voice Platform
		Universal Contact Service
		Workspace Web Edition
	TCP	Universal Contact Service
8090	HTTP	Genesys Pulse
8091		Genesys Engagement Service
8092		
8095	HTTPS	Designer
8107	HTTP	Telemetry Service
8200		Genesys Voice Platform
8249		Genesys Info Mart
8300		Genesys Voice Platform
8500		Genesys Web Services and Applications
	TCP	Interaction Server
8500/8501		Genesys Voice Platform
8888	HTTP	Designer
		Interaction Server
	TCP	CX Contact
		Genesys Authentication

Port	Protocol	Service
		Genesys Pulse
		Genesys Voice Platform
		Interaction Server
		Tenant Service
9090	HTTP	Genesys Voice Platform
9091		Genesys Pulse
9092	Kafka	Genesys Info Mart
9092/9093	TCP	Tenant Service
9098	HTTP	Genesys Engagement Service
9101		Genesys Customer Experience Insights
9107		Telemetry Service
9116		Genesys Voice Platform
9200	TCP	Genesys Authentication
		Genesys Web Services and Applications
		Intelligent Workload Distribution
		Universal Contact Service
9205	HTTP	Designer
9249		Genesys Info Mart
9801	TCP	Genesys Voice Platform
Logical connection only		Genesys Customer Experience Insights

Service level connection tables

Service	Link
CX Contact	Connections Table
Designer	Connections Table
Digital Channels	Connections Table
Event Stream	Connections Table
Genesys Authentication	Connections Table
Genesys Customer Experience Insights	Connections Table
Genesys Engagement Service	Connections Table
Genesys Info Mart	Connections Table
Genesys Pulse	Connections Table
Genesys Voice Platform	Connections Table

Service	Link
Genesys Voice Platform	Configuration Server Connections Table
Genesys Voice Platform	Media Control Platform Connections Table
Genesys Voice Platform	Reporting Server Connections Table
Genesys Voice Platform	Resource Manager Connections Table
Genesys Voice Platform	Service Discovery Connections Table
Genesys Web Services and Applications	Connections Table
Intelligent Workload Distribution	Connections Table
Interaction Server	Connections Table
Telemetry Service	Connections Table
Tenant Service	Connections Table
Universal Contact Service	Connections Table
Voice Microservices	Cross-region Connections Table
Workspace Web Edition	Connections Table

Understanding responsibilities

Learn about the division of responsibilities between Genesys and your organization in deploying Genesys Multicloud CX private edition.

Related documentation:

-
-

RSS:

- For private edition

Genesys Multicloud CX services are containerized and delivered to your organization through the JFrog Artifactory Edge repository. Genesys ensures that the containers can run on infrastructure such as a public or private cloud (refer to the release notes for the complete list of supported platforms), or bare-metal servers that reside within your corporate data center. However, it is your responsibility to set up the infrastructure that is suitable for deploying Genesys Multicloud CX services, including deployment of Genesys-recommended third-party prerequisites in your clusters.

After you set up the clusters and third-party prerequisites, you can proceed with deploying Genesys Multicloud CX services.

The following table presents the responsibilities between Genesys and the organization(s) responsible for deploying Genesys Multicloud CX private edition.

Organization(s) deploying private edition	Genesys
Deploys and manages your Kubernetes technology stack.	Provides containerized software that supports multiple Kubernetes-certified environments. Provides sizing estimates to determine the cluster size, number of nodes, and so on, to set up the infrastructure.
Deploys and manages all required third-party services such as PostgreSQL, Redis, Elasticsearch, and so on. Check Software requirements for more information.	Provides documentation that lists the recommended third-party services and their supported versions. Provides guidance or the need for any Genesys-specific configuration in the third-party services.
Provides environment-specific configuration values to override the default values.	Ensures that Genesys Multicloud CX services and their Helm charts are accessible to your organization through the JFrog Artifactory Edge repository. Provides the ability to override the Helm

Organization(s) deploying private edition	Genesys
	charts with environment-specific values.
Provides the network infrastructure with required access to manage voice and data traffic.	Provides network requirements information such as Ingress controller, load balancers, and so on for each service.
Ensures security of the infrastructure by implementing security protocols.	Ensures that Genesys-provided container images and artifacts enable your organization to implement your security policies and guidelines, based on industry best practices and security standards.
Configures the preferred logging software to capture Genesys Multicloud CX services logs.	Provides support to standard out/standard error logging, which enables your organization to use popular logging software such as Fluentd to collect and analyze log data. Explains the configuration procedure with Fluentd as an example.
Configures the preferred monitoring software to capture Genesys Multicloud CX services metrics.	Provides support to popular monitoring software such as Prometheus, to monitor your operations using metrics provided by each service. Explains the configuration procedure with Prometheus as an example.
Manages upgrades by setting a continuous deployment (CD) pipeline and performs timely deployment testing.	Provides new software updates through the JFrog Artifactory Edge repository, which enables you to perform in-service upgrades using the documented steps a CD pipeline must implement. Also provides a sample procedure that helps to set up a CD pipeline in your environment.

Quick deployment tour

Contents

Provides an overview of the overall deployment process.

Related documentation:

-
-

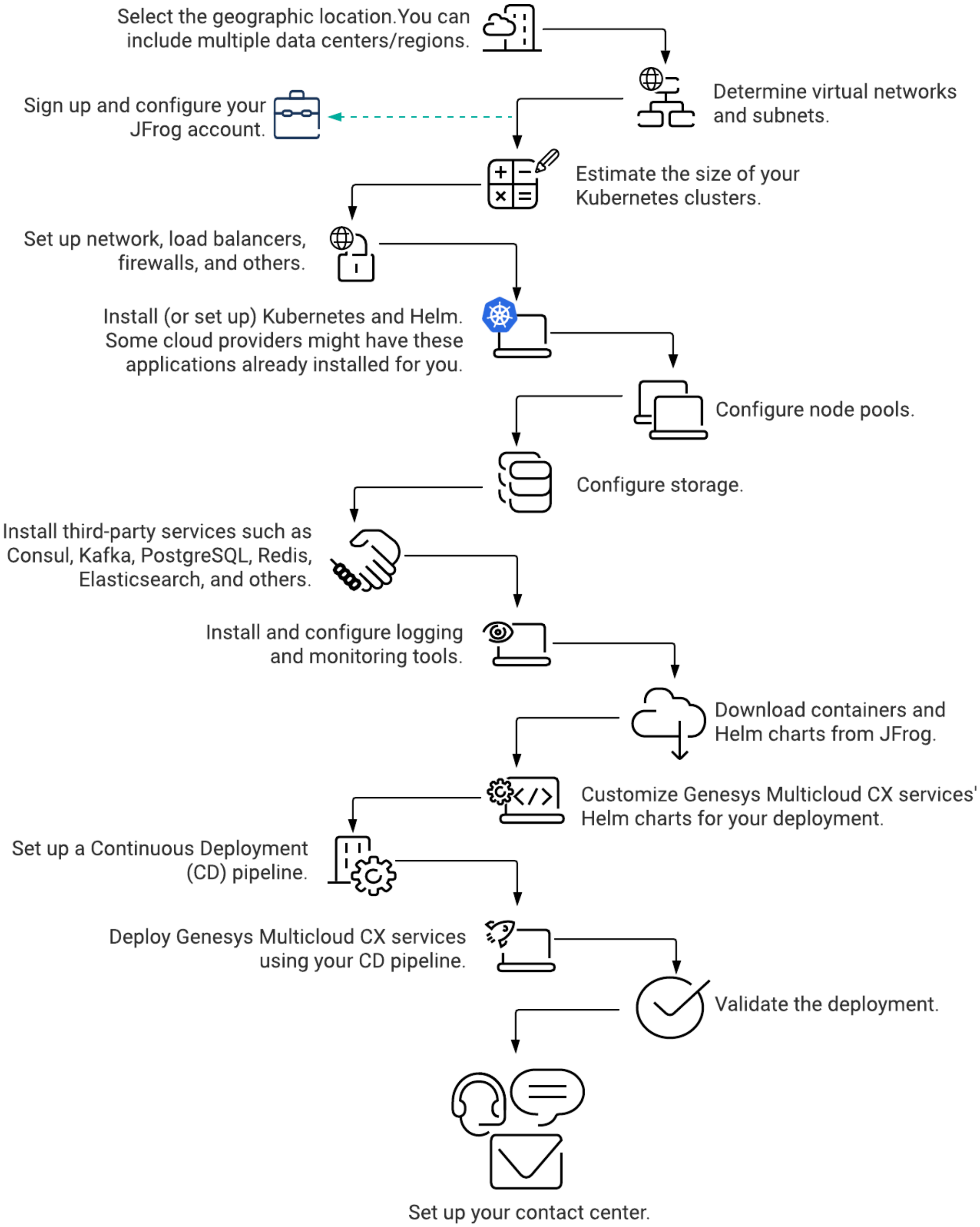
RSS:

- For private edition

Genesys Multicloud CX Private Edition Deployment overview

The following picture takes you through a quick tour of the steps involved in deploying Genesys Multicloud CX private edition. See the process table below the image for links to the relevant topics.

> **Important**
>
> You must follow the same steps (as shown in the following picture) for setting up the cloud private edition infrastructure in different locations, for example, US West and US East. Repeat the same steps for setting up different environments such as pre-production, production, and so on.


Select the geographic location.You can include multiple data centers/regions.
Determine virtual networks and subnets.
Sign up and configure your JFrog account.
Estimate the size of your Kubernetes clusters.
Set up network, load balancers, firewalls, and others.
Install (or set up) Kubernetes and Helm. Some cloud providers might have these applications already installed for you.
Configure node pools.
Configure storage.
Install third-party services such as Consul, Kafka, PostgreSQL, Redis, Elasticsearch, and others.
Install and configure logging and monitoring tools.
Download containers and Helm charts from JFrog.
Customize Genesys Multicloud CX services' Helm charts for your deployment.
Set up a Continuous Deployment (CD) pipeline.
Deploy Genesys Multicloud CX services using your CD pipeline.
Validate the deployment.
Set up your contact center.

Process	Related topics
Select the geographic location	Architecture
Determine virtual networks and subnets	Networking overview
Set up network, load balancers, firewalls and others	Voice connectivity
Configure node pools	Architecture
Configure storage	Storage requirements
Install third-party services	Software requirements
Install and configure logging and monitoring tools	Configuring logging Configuring monitoring
Download containers and Helm charts from JFrog	Downloading your Genesys Multicloud CX containers
Customize Genesys Multicloud CX services' Helm charts for your deployments	Overriding Helm chart values
Set up a Continuous Deployment (CD) pipeline	Setting up a CD pipeline
Deploy Genesys Multicloud CX services using your CD pipeline	Genesys Multicloud CX private edition services

Important

In addition to the content available in this guide, supplemental technical reference information is also available in our public repository. You can access it from here. For easy navigation, we have also linked to it from other applicable sections in this guide.

Network settings

Contents

Describes the network settings required for Kubernetes clusters in Genesys Multicloud CX private edition. For more information about networking outside Kubernetes clusters, see Networking overview.

Related documentation:

-
-

RSS:

- For private edition

Enabling Container Networking Interface

In your Kubernetes cluster, enable Container Networking Interface (CNI) or its equivalent to establish communication between pods in the cluster.

Configuring Ingress Controller

You must set up an ingress controller to manage all the HTTP and WebSocket ingress traffic, and to support Cluster IP. The ingress controller you choose must have the following properties:

- Cookies usage
- Header requirements - client IP and redirect, and passthrough
- Session stickiness
- Allowlisting (optional)
- TLS for ingress (optional) - ability to enable or disable TLS on the connection.

You can define these parameters in the **values.yaml** file for applicable services. For more information, see the related service-level guides.

DNS and Service Mesh

DNS

Genesys recommends having a CoreDNS within the Kubernetes clusters along with Node LocalDNS for performance.

Service Mesh

Genesys Multicloud CX services require Consul Service Mesh that dynamically routes traffic to the right available service instance. Deploy Consul Service Mesh within the cluster where Genesys Multicloud CX services are deployed.

Network Policy

Genesys does not supply or enforce any network policy. You can create your own network policy for services that require a network policy and configure them in the Helm v3 charts.

For more information about network policy requirements, see the related service-level guides.

Creating namespaces

Contents

Recommendations in creating namespaces for Genesys Multicloud CX services deployment.

Related documentation:

-
-

RSS:

- For private edition

Namespaces for Genesys Multicloud CX services

A namespace provides a virtual cluster for applying access control policies and setting the scope of named resources such as internal DNS names, pods, services, deployments, and constraints for resource quotas.

For deploying Genesys Multicloud CX services, Genesys requires you to create a namespace per service group and deploy the associated services within that namespace.

Create the namespaces using the naming conventions given in the following table and defined in the Helm charts of the respective services. Note that most of the service groups contain several microservices.

The naming conventions meet Kubernetes requirements that the names of namespaces within a cluster must be unique. Note that, as described under Kubernetes clusters, you must use separate Kubernetes clusters if you want to deploy private edition instances in separate environments for testing, staging, production, and so on.

> **Important**
>
> Make sure that you follow the naming conventions of the namespaces as given in the following table.

For more information, refer to the service guides of the individual services you are deploying.

Service Group	Name
Designer	designer
Genesys Web Services (GWS/GAPI)	gws
Genesys Engagement Service (Callback and Mobile)	ges
Historical Reporting Back-end (GIM)	gim,gca,gsp

Historical Reporting Front-end (GCXI)	gcxi
Realtime Reporting	pulse
Digital/Nexus	nexus
Digital-Legacy (Ixn Server)	ixn
UCS-X	ucsx
IWD	iwd - plus 2 additional namespaces - iwddm, iwdem
CX-Contact	cxc
GVP	gvp
WebRTC	webrtc
Voice Microservices	voice
Voice Tenants	voice
Voice Legacy (Config, Stat Server, URS, OCS)	voice
WFM 3rd party Connector	gluswfm
Telemetry	tlm
BDS (Billing)	bds
Genesys Authentication services	gauth

Namespace for third-party services

You can create a different namespace for installing the backend infrastructure services like Redis, PostgreSQL, etc. as long as the Genesys Multicloud CX service deployments have the required network access and the services have resolvable DNS names. The best way to manage your backend infrastructure services and Genesys Multicloud CX services is to decouple and deploy them in different clusters.

Configuring logging

Contents

Provides an overview of logging architecture in Genesys Multicloud CX private edition, different types of logging mechanisms, and related configurations.

Related documentation:

-
-

RSS:

- For private edition

Logging approaches and configuration

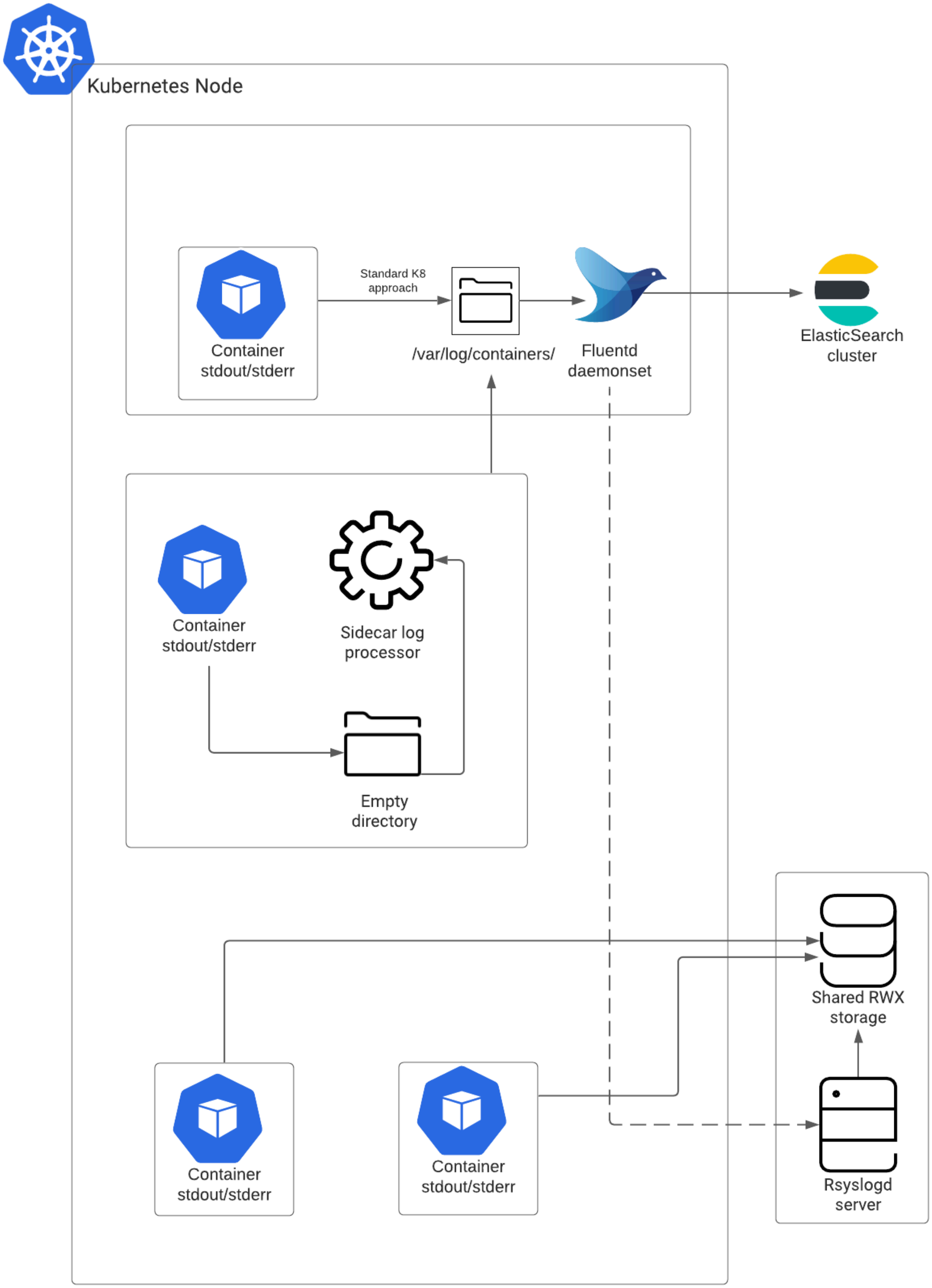
This section explains the approaches of logging used by Genesys Multicloud CX services to write log files that contain the important diagnostic information for various issues that may arise. Support of Genesys services rely on access to these application logs.

For more details, refer to Solution-level logging approaches.

Logging architecture

This section explains the logging architecture of Genesys Multicloud CX private edition in detail.

Let's explore the logging architecture, the components involved, and its functionality through the following diagram.


Kubernetes Node
Container stdout/stderr
Standard K8 approach
/var/log/containers/
Fluentd daemonset
ElasticSearch cluster
Container stdout/stderr
Sidecar log processor
Empty directory
Shared RWX storage
Rsyslogd server
Container stdout/stderr
Container stdout/stderr

Logging architecture components

Elasticsearch cluster

Elasticsearch cluster deployed on multiple node aggregates the structured logs from Fluentd and indexes them. This includes the logs from services that follow Secondary and Complementary logging methods. You can use a log visualizer tool like Kibana to view, search, or filter the indexed logs from Elasticsearch.

Fluentd / Fluent-bit

Fluentd is a log collector commonly used with container platforms. It collects logs from the cluster and forwards them to Elasticsearch or an externally accessible storage such as Rsyslog server or both depending on your configuration. Fluentd /Fluent-bit collects the application logs of Genesys Multicloud CX services from **/var/log/containers**. While deploying cluster wide logging each node Fluentd /Fluent-bit will be deployed to each node.

Shared RWX storage

The unstructured logs are directly written in the RWX shared storage. For services writing unstructured logs, you must mount PVC/PV. To access logs externally, use a server like NFS or S3.

Syslog server storage

Optionally, you can implement a syslog server to store the structured logs other than the Elasticsearch log store. Syslog server writes the logs in a flat file and enables you to share them externally. Genesys recommends Rsyslog server for this purpose, however you can select any syslog server of your choice. For more information, refer the deployment procedure.

Configuring monitoring

Contents

Provides an overview of monitoring architecture in Genesys Multicloud CX private edition, different metrics collected, and related configurations.

Related documentation:

-
-

RSS:

- For private edition

Monitoring approach and configuration

This section provides information regarding metrics, alerts, and the monitoring approach for services. Services provide the necessary interface to use your own monitoring and logging tools, Prometheus-based metrics, and the endpoint that the Prometheus platform can scrape for alerting and monitoring. You must enable Prometheus to scrape user workload. Once enabled, Prometheus scrapes all metrics from endpoints exposed by services.

Some services optionally use Pushgateway to push metrics from jobs that cannot be scraped.

Refer to the following sections for more details about monitoring tools, metrics, handling alerts and Grafana configuration:

- Monitoring overview and approach
- Understanding GKE monitoring
- Enabling monitoring in GKE Platform
- System metrics
- Handling alerts
- Grafana configuration
- Monitoring Dashboards API

Order of services deployment

Contents

Learn about the order you must follow to deploy Genesys Multicloud CX services.

Related documentation:

-
-

RSS:

- For private edition

Deployment order of Genesys Multicloud CX services

There are many dependencies between Genesys Multicloud CX services. Therefore, certain services must be deployed in a specific sequence. And some services can be deployed in parallel with other services or concurrently with other services. For example, Genesys Web Services (GWS) and its components depends on Genesys Authentication services (GAuth) for authentication purposes. Hence, GAuth service must be deployed before GWS service.

Genesys Multicloud CX services must be deployed in the following order:

1. Deploy Consul and Kafka. Note that Consul and Kafka must deployed as part of the cloud private edition infrastructure.
2. Genesys authentication service (GAuth).
3. Microservices pertaining to Voice service.
4. Tenant service.
5. Agent Setup, Genesys Web Services (GWS), Workspace Web Edition (WWE), and WebRTC.
6. Genesys Voice Platform (GVP) service.
7. GIM Stream Processor (GSP), GIM Config Adapter (GCA), Genesys Info Mart (GIM), Designer, Universal Contact Service (UCS), Intelligent Workload Distribution (IWD), Telemetry, Nexus, CX Contact, Genesys Engagement Service (GES), and Pulse.
8. Interaction Server and IWD Datamart (IWDDM).
9. Genesys Customer Experience Insights (GCXI), and Gplus Adapter for Workforce Management (Gplus WFM).

Downloading your Genesys Multicloud CX containers

Contents

Genesys Multicloud CX containers are accessible through JFrog. You can also automate downloads to set up a Continuous Delivery (CD) pipeline.

Related documentation:

-
-

RSS:

- For private edition

Overview

Important

Use the information on this page to set up and manage your own repository. You can replicate the Genesys repository at your end. Do not pull images directly to your production environment. Refer to Setting up a CD pipeline for more information on managing your own repository.

Genesys uses JFrog to deliver and distribute its release containers. JFrog is a hybrid, universal, end-to-end devops platform. It is a fully automated platform for distributing software releases from code to production. You can pull new releases from the JFrog Artifactory Edge repository.

Warning

Genesys Engage will deprecate the JFrog Edge on June 15, 2024. Please contact your Genesys Account Representative if your organization requires access to the new utility.

Use the information in this topic to set up your Continuous Delivery (CD) pipeline.

Important

Your CD pipeline must accommodate any necessary steps to meet your corporate requirements such as, performing security scans and validation testing.

The Engage private edition release containers can be accessed through either of the following:

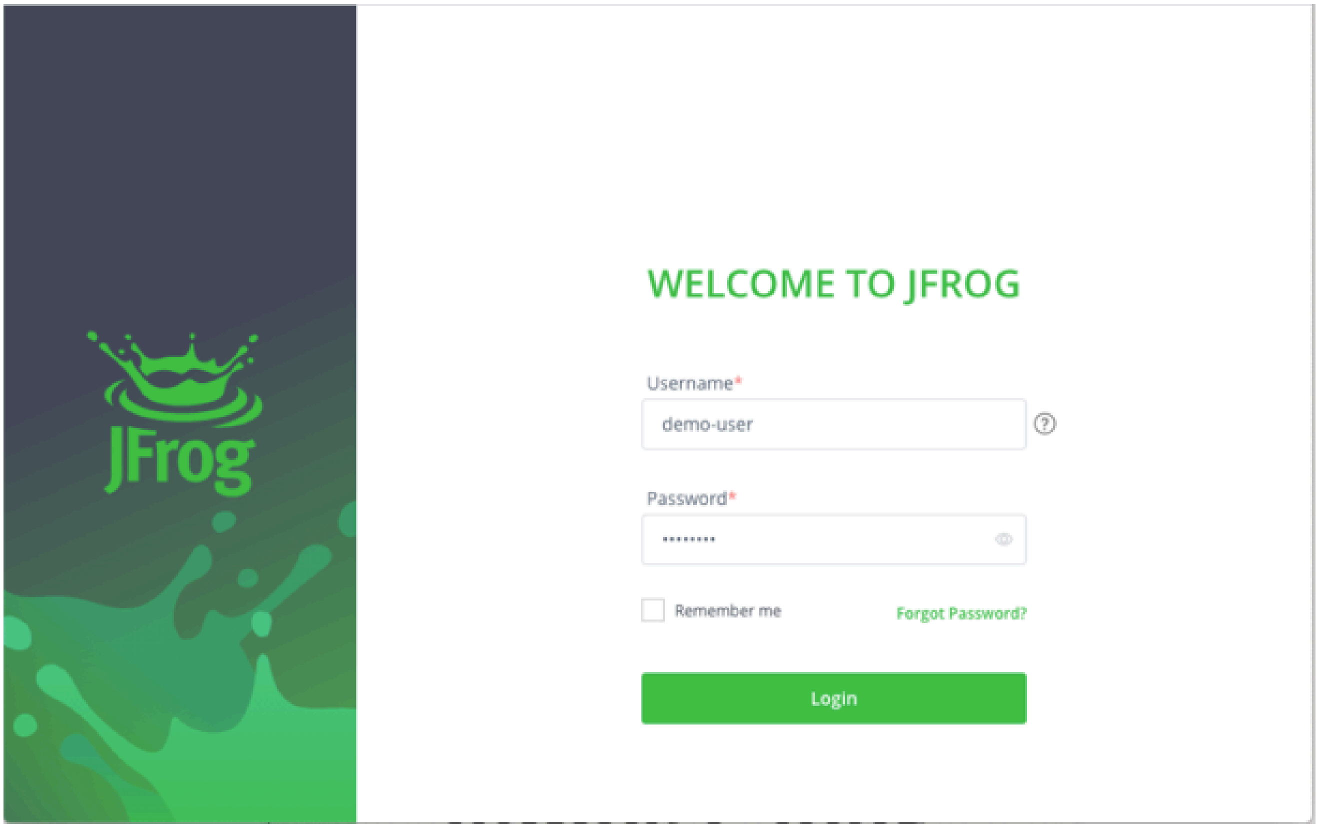
- Artifactory Edge Portal
- Artifactory Edge API
- Command Line Interfaces (CLI) for Docker, Helm, or cURL.

You can set up automated downloads for any of the available repositories using the **Set Me Up** screen for the corresponding repository.

Accessing repositories on JFrog

1. Navigate to the following URL using your browser:
   https://pureengageuse1.jfrog.io/ui/login/
2. Use your JFrog credentials from Genesys to log in.

> ### Important
> Credentials to access the Genesys repository on JFrog are automatically emailed to new accounts. Please contact your Genesys Account Representative if you have not received your credentials.

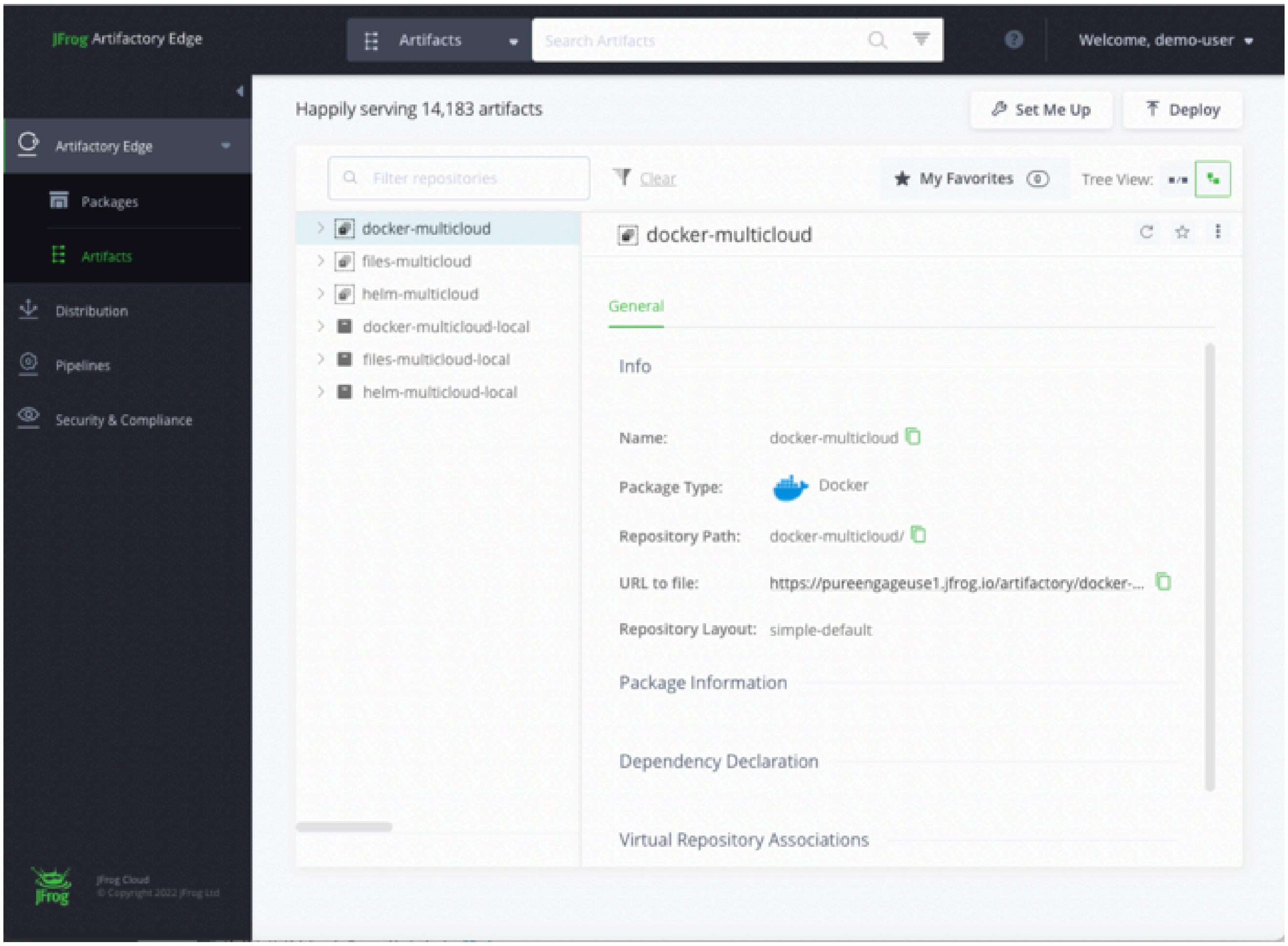


Artifactory Edge contains the following six Genesys repositories:

> **Important**
> A virtual repository aggregates several repositories with the same package type under a common URL.

Name of repository	Type of artifacts
helm-multicloud-local	Helm charts local repository
files-multicloud-local	Configuration files local repository
docker-multicloud-local	Docker local repository
helm-multicloud	Helm virtual repository
files-multicloud	Files virtual repository
docker-multicloud	Docker virtual repository

Once you log in, select **Artifacts** from the **Artifactory Edge** menu from the left pane. All repositories available for download from Genesys are listed.



Name: docker-multicloud

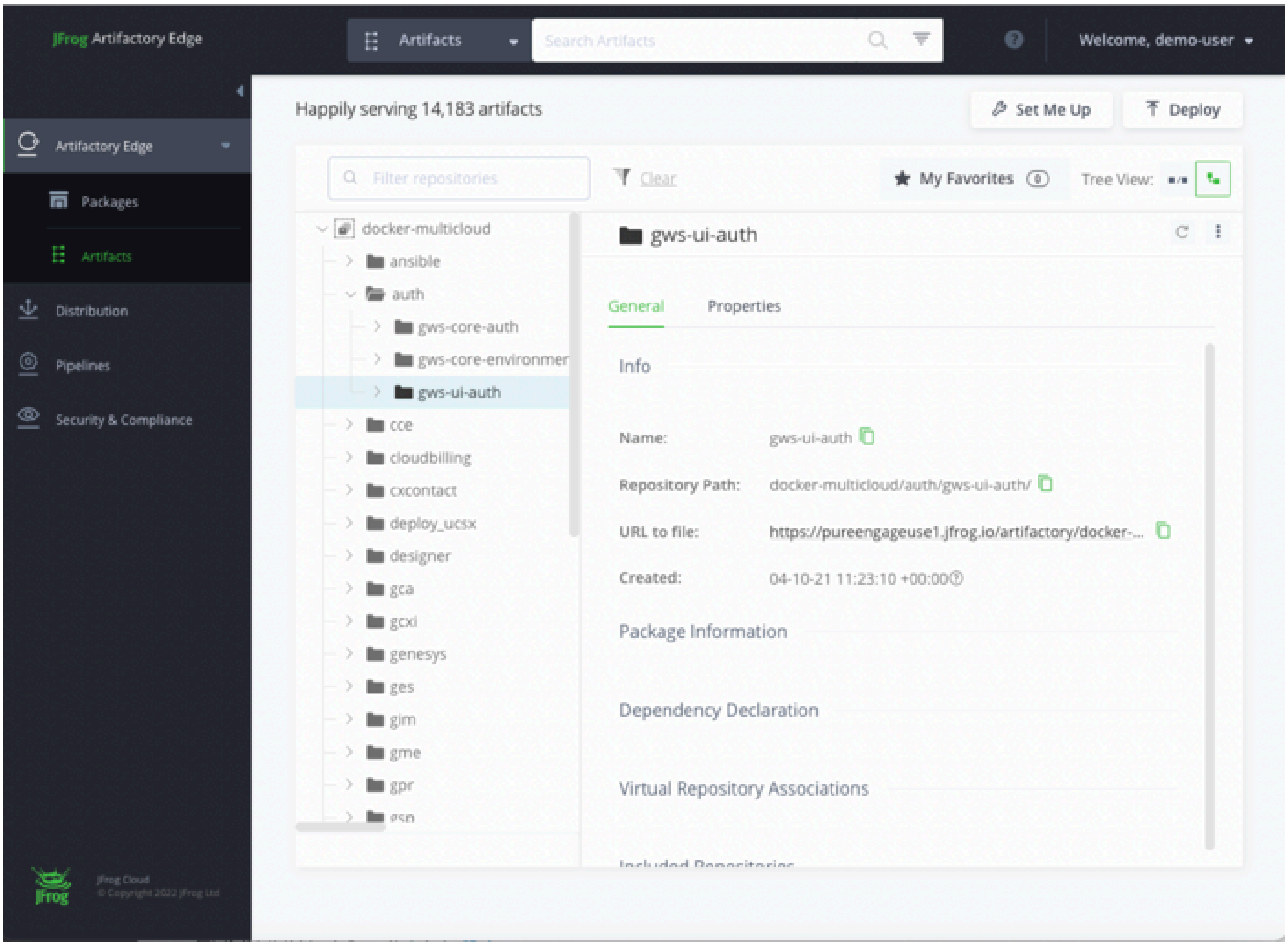
Package Type: Docker

Repository Path: docker-multicloud/

URL to file: https://pureengageuse1.jfrog.io/artifactory/docker-...

Repository Layout: simple-default

You can expand each repository and navigate to any of the files within a folder to view its properties on the left pane. Note that as all available files are listed, you must navigate to the one you require based on the date and version number.



Happily serving 14,183 artifacts

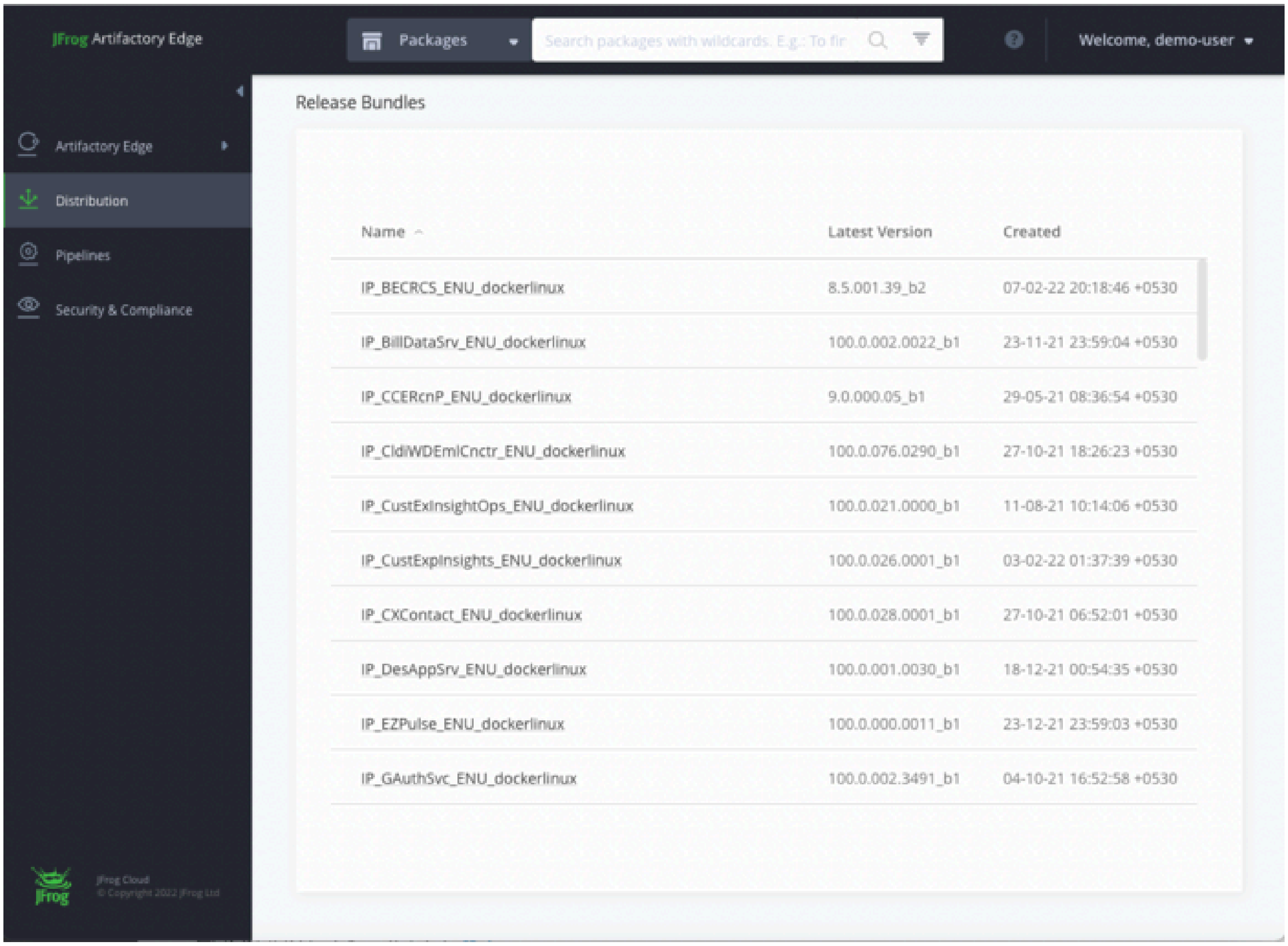
Name: gws-ui-auth

Repository Path: docker-multicloud/auth/gws-ui-auth/

URL to file: https://pureengageuse1.jfrog.io/artifactory/docker-...

Created: 04-10-21 11:23:10 +00:00

The **Distribution** view lists all files IPs and the files within each IP.



Release Bundles

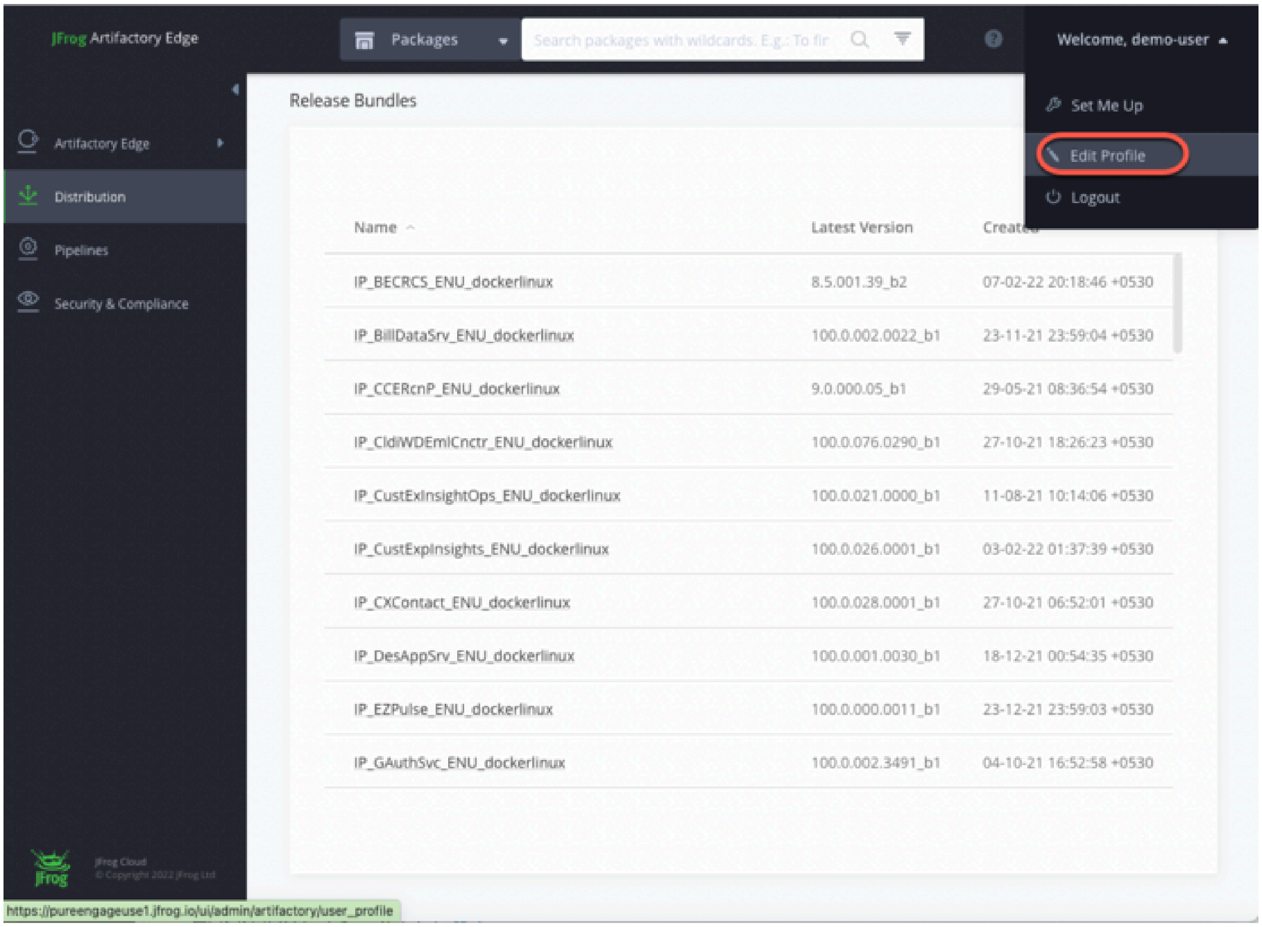
Name	Latest Version	Created
IP_BECRCS_ENU_dockerlinux	8.5.001.39_b2	07-02-22 20:18:46 +0530
IP_BillDataSrv_ENU_dockerlinux	100.0.002.0022_b1	23-11-21 23:59:04 +0530
IP_CCERcnP_ENU_dockerlinux	9.0.000.05_b1	29-05-21 08:36:54 +0530
IP_CldWDEmlCnctr_ENU_dockerlinux	100.0.076.0290_b1	27-10-21 18:26:23 +0530
IP_CustExInsightOps_ENU_dockerlinux	100.0.021.0000_b1	11-08-21 10:14:06 +0530
IP_CustExpInsights_ENU_dockerlinux	100.0.026.0001_b1	03-02-22 01:37:39 +0530
IP_CXContact_ENU_dockerlinux	100.0.028.0001_b1	27-10-21 06:52:01 +0530
IP_DesAppSrv_ENU_dockerlinux	100.0.001.0030_b1	18-12-21 00:54:35 +0530
IP_EZPulse_ENU_dockerlinux	100.0.000.0011_b1	23-12-21 23:59:03 +0530
IP_GAuthSvc_ENU_dockerlinux	100.0.002.3491_b1	04-10-21 16:52:58 +0530

You can set up automated downloads for any repository using the **Set Me Up** screen for the corresponding repository.

Signing up for update notifications

When you log in for the first time, sign up for email notifications on any updates to the packages in the repository. Note that you can set this up later too. But we recommend you set this up in order to receive regular notifications on any updates to the packages.

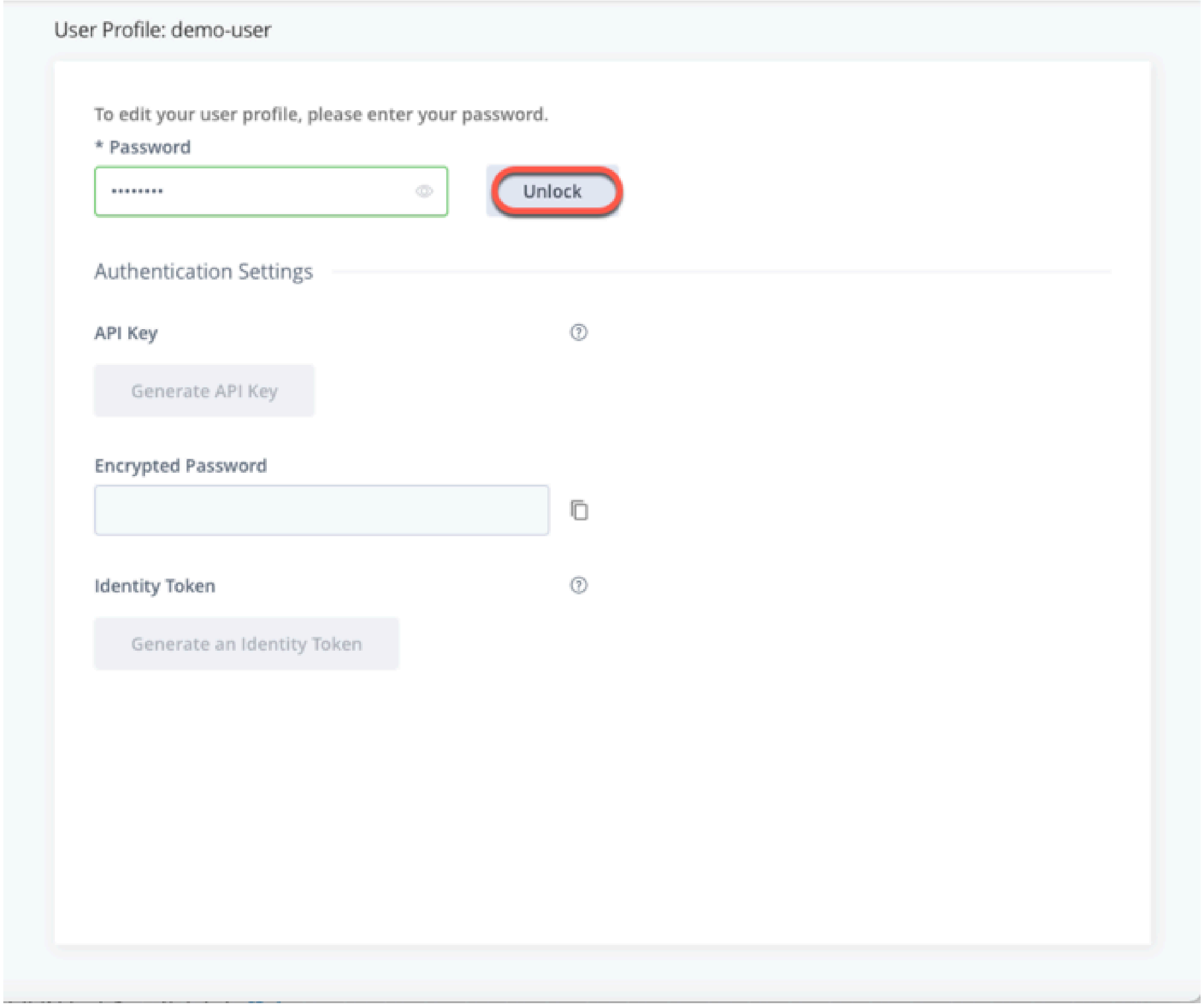
1. Click on your username at the top right corner of the screen.
2. Select the **Edit Profile** option from the drop-down.



Release Bundles

Name	Latest Version	Creat
IP_BECRCS_ENU_dockerlinux	8.5.001.39_b2	07-02-22 20:18:46 +0530
IP_BillDataSrv_ENU_dockerlinux	100.0.002.0022_b1	23-11-21 23:59:04 +0530
IP_CCERcnP_ENU_dockerlinux	9.0.000.05_b1	29-05-21 08:36:54 +0530
IP_CldWDEmlCnctr_ENU_dockerlinux	100.0.076.0290_b1	27-10-21 18:26:23 +0530
IP_CustExInsightOps_ENU_dockerlinux	100.0.021.0000_b1	11-08-21 10:14:06 +0530
IP_CustExpInsights_ENU_dockerlinux	100.0.026.0001_b1	03-02-22 01:37:39 +0530
IP_CXContact_ENU_dockerlinux	100.0.028.0001_b1	27-10-21 06:52:01 +0530
IP_DesAppSrv_ENU_dockerlinux	100.0.001.0030_b1	18-12-21 00:54:35 +0530
IP_EZPulse_ENU_dockerlinux	100.0.000.0011_b1	23-12-21 23:59:03 +0530
IP_GAuthSvc_ENU_dockerlinux	100.0.002.3491_b1	04-10-21 16:52:58 +0530

3. Enter your password and click **Unlock**.


User Profile: demo-user

To edit your user profile, please enter your password.

* Password

••••••••

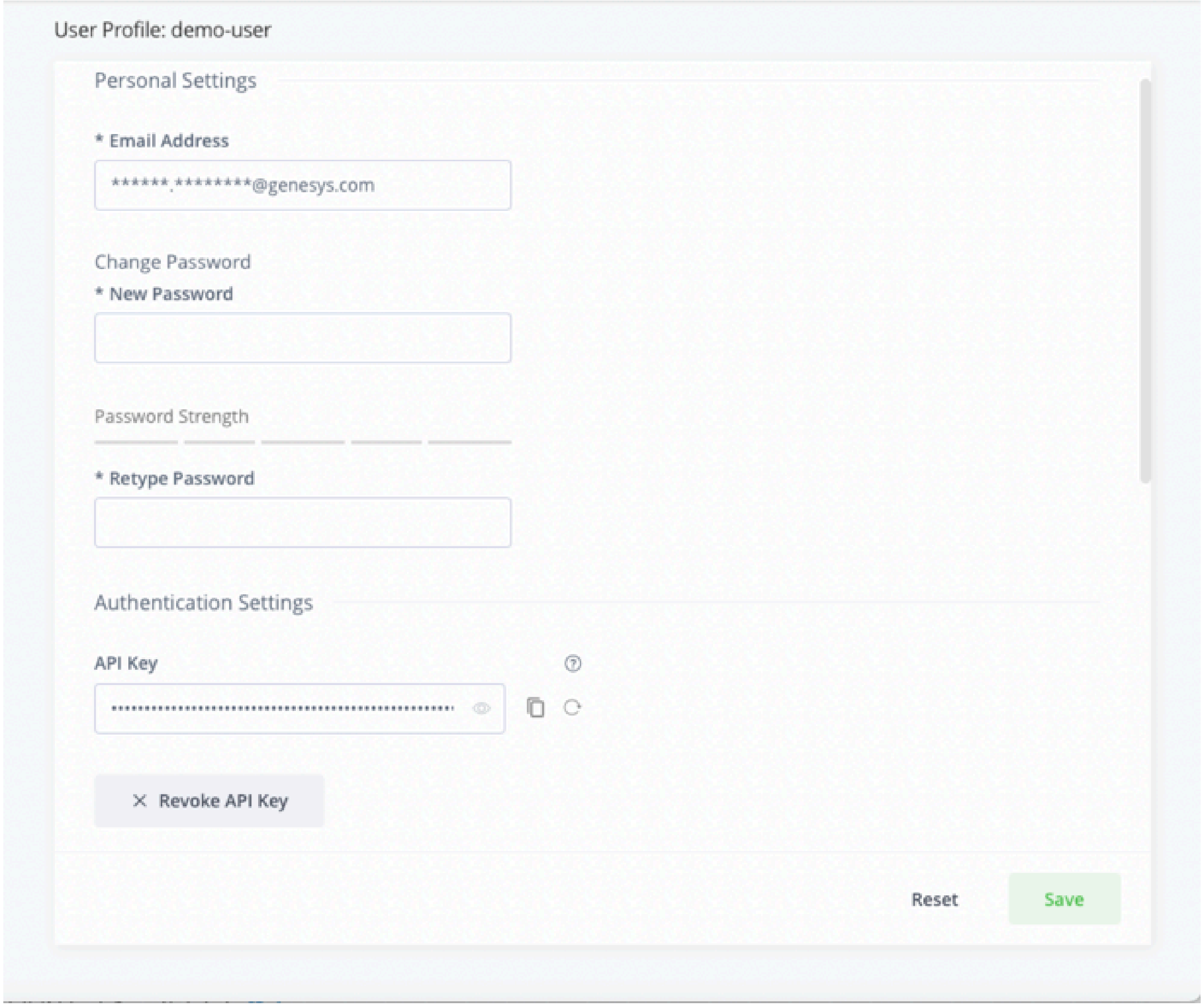
Authentication Settings

API Key

Encrypted Password

Identity Token

4. Verify your email address. Update if required and click **Save**.



User Profile: demo-user

Personal Settings

* Email Address: ******.********@genesys.com

Change Password

* New Password

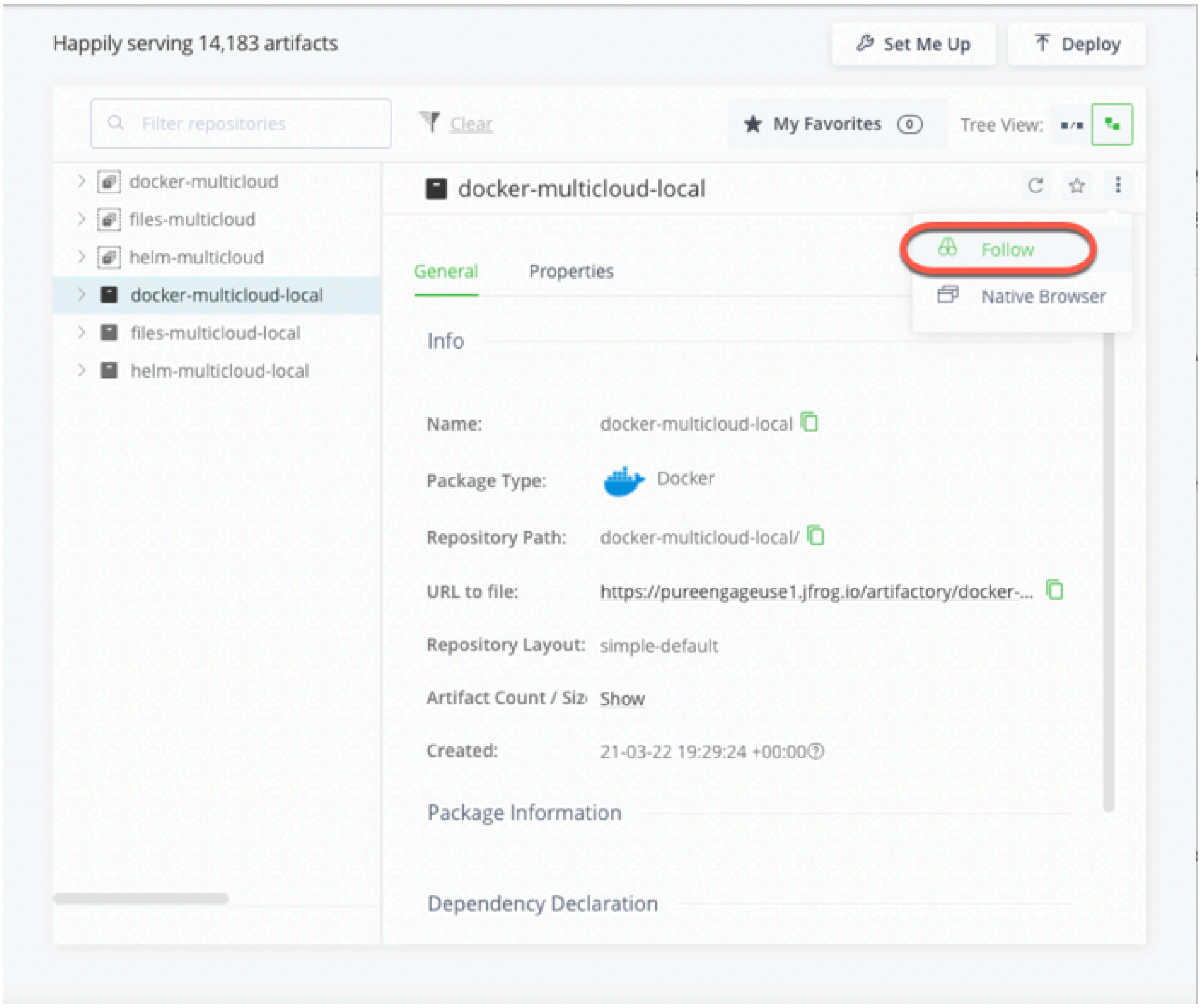
Password Strength

* Retype Password

Authentication Settings

API Key: ••

5. Navigate to the repository for which you want update notifications.
6. Click **Actions** at the top right corner, and then click **Follow** from the drop-down.



Happily serving 14,183 artifacts

Name: docker-multicloud-local

Package Type: Docker

Repository Path: docker-multicloud-local/

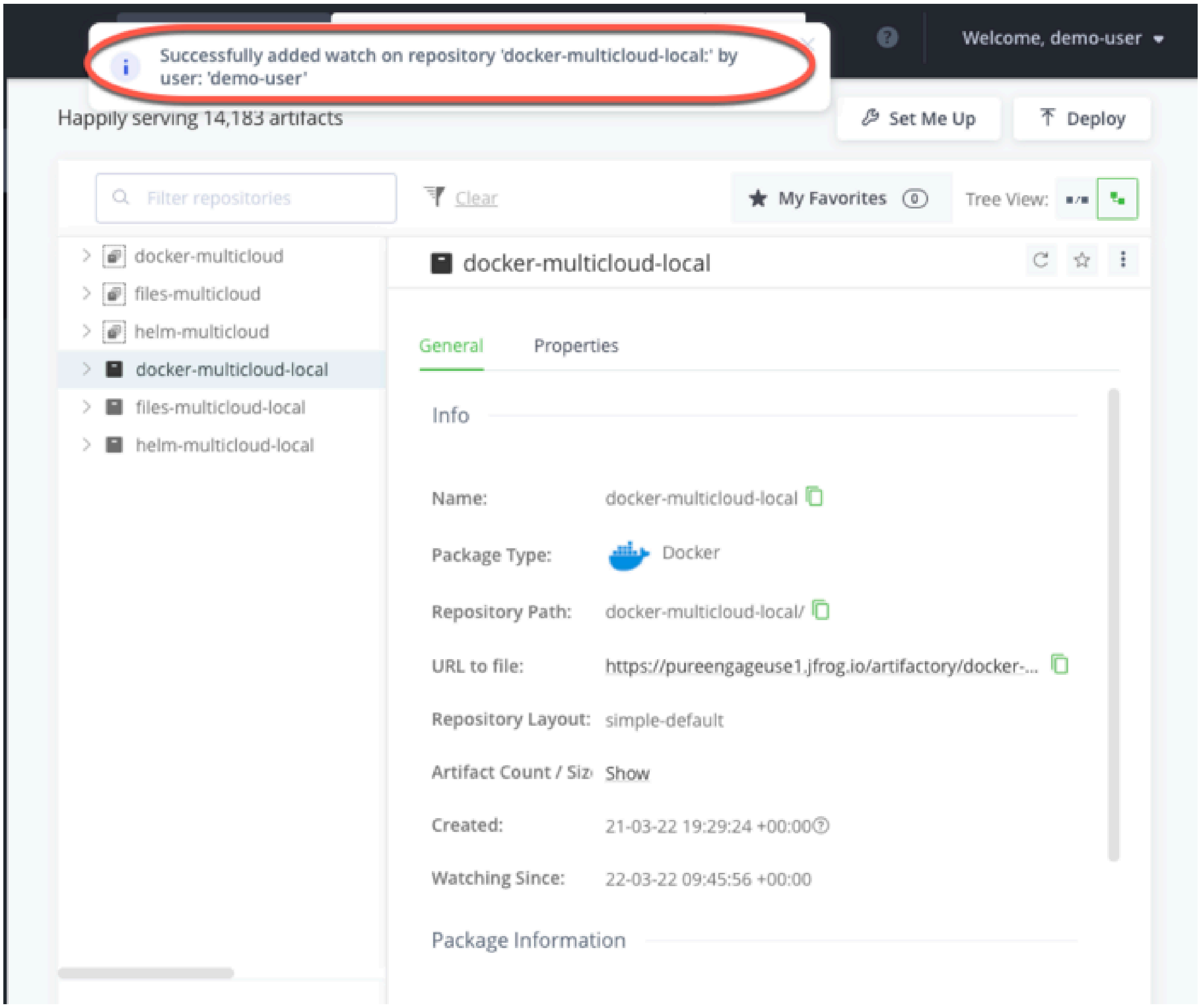
URL to file: https://pureengageuse1.jfrog.io/artifactory/docker-...

Repository Layout: simple-default

Artifact Count / Siz Show

Created: 21-03-22 19:29:24 +00:00

A confirmation message is displayed.



Successfully added watch on repository 'docker-multicloud-local:' by user: 'demo-user'

Happily serving 14,183 artifacts

docker-multicloud-local

Name: docker-multicloud-local

Package Type: Docker

Repository Path: docker-multicloud-local/

URL to file: https://pureengageuse1.jfrog.io/artifactory/docker-...

Repository Layout: simple-default

Artifact Count / Siz Show

Created: 21-03-22 19:29:24 +00:00

Watching Since: 22-03-22 09:45:56 +00:00

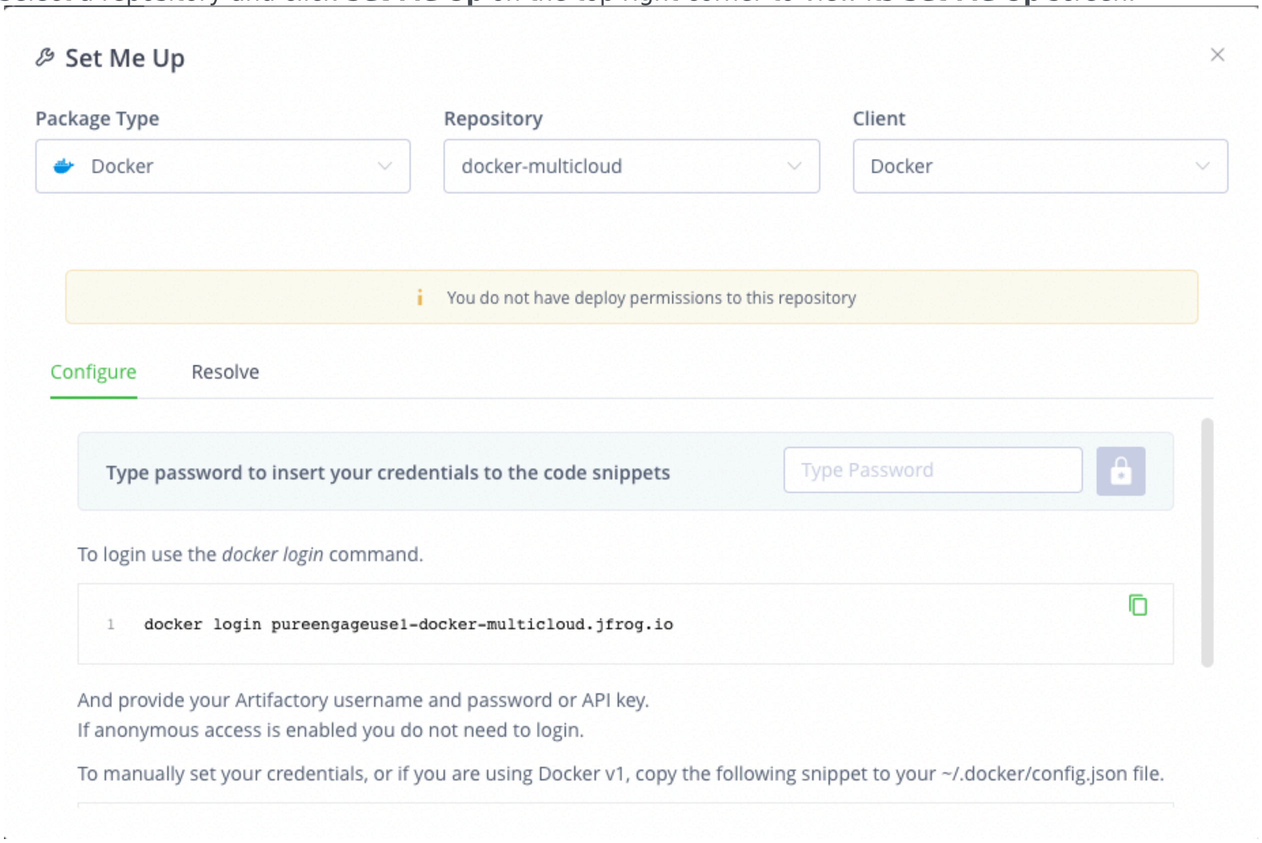
Important

Update notifications are accumulated across 1-minute intervals and sent in a single email.

Setting up automated downloads

You can integrate with external tools to automate your downloads from JFrog. The **Set Me Up** screen provides quick access to information on how to configure your different clients to work with the corresponding repositories you have created.

- Select a repository and click **Set Me Up** on the top right corner to view its **Set Me Up** screen.



Set Me Up

Package Type: Docker

Repository: docker-multicloud

Client: Docker

You do not have deploy permissions to this repository

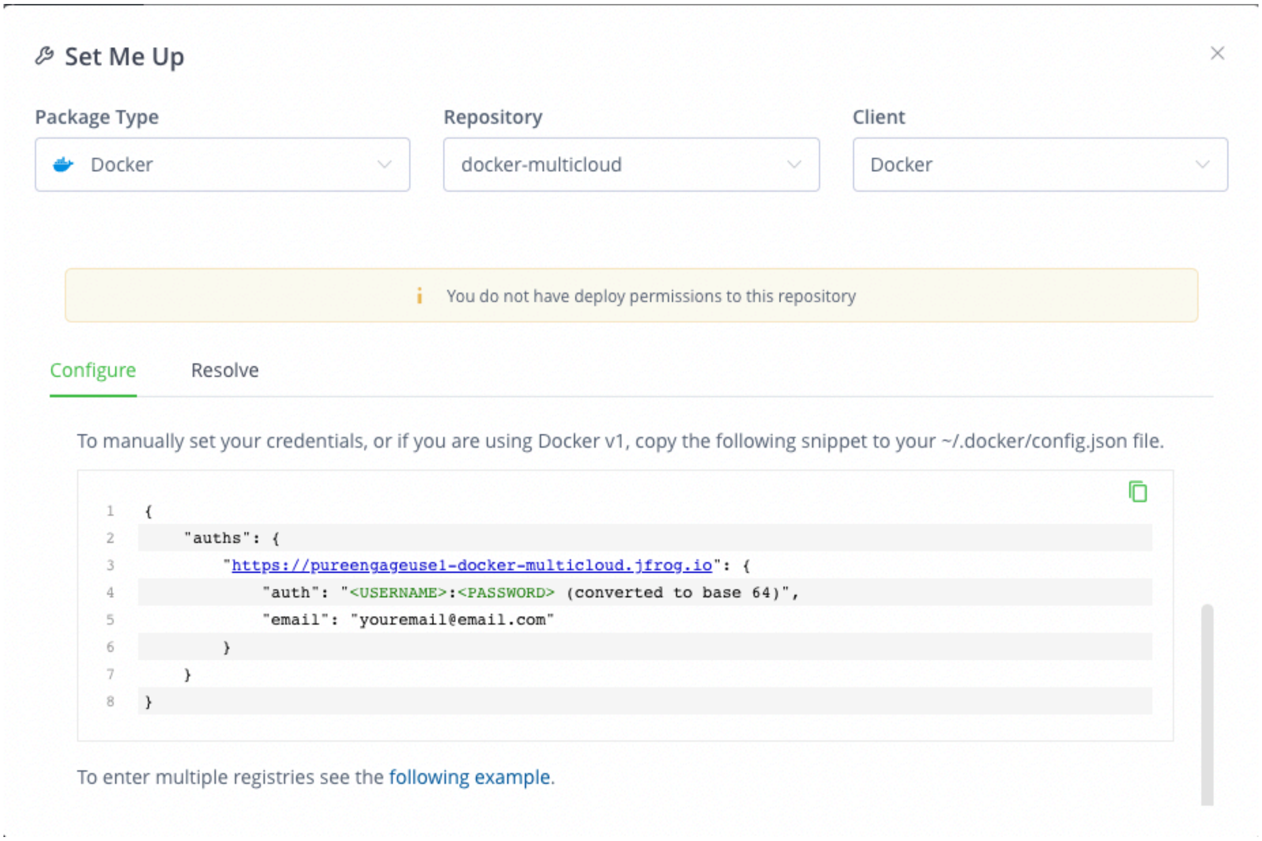
Type password to insert your credentials to the code snippets

To login use the *docker login* command.

```
docker login pureengageuse1-docker-multicloud.jfrog.io
```

And provide your Artifactory username and password or API key.
If anonymous access is enabled you do not need to login.

To manually set your credentials, or if you are using Docker v1, copy the following snippet to your ~/.docker/config.json file.



You do not have deploy permissions to this repository

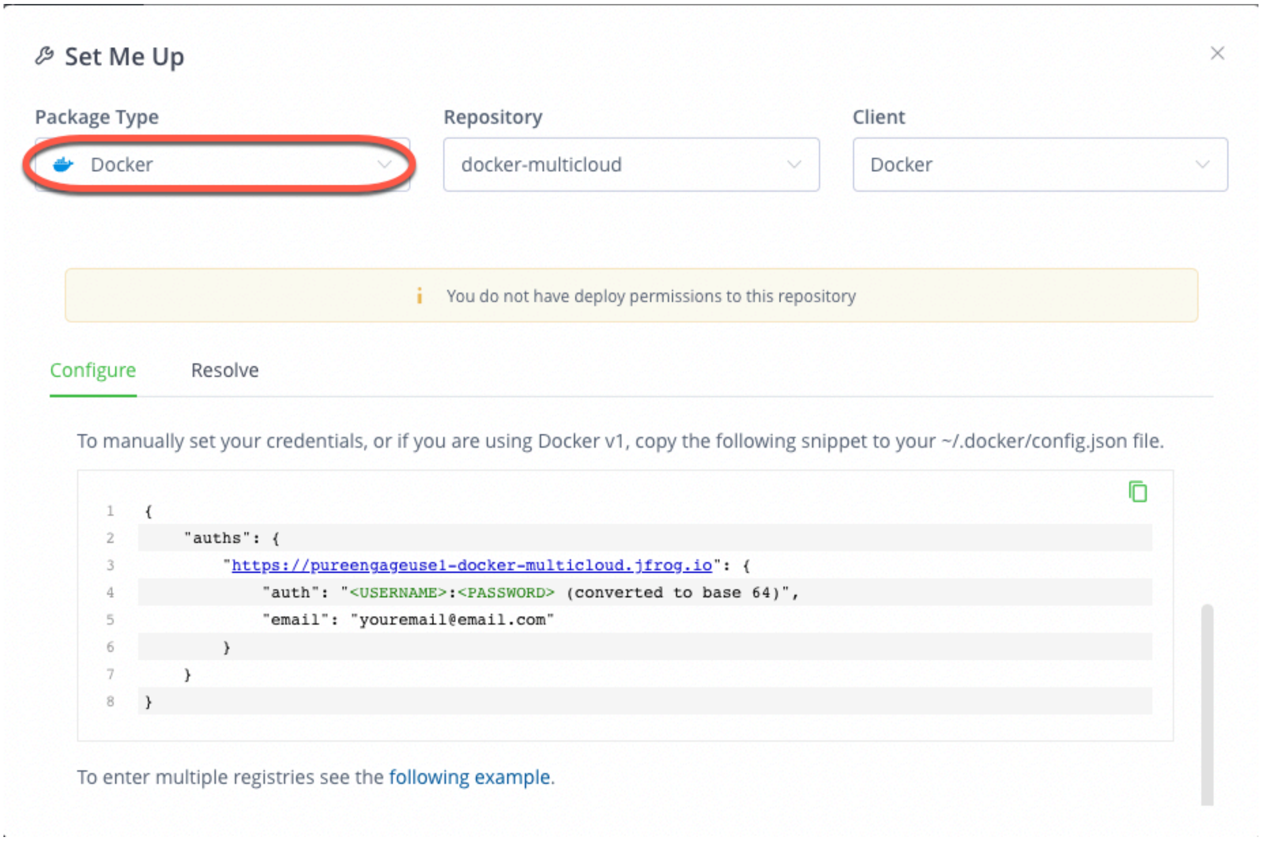
To manually set your credentials, or if you are using Docker v1, copy the following snippet to your ~/.docker/config.json file.

```
{
    "auths": {
        "https://pureengageuse1-docker-multicloud.jfrog.io": {
            "auth": "<USERNAME>:<PASSWORD> (converted to base 64)",
            "email": "youremail@email.com"
        }
    }
}
```

To enter multiple registries see the following example.

Downloading using Docker CLI

1. On the **Set Me Up** screen, select **Docker** from the **Package Type** drop-down.



You do not have deploy permissions to this repository

To manually set your credentials, or if you are using Docker v1, copy the following snippet to your ~/.docker/config.json file.

```
{
    "auths": {
        "https://pureengageuse1-docker-multicloud.jfrog.io": {
            "auth": "<USERNAME>:<PASSWORD> (converted to base 64)",
            "email": "youremail@email.com"
        }
    }
}
```

To enter multiple registries see the following example.

2. Provide the following docker login command in the **General** section as shown below:
   `docker login pureengageuse1-docker-multicloud.jfrog.io`
3. Provide your Artifactory username and password or the API key in the provided input field.

> ## Important
> You can set up your API key from the **Edit Profile** option.

4. To manually set your credentials, or if you are using Docker v1, copy the following snippet to your **~/.docker/config.json** file:

```
{
         "auths": {
         "https://pureengageuse1-docker-multicloud.jfrog.io" : {
"auth": ": (converted to base 64)", "email": "youremail@email.com"
                 }
         }
}
```

5. To pull an image use the docker pull command specifying the docker image and tag names:
   `docker pull pureengageuse1-docker-multicloud.jfrog.io/:`

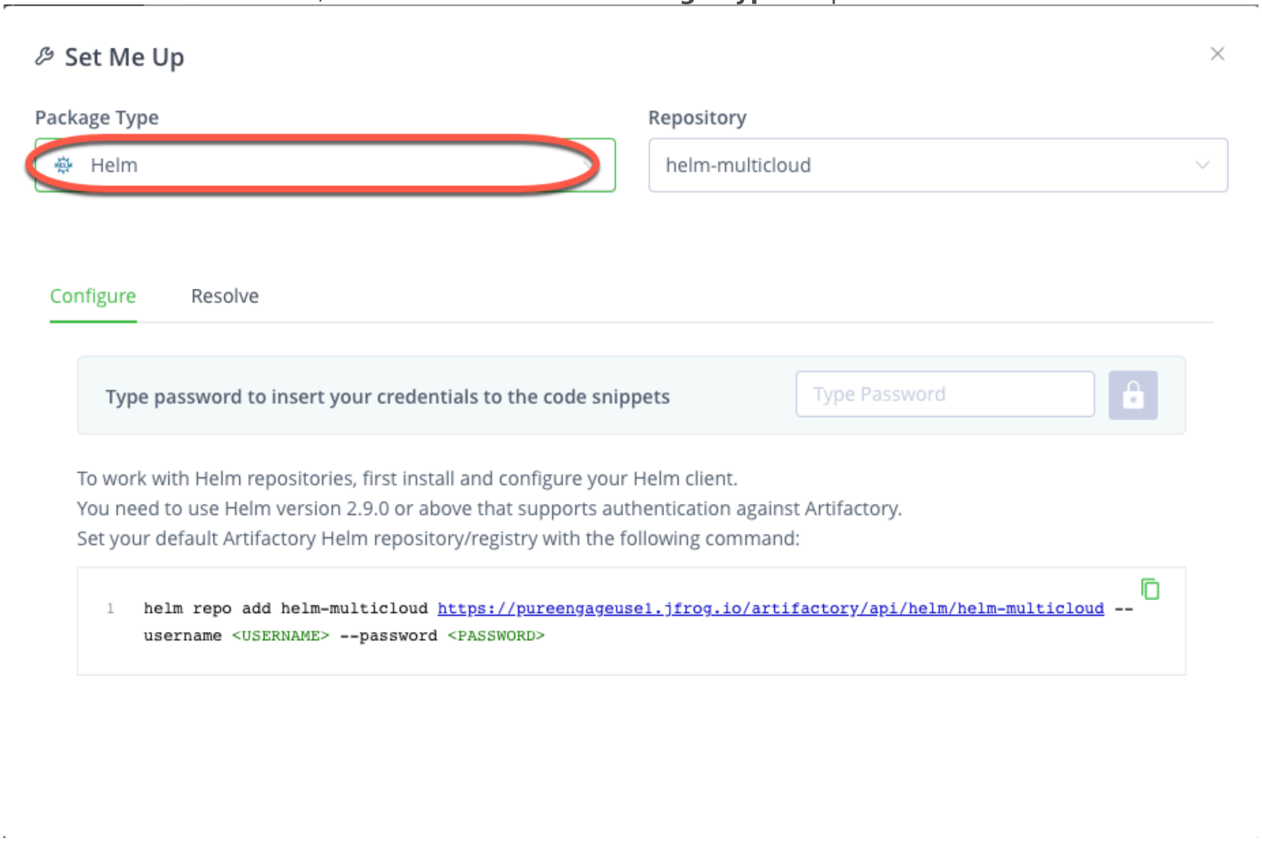
> **Important**
>
> Tagging allows you to group related container images together.

Downloading using the Helm CLI

To work with Helm repositories, you must have a Helm client installed and configured before you perform the following steps:

> **Important**
>
> You must use Helm version 2.9.0 or a higher version that supports authentication against Artifactory.

1. On the **SET ME UP** screen, select **Helm** from the **Package Type** drop-down.



Package Type: Helm

Repository: helm-multicloud

Type password to insert your credentials to the code snippets

To work with Helm repositories, first install and configure your Helm client.
You need to use Helm version 2.9.0 or above that supports authentication against Artifactory.
Set your default Artifactory Helm repository/registry with the following command:

```
helm repo add helm-multicloud https://pureengageuse1.jfrog.io/artifactory/api/helm/helm-multicloud --
username <USERNAME> --password <PASSWORD>
```

2. In the **General** section, set up your default Artifactory Helm repository/registry with the following command:
   `helm repo add helm-multicloud https://pureengageuse1.jfrog.io/artifactory/helm-multicloud --username --password`

3. In the **Resolve** section, provide the following commands to install a Helm Chart from the selected repository using your Helm command line client:
   ```
   helm repo update
   helm install helm-multicloud/[chartName]
   ```

Downloading using cURL

You can also download a package from the Edge Artifactory by accessing its API through a cURL command.

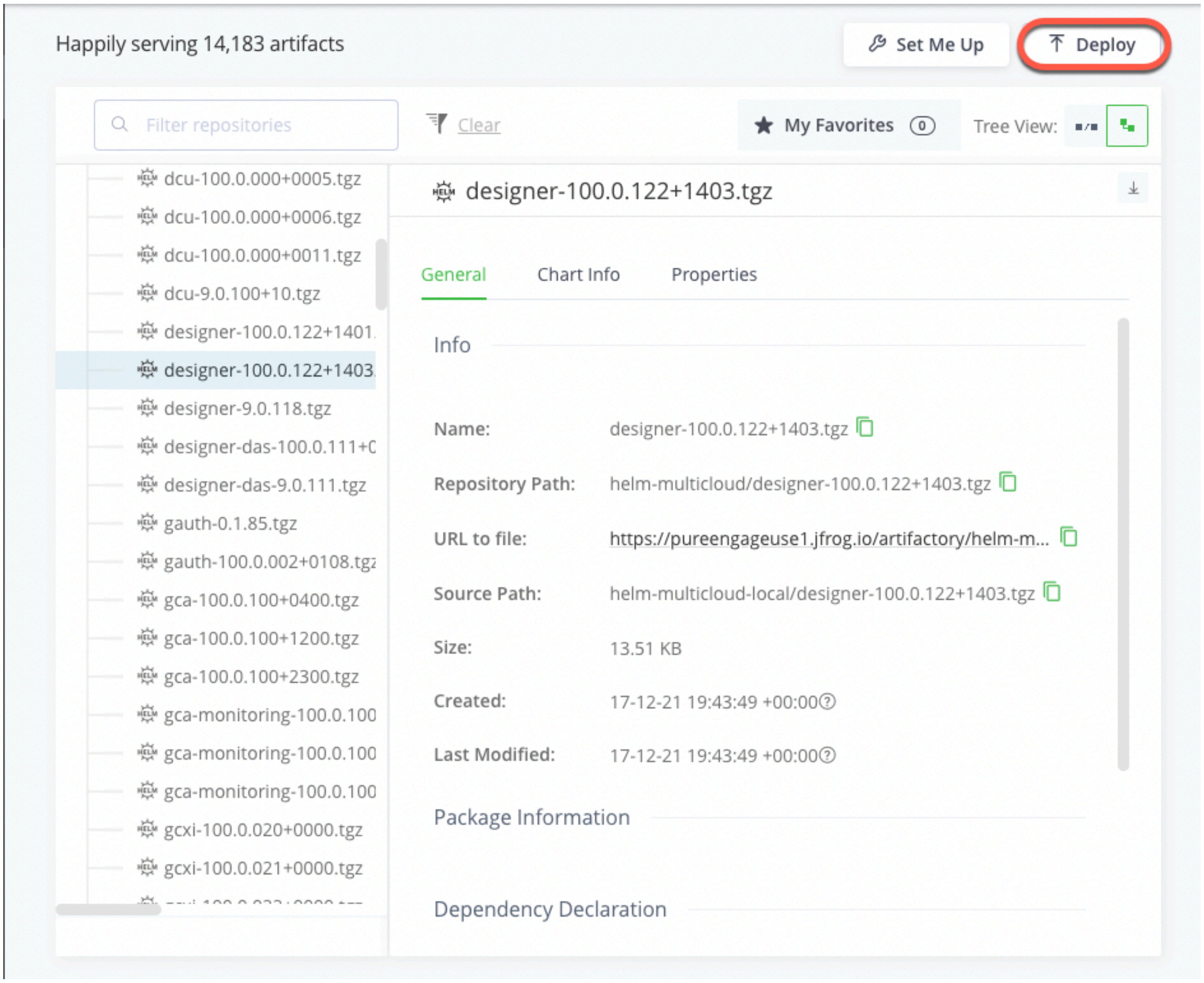
For example,

`curl -u: -O "https://pureengageuse1.jfrog.io/artifactory/helm-multicloud/cxcontact-022.03.121.tgz"`.

Downloading manually

JFrog Artifactory also supports manual downloads if you do not want to set up a CD pipeline.

1. Select the required artifact using the *Tree* browsing method or the *Simple* browsing method.
2. Click the **Deploy** option on the top right corner of the screen.



Name: designer-100.0.122+1403.tgz

Repository Path: helm-multicloud/designer-100.0.122+1403.tgz

URL to file: https://pureengageuse1.jfrog.io/artifactory/helm-m...

Source Path: helm-multicloud-local/designer-100.0.122+1403.tgz

Size: 13.51 KB

Created: 17-12-21 19:43:49 +00:00

Last Modified: 17-12-21 19:43:49 +00:00

For more information on browsing through the artifacts in the Artifact Repository Browser, refer to the Browsing Artifacts topic on the JFrog documentation site.

Additional reading material

- JFrog Artifactory Edge (an *Edge node*) is an edition of JFrog Artifactory with features customized to serve the primary purpose of distributing software to a runtime system such as a data center, a point-of-sale, or even a mobile device.
  For more information, refer to the JFrog Artifactory Edge topic on the JFrog documentation site.
- Local repositories are physical, locally-managed repositories into which you can deploy artifacts. Whereas, a virtual repository (or repository group) aggregates several repositories with the same package type under a common URL.
  For more information, refer to the Repository Management topic on the JFrog documentation site.

Overriding Helm chart values

Contents

Override values passed into the Helm chart through the Values.yaml file.

Related documentation:

-
-

RSS:

- For private edition

The values.yaml file

The **values.yaml** file is available for each service as part of the Helm chart. It is a source of content for the **Values** built-in object offered by Helm templates. The **Values** built-in object provides access to the values passed into a chart. These values control various infrastructure and deployment related settings for the corresponding Genesys Multicloud CX services.

> Important
>
> The name of the service that the file belongs to, is included as part the file name. For example, **designer-values.yaml** or **values_gauth.yaml** or **values-gws.yaml**. Refer to the individual service guides for exact file names.

Settings related to the following and more are available in a **values.yaml** file: *replica count*, *maximum replicas*, *deployment strategy*, *image repository*, *image tag*, *secrets*, *health probes*, *annotations*, *tolerations*, *security*, and *storage*. For a comprehensive list of the settings and their allowed values, refer to the individual service guides.

> Important
>
> - Service-specific parameters and environment variables are stored in ConfigMaps. Each service guide has more information pertaining to its ConfigMaps.

Overriding values

Default values are specified for most settings in a v**alues.yaml** file and these values are passed into the chart. For an initial deployment, if you want to change any of the values, you can do so by directly editing the values.yaml file using a plain text editor.

Apart from editing the YAML file directly, there are two methods you can use to override the values that are being passed into the chart from the **values.yaml** file. The two methods are as follows:

Using the --set flag

You can use a `--set` flag in your Helm commands to override the value of a setting in the YAML file. Specify the name of the setting and its new value after the `--set` flag in the Helm command.

Examples:

1. If you want to override the deployment strategy specified for a service during installation, you can use the Helm upgrade command with a `--set` flag as follows:
   `helm upgrade --install -green -f -values.yaml -100.0.112+xxxx.tgz --set .deployment.strategy=blue-green`
   The `--set` flag in the above command overrides the value for the `.deployment.strategy` setting in the **values.yaml** file and sets it to `blue-green`. So, irrespective of the value in the file for this particular setting, the service is installed using the `blue-green` strategy.
2. If you want to override the version of the service to install during initial deployment, you can use the Helm upgrade command as follows:
   `helm upgrade --install -f -values.yaml -100.0.112+xxxx.tgz  --set .image.tag= 9.0.1xx.xx.xx`
   The `--set` flag in the above command overrides the image tag version in the **values.yaml** file and provides a new version.

> ### Important
> When overriding values from the **values.yaml** file, note that strings must be specified within quotes to avoid type conversion errors. For example, `--set-string designer.designerConfig.envs.DES_ES_PORT="9200"` is different from `--set designer.designerConfig.envs.DES_ES_PORT=9200`. In the second case, `9200` is passed as an integer and not a string.

Using the --values flag

You can use a `--values` flag in your Helm commands to override the values in a chart and pass in a new file. Specify the name of the new file after the `--values` flag in the Helm command.

Example:

- `helm upgrade --install -f values.yaml -9.0.xx.tgz --values .yaml`The `--values` flag in the above command is passing on a new file with values to override the values in the chart.

Important

For detailed information on settings available in the **values.yaml** file, allowed values, default values, and override scenarios, refer to the individual service guides.

Service priorities for Genesys Multicloud CX services

Contents

Learn about service priorities of Genesys Multicloud CX Services.

Related documentation:

-
-

RSS:

- For private edition

Genesys has assigned a service priority class for each Genesys Multicloud CX service based on the Kubernetes Pod Priority guidelines. The guideline states to use a value of one million for high priority pods and values of two billion and above for Kubernetes itself for cluster critical Pods like kube-proxy and core-dns. Genesys has designed the service priority values for each priority class such as *Critical*, *Medium*, and *Low*, and categorized the services under different service priority classes based on their business function. For example, Voice services are given 'Critical' priority because they cannot handle long delays. You can override this value in your Helm charts before deployment.

Before overriding, remember that the Pods will be evicted from the node based on the service priority you set. Hence, it is essential to assign service priority based on your business requirements.

Overriding Service Priority

If you want to override the service priority for a service,

- In the **values.yaml** file of the corresponding service, locate the **priorityClassName** optional variable.
- Override the default service priority value by assigning the required value. You can assign any one of the following values—**genesysengage-critical-priority, genesysengage-medium-priority,** or **genesysengage-low-priority**. After overriding, your **values.yaml** configuration looks like the following:

```
priorityClassName: genesysengage-medium-priority
```

The following table illustrates the Genesys chosen priority class and its priority value.

Priority	Priority Value	Usage Notes
Critical	10,000,000	Use this priority for Genesys Multicloud CX services that must **not** be evicted due to resource limitations and can evict all other lower priority services, when needed.
High	1,000,000	Use this priority for Genesys Multicloud CX services that might be evicted by critical services but will evict lower priority services,

Priority	Priority Value	Usage Notes
		when needed.
Medium	100,000	Use this priority for Genesys Multicloud CX services that might be evicted by critical or high priority services but will only evict lower or default priority services, when needed.
	0	Use this priority for Genesys Multicloud CX services that can be evicted for more than 24 hours, if needed.

The following table illustrates the recommended priority for each Genesys Multicloud CX service at a granular level.

Services Groups	Services	Service Priority
Designer	Designer	medium
	Designer Application Service	critical
Genesys Web Services (GWS/GAPI)	9.x GWS Chat Service	high
	9.x GWS Configuration Service	critical
	9.x GWS Environment Service	critical
	9.x GWS Feedback Service	medium
	9.x GWS Interaction Service	high
	9.x GWS OCS Service	high
	9.x GWS Provisioning Service	high
	9.x GWS Setting Service	critical
	9.x GWS SPL Service	high
	9.x GWS Statistics Service	high
	9.x GWS UCS Service	high
	9.x GWS Voice Service	critical
	9.x GWS Workspace Service	critical
	Workspace Web Edition (9.x)	critical
	Agent Setup	critical
Genesys Engagement Service (Callback and Mobile)	Genesys Engagement Service	high
Genesys Cloud CX Hybrid Integration	Conversation Provider	high
	User Event Generator	high
	Data Sync	high

	Screen Recording Gateway	high
	Lightweight Authentication Service	high
Historical Reporting Back-end	GIM	high
	GCA	medium
	GSP	high
Historical Reporting Front-end	GCXI	medium
Realtime Reporting	Quick Update	high
	Pulse web backend	high
	Object Browser	high
	Tenant Load Distribution Server (LDS)	high
	Tenant Collector	high
Digital/Nexus	Nexus	high
	Interaction Server (IXN)	high
	UCS-X	high
IWD	IWD	high
	IWD DataMart	medium
	Email Service	high
CX-Contact	CX Contact API Aggregator	High
	CX Contact Campaign Manager	high
	CX Contact Compliance Manager	high
	CX Contact Job Scheduler	high
	CX Contact List Builder	high
	CX Contact List Manager	high
	CX Contact UI	high
GVP	Voice Platform MCP	critical
	Voice Platform MRCP Proxy	critical
	Voice Platform Reporting Server	high
	Voice Platform RM	critical
	Voice Platform Config Server	critical
	Voice Platform Tenant Provisioner	critical
WebRTC	WebRTC CoTurn Service	critical
	WebRTC Gateway Service	critical

Voice Microservices	Voicemail Service	high
	Dialplan Service	critical
	Config Service	critical
	Orchestration Service	critical
	Frontend Service	critical
	SIP Cluster Service	critical
	Registrar Service	critical
	Agent State Service	critical
	Call State Service	critical
	SIP Proxy	critical
	Tenant Service*	critical
PECA Portal (Hub)	Static Web page per tenant. This page will be deployed in Azure’s CDN in regions where the tenant is deployed.	critical
WFM 3rd party Connector	Aria Adapters	high
Telemetry	Telemetry Service	high
BDS	Generates usage billing data	medium
Genesys Authentication Services	Authentication Service (API)	critical
	Environment Service	critical
	Authentication UI	critical

**In private edition, the following functions are rearchitected into Tenant Service:*

- *Tenant call control functions (T-Servers)*
- *Configuration functions*
- *Routing functions*
- *Statistical functions*
- *Outbound Contact Server (OCS) functions*

Setting up a CD pipeline

Contents

Provides recommendations on setting up a Continuous Deployment (CD) pipeline in a your cloud private edition infrastructure for automated deployments.

Related documentation:

-
-

RSS:

- For private edition

CD Pipeline for Genesys Multicloud CX private edition

Genesys delivers its artifacts in the JFrog Artifactory Edge repository. You can pull the containers from JFrog and push it into your pipeline stream for automated deployments. Genesys strongly recommends automating deployments for its services via a CD pipeline.

Prerequisites

The tools you use in your CD pipeline must execute Helm charts as part of a pipeline.

Procedure

Here is a quick overview of the pipeline steps involved in a typical CD environment.

1. Download a container from the Genesys repository.
2. Push the downloaded container into your internal container registry or to a quarantine location for security scans.
3. Perform security scans on the container.
4. If you are confident with the scan results, promote the container to a test/pre-production environment.
5. In the test/pre-production environment, upgrade the container by referring the upgrade procedure of the specific service in its service-level documentation. You can also check out the high level upgrade procedure in this guide.
6. Test the updated environment by running the automated tests.
7. Once the test results are satisfying, promote the container to the next environment (if applicable to your organization) for further validation before moving to production.

> ### Important
> Your organization might have different environments other than the pre-production environment in order to test a new version of a container rigorously. Therefore, promoting a container to the next environment could mean a different environment for some users and production environment for other users.

8. Upgrade the container in the production environment by referring the upgrade procedure of the specific service in its service-level documentation. If you encounter any issue with the upgrade, you can always rollback to the previous point before the upgrade by referring the upgrade procedure of the specific service in its service-level documentation. You can also check out the high level rollback procedure in this guide.

Frequently asked questions

The following FAQs answer important considerations when you are planning your CI/CI pipeline.

What repository will Genesys use to provide Helm charts to customers?

Genesys will provide continuous delivery updates in the JFrog Artifactory Edge repository used for the initial deployment, as described in Downloading your Genesys Multicloud CX containers.

How many Helm charts are packaged for a Genesys Multicloud CX service?

Genesys typically packages one Helm chart per service. However, for specific services like Genesys Web Services (GWS), you can use the same Helm chart and deploy different services by varying the values in the Helm chart.

How do you solve dependencies between different Genesys components during deployment?

During initial deployment, we enforce a specific deployment order to be followed when you deploy Genesys Multicloud CX services. This will resolve the requirements on dependencies between different services. Once your initial deployment is up and running, you can upgrade individual services at different times.

We recommend you create a platform level CD pipeline to perform initial deployment in the required order.

Will introducing a new component or Helm parameter affect the existing ecosystem of services?

No. You can deploy the new service by following its instructions provided the core components like GAuth, GWS, Tenant service, etc. are already deployed.

Important

For additional information on pipelines and examples, refer to the Private Edition page in our public repository.

Upgrade overview

Contents

Provides an overview of Genesys Multicloud CX services upgrade.

Related documentation:

-
-

RSS:

- For private edition

Genesys Multicloud CX services are constantly evolving with new functionalities to provide the best user experience. To leverage these new capabilities, we recommend you to upgrade Genesys Multicloud CX services when there is a new release available. Keeping your systems always up to date reduces the risk of service outage and also allows us to support you better. In case you delay an upgrade for business reasons, make sure you are not behind two minor releases, that is, **N-2** releases, where **N** is the current release version in production. Learn about Genesys Multicloud CX service versions and Helm version from the Understanding versions page.

> Important
>
> If you are more than two minor releases behind, contact your Genesys Account Representative to avail our professional services support to bring your services to the latest.

Unlike traditional upgrading methods, upgrading containers through Kubernetes Upgrade strategies provide a bundle of benefits such as zero-downtime, low risk of failure, no major service outage, and easy fallback options to the previous release.

Next steps

Learn the Upgrade strategies supported by Genesys Multicloud CX services.

Upgrade strategies

Contents

Talks about different Upgrade strategies supported by Genesys Multicloud CX services.

Related documentation:

-
-

RSS:

- For private edition

Genesys supports industry standard upgrade strategies to upgrade Genesys Multicloud CX services. Our services are designed to support specific strategies based on the business function it fulfills.

Currently supported Upgrade strategies are:

- Blue/Green
- Canary
- Rolling Update

The following sections describe the fundamentals of each upgrade strategy.

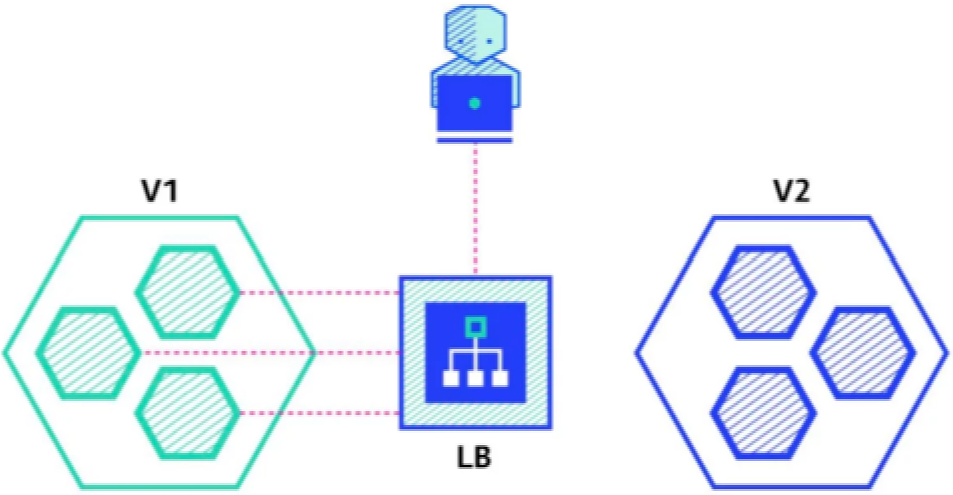
Blue/Green strategy

This is a release management technique that is employed to reduce the risk and provide zero-downtime for your services during upgrades. This method involves the following high level steps:

- You will create two identical production environments called ‘**blue**’ and ‘**green**’ and identify any one of them as ‘**active**’ and another one as ‘**idle**’. You can also imagine the active and idle environments as production and pre-production respectively.
- You deploy and validate the new release in the blue (idle) environment by running quality assurance and user acceptance tests.
- Once you are satisfied and there are no critical issues found, switch all the user traffic from green (active) environment to blue (idle) environment with the help of a router.
- Your blue environment becomes active now and the green environment becomes idle.

> **Important**
>
> You can keep the green (idle) environment as a fallback option for some time until you gain confidence with the new release running on the blue (active) environment. In case of unexpected issues arise with the newly deployed blue environment, you can always rollback to the last version by switching back to green.

In Kubernetes environments, this can be easily achieved by orchestrating the new resources like pods, containers, etc. and killing them when they are not needed. Services like GWS, and WebRTC support Blue/Green upgrade method.

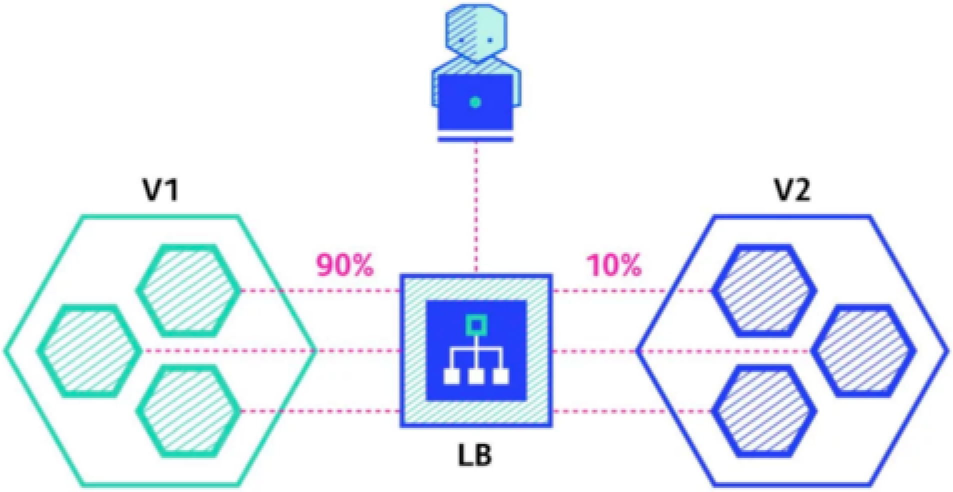


Canary strategy

In Canary deployments, you will upgrade only a subset of pod instances with the new release and make it available for limited number of users. In this upgrade strategy, both the subset of pod instances (with the new release) and the production pod instances (with the previous release) receives the live production user traffic. You can monitor the user behavior for bugs or performance issues from the upgraded pod instances. When the results are satisfying, you can incrementally roll out the new release to the wider group of pod instances in batches.

Canary deployment also offers easy rollback options to a previous version of the service.

Services like Voice supports Canary upgrade method.



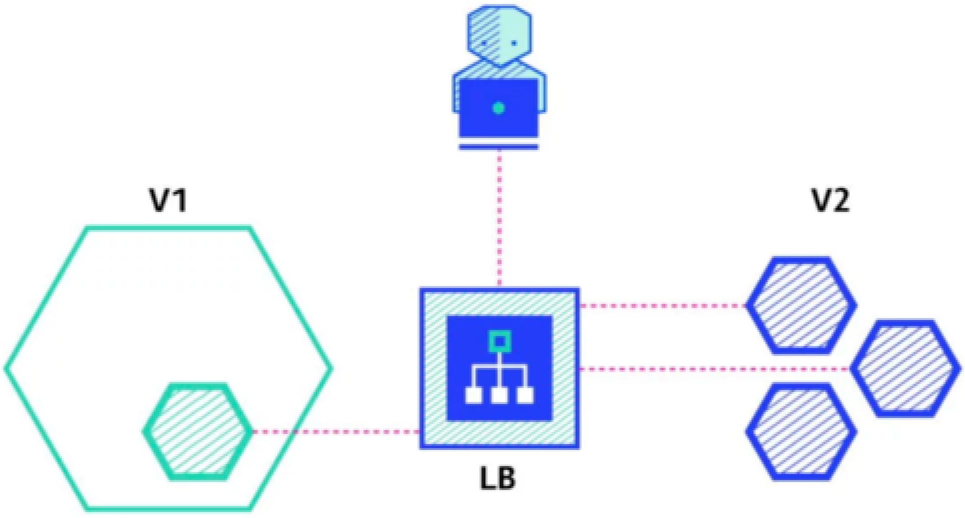
Rolling Update strategy

This upgrade strategy is similar to Canary. In Rolling Update strategy, you will replace the old pod instances running in production one-by-one with the new ones gradually. You will select a pod instance, deactivate it from the node, update with the new software, and then connect to the node.

During upgrade, the load shared by the pod instance being updated will be shared by other pod instances actively running in the ecosystem. Observe the behavior of the new pod instance, if it is satisfactory, rollout the update to all other pod instances in a similar fashion. This method ensures that at any point of time, the users are served with maximum number of pod instances.

Rolling Update strategy is suitable for upgrading a complex application that runs on multiple Kubernetes nodes (server cluster). It is also suitable for applications that directly interface with a load balancer so that the traffic in the absence of a pod instance (undergoing upgrade) is shared by the remaining active pod instances.

Services like GAuth, and Designer support Rolling Update method.



Next steps

Learn how to upgrade a Genesys Multicloud CX service from the Upgrade process page.

Upgrade process

Contents

Provides at a glance view of the processes involved in upgrading a Genesys Multicloud CX service.

Related documentation:

-
-

RSS:

- For private edition

Genesys delivers its container images and Helm Charts in JFrog Artifactory Edge repository, a publicly accessible repository. When there is a new version of the service available, you will receive notifications in your JFrog account.

> ### Tip
>
> If you have not subscribed to receive JFrog notifications, then visit Downloading your Genesys Multicloud CX containers page and set up your JFrog account to receive notifications.

You can easily pull the latest version of a Genesys Multicloud CX service from JFrog Artifactory Edge repository to your designated (quarantine) location for security scans or Continuous Delivery (CD) pipeline.

The following section details the upgrade process at a high level. It helps you to plan, prepare, and perform the upgrade of Genesys Multicloud CX services in cloud private edition infrastructure.

Prerequisites

- Access to JFrog account
- Established CD pipeline
- Established Backup process

Upgrade process

1. Select the Kubernetes upgrade strategy for the Genesys Multicloud CX service you are upgrading. Refer Upgrade strategies and select a strategy that best suits your production environment for the specific service.
2. Backup the data before starting the upgrade. If something goes wrong, you can always restore or rollback to the previous point before the upgrade.

3. Pull the latest containers and Helm charts from JFrog into your container registry.

> **Important**
>
> You can perform security scans on the pulled in containers and Helm charts from within the container registry. Security scanning depends on your organization's security policy and might not be applicable for all users.

4. Prepare your environment for the new upgrade. This step depends on the upgrade particular to that release. For example, you might have to create a new directory or pass a modified a yaml file.
5. Modify your Helm charts with appropriate overridable values.
6. Set up the CD pipeline in your environment.
7. Depending on the upgrade strategy you selected for the service, you will either upgrade a complete infrastructure, a subset of pod instances, or one pod instance at a time.
8. Run the `helm upgrade` command for your service by following the steps specific to the upgrade strategy you selected for the service. Keep in mind that the upgrade procedure varies for each upgrade strategy. Refer the service level documentation of the service you are upgrading for comprehensive explanations.
9. Test the upgrade by using the instructions given in the service level documentation of the service you have upgraded.

> **Important**
>
> If any of your test case fails or if you observe performance degradation, you can always rollback to the previous release.

10. Roll out the new version to all the users.

Rollback

Contents

Rolling back a Genesys Multicloud CX service to a previous release.

Related documentation:

-
-

RSS:

- For private edition

You can rollback a newly upgraded Genesys Multicloud CX service back to its previous version when you observe major issues or performance degradation. The upgrade strategies supported by Genesys Multicloud CX services provide flexible and easy methods to rollback to a previous version of the service.

Rolling back a service in Blue/Green upgrade strategy

If your Genesys Multicloud CX service is upgraded by using Blue/Green upgrade strategy, you can rollback to a previous version by switching the router back to the idle environment. For example, if you have diverted the traffic from green to blue, you can switch it back to green by running a `helm upgrade` command that mentions the color of the environment you want to switch to.

An example rollback command for Designer service is as follows:

```
helm upgrade --install designer-ingress -f designer-values.yaml
designer-100.0.112+xxxx.tgz --set designer.deployment.strategy=blue-green-ingress --
set designer.deployment.color=green
```

Rolling back a service in Rolling Update upgrade strategy

If your Genesys Multicloud CX service is upgraded by using the Rolling Update upgrade strategy, you can rollback to a previous version by modifying the **values.yaml** file with previous version's image tag and run the `helm upgrade` command.

An example rollback command for Designer service is as follows:

```
helm upgrade --install designer -f designer-values.yaml designer-100.0.112+xxxx.tgz
--set designer.image.tag=9.0.1xx.xx.xx
```

Rolling back a service in Canary upgrade strategy

If your Genesys Multicloud CX service is upgraded by using the Canary upgrade strategy, you can rollback to a previous version by creating a branch for the service, update the service image version/helm chart version or both.

Important

The examples given in this article are for reference purposes. Your service might rollback to a previous release using different method or by using a different set of parameters in the `helm upgrade` command. For correct instructions, always refer the procedures in your service's deployment guide.

Uninstall

Contents

Information on how to uninstall your service.

Related documentation:

-
-

RSS:

- For private edition

Uninstalling your service

To uninstall your deployed release, refer to the specific service's uninstall instruction:

Service	Uninstall Instruction
CX Contact	Uninstall CX Contact
Designer	Uninstall Designer
Digital Channels	Uninstall Digital Channels
Email	Uninstall Email
Genesys Authentication	Uninstall Genesys Authentication
Genesys Customer Experience Insights	Uninstall RAA
Genesys Engagement Service	Uninstall GES
Genesys Info Mart	Uninstall GIM
Genesys Pulse	Uninstall Genesys Pulse
Genesys Voice Platform	Uninstall GVP
Intelligent Workload Distribution	Uninstall IWD
Interaction Server	Uninstall Interaction Server
IWD Data Mart	Uninstall IWD Data Mart
Telemetry Service	Uninstall Telemetry Service
Tenant Service	
Universal Contact Service	Uninstall UCS
Voice Microservices	
Web Services and Applications	Uninstall GWS Ingress
WebRTC Media Service	Uninstall WebRTC
Workspace Web Edition	Uninstall Workspace Web Edition

Public Repository Links

Contents

This topic provides a list of links referenced in our public repository. They contain useful technical reference information that supplements the content in our private edition documents. There are also links to private edition-related announcements and a roadmap that indicates work items that are being currently worked on.

Related documentation:

-
-

RSS:

- For private edition

Technical reference information

- PE Wiki Home
- PE Cheat Sheet
- Troubleshooting 3rd Party Services
- GKC SBC Case Study
- PE Knowledge Base

Observability

- Monitoring
- Logging

Announcements

- PE Announcements

Roadmap

- PE Roadmap

Discussion forum

- PE Discussions

Need help?

- Before you contact us