GENESYS™

Voice Microservices Private Edition Guide

Configure the Voicemail Service

8/1/2025

Contents

Learn how to configure the Voicemail Service.

Related documentation:

-
-
-

RSS:

- For private edition

Override Helm chart values

For general information about overriding Helm chart values, see Overriding Helm Chart values in the *Genesys Multicloud CX Private Edition Guide*.

To assist you, Genesys provides override values for Voice Microservices in the Values.yaml file. If you need to change the default configuration, you must modify the Values file. The following section describes some common changes to these values.

Overriding values.yaml for the Voicemail Service

Common changes:

```
#Configure the GWS Base URL
context:
...
  envs:
    ...
    gwsBaseUrl:  # sample URL https:///auth/v3
    ...
```

You can override the Voicemail Service values for the following parameters based on what kind of storage you're using.

Persistent Volumes & Persistent Volume Claims

For PV and PVCs, use the following example:

```
#Blob Storage to be disabled and no value for mounts and volumes
  blobStorage:
    general:
      mode: 'k8s | csi'
      enabled: false
    mounts:
    volumes:
```

```
context:
...
  envs:
    ...
    storageInterface: "FileSystem"
    voicemailHome: "/storage/data"
    ...
```

Azure blob storage

For Azure Blob storage, use the following examples.

Azure Blob storage with the CSI driver:

1. Store the Azure Blob storage account access keys into Key Vault with a key-value pair:
   - key=
   - value=
2. Create a CSI driver from the Key Vault into the Kubernetes cluster with the name "keyvault-voice-voicemail-storage-csi-secrets".
3. Do the following changes in values.yaml:

   ```
   #Blob Storage is enabled
     blobStorage:
       general:
         mode: 'k8s | csi' # Secrets needs to mounted via K8s or CSI driver
         enabled: true
       mounts:
         - name: voicemail-secrets
           readOnly: true
           mountPath: "/opt/genesys/katana/voicemail/secret"
       volumes:
         - name: voicemail-secrets
           csi:
             driver: secrets-store.csi.k8s.io
             readOnly: true
             volumeAttributes:
               secretProviderClass: keyvault-voice-voicemail-storage-csi-secrets
   ```

   ```
   context:
   ...
     envs:
       ...
       storageInterface: "AzureBlob"
       voicemailHome: ""
       ...
   ```

Azure Blob storage with a Kubernetes secret:

1. Create a K8s secret cluster with the name "voicemail-storage-secrets" having a key-value pair:
   - key=
   - value=
2. Do the following changes in **values.yaml**

```
#Blob Storage is enabled
  blobStorage:
    general:
      mode: 'k8s | csi' # Secrets needs to mounted via K8s or CSI driver
      enabled: true
    mounts:
      - name: voicemail-secrets
        readOnly: true
        mountPath: "/opt/genesys/katana/voicemail/secret"
    volumes:
      - name: voicemail-secrets

secret:
          secretName: voicemail-storage-secrets


context:
...
  envs:
    ...
    storageInterface: "AzureBlob"
    voicemailHome: ""
    ...
```

AWS S3 bucket

For AWS S3 bucket with a Kubernetes secret, use the following examples.

1. Create a Kubernetes secret cluster with the name "voice voice-voicemail-s3-secrets" having a key-value pair:
    - accessKey=
    - secretkey=
    - bucketName=
    - bucketPath=

```
{
  "accessKey": "",
  "secretkey": "",
  "bucketName": "",
  "bucketPath": ""
}
```

2. Do the following changes in **values.yaml**:

```
#AWS S3 is enabled
  s3Storage:
    general:
      mode: 'k8s' # Secrets needs to mounted via K8s
      enabled: true
    mounts:
      - name: voicemail-s3-secrets
        readOnly: true
        mountPath: "/opt/genesys/katana/voicemail/secret"
    volumes:
      - name: voicemail-s3-secrets
        secret:
          secretName: voice-voicemail-s3-secrets
```

```
context:
...
  envs:
    ...
    storageInterface: "AWS_S3"
    voicemailHome: ""
    ...
```

You can now run the Helm install to deploy a Voicemail Service.

Create the IVR profile

The IVR profile is required for GVP to connect with the Voicemail Service.

To create the IVR profile for the Voicemail service:

1. Configure the following parameters:
   - Name = "voicemailservice"
   - Display name = "voicemailservice"
   - Annex options:
     - [gvp.general]\service-type=voicexml
     - [gvp.policy] section with default values
     - [gvp.service-parameters]\voicexml.gvp.appmodule="fixed,VXML-NG
     - [gvp.service-prerequisite]\initial-page-url and alternate-voice-xml = "http://voice-voicemail-service.voice.svc.cluster.local:8081/fs"
     - [gvp.service-prerequisite]\REQUESTURI_TIMEOUT = 5
2. In Tenant > Annex options:
   - [gvp.dn-groups]\voicemailservice = "55551111"
   - [gvp.dn-group-assignments]\voicemailservice = DB id of the IVR profile ("voicemailservice")
3. In the GVP namespace, add port 8081 to allow outbound communication. The Voicemail service runs in the Voice namespace with port 8081.