Voice Microservices Private Edition Guide

Consul requirements for Voice services

5/3/2026

Contents

Find details about Voice services settings that you must configure in Consul before you proceed to configure the Voice Microservices. Some of the configuration in Consul must be performed when you deploy Consul.

Related documentation:

-
-
-

RSS:

- For private edition

Before you deploy the Voice Services, you must deploy the infrastructure services. See Third-party prerequisites for the list of required infrastructure services.

It is your responsibility to deploy and manage all required third-party services, however – in addition to any other Consul configuration you require – there are specific Consul features that you must enable for Voice services.

Complete the work on this page before you make any changes described in Configure Voice Microservices.

Configure Consul features for Voice services

You can find system-level information about Consul on the Software requirements, Network settings, and Order of services deployment pages in *Setting up Genesys Multicloud CX Private Edition*.

When you deploy Consul, you must enable the following features for the Voice services:

- connectinject – To deploy sidecar containers in Voice pods.
- controller – To provide service intention functionality.
- syncCatalog – To sync Kubernetes services to Consul. Set **toK8S: false** and **addK8SNamespaceSuffix: false** for syncing services from Kubernetes to Consul.
- AccessControlList – To enable ACL, set **manageSystemACLs: true**.
- storageclass – To set the storage class to a predefined storage class.
- TLS – To enable TLS, set **enabled: true**. Additional information is required to set up TLS; the following sample includes that information.

The following sample shows the features configuration in Consul:

```
# config.yaml
```

```
global:
  name: consul
  tls:
    enabled: true
    caCert:
      secretName: consul-ca-cert
      # The key of the Kubernetes secret.
      secretKey: tls.crt
    caKey:
      # The name of the Kubernetes secret.
      secretName: consul-ca-key
      # The key of the Kubernetes secret.
      secretKey: tls.key
  acls:
    manageSystemACLs: true
connectInject:
  enabled: true
controller:
  enabled: true
syncCatalog:
  enabled: true
  toConsul: true
  toK8S: false
  addK8SNamespaceSuffix: false
```

Create a Consul bootstrap token

When you enable an Access Control List (ACL) in Consul, you must ensure that Voice services have access to read and write to Consul. To provide access, you create a token with permissions for Voice services in the Consul UI.

1. You can create the ACL bootstrap token when you deploy Consul, although it is possible to do this configuration later as part of the Voice Services deployment. You use the bootstrap token to log into the Consul UI to create a new ACL. Use the following command to get the bootstrap token:

   ```
   kubectl get secret consul-bootstrap-acl-token -n  -o go-template='{{.data.token |
   base64decode}}
   ```

2. Create a new token to which you'll assign the permissions required for Voice services. For example, we'll create a token with a value of a7529f8a-1146-e398-8bd7-367894c4b37b. You create a Kubernetes secret with this token. For example:

   ```
   kubectl create secret generic consul-voice-token -n voice --from-literal='consul-
   consul-voice-token=a7529f8a-1146-e398-8bd7-367894c4b37b'
   ```

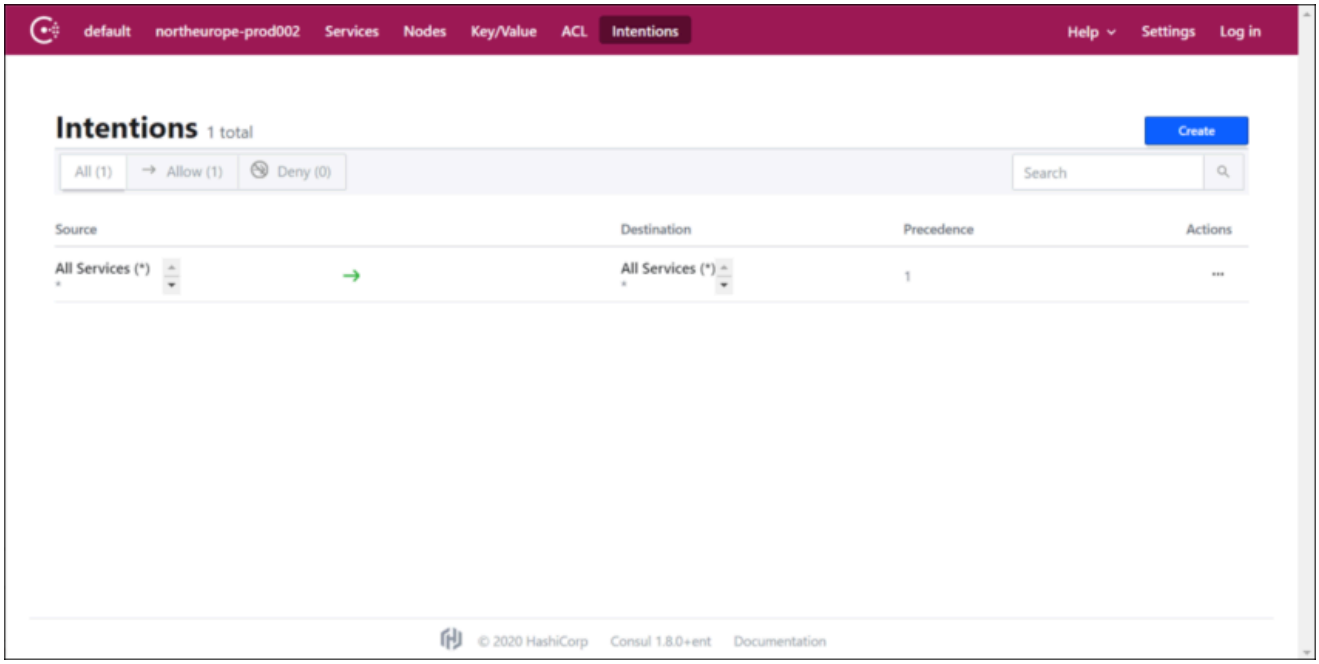
3. Create a policy (voice-policy) with the following list of permissions and assign it to the new token:

   ```
   service_prefix "" {
     policy = "read"
     intentions = "read"
   }
   service_prefix "" {
     policy = "write"
     intentions = "write"
   }
   node_prefix "" {
     policy = "read"
   }
   ```

```
node_prefix "" {
  policy = "write"
}
agent_prefix "" {
  policy = "read"
}
agent_prefix "" {
  policy = "write"
}
session_prefix "" {
  policy = "write"
}
session_prefix "" {
  policy = "read"
}
namespace_prefix "" {
 key_prefix "" {
  policy = "write"
 }
 session_prefix "" {
  policy = "write"
 }
}
key_prefix "" {
  policy = "read"
 }
key_prefix "" {
  policy = "write"
 }
```

Create Intentions in the Consul UI

Voice services use the Consul service mesh to connect between services. Consul has provision to either allow or deny the connection between services. This is done using *intentions*. Log into the **Intentions** tab using the bootstrap token and create a new intention to allow all source services to all destination services as shown in the following screenshot.


default
northeurope-prod002
Services
Nodes
Key/Value
ACL
Intentions
Help
Settings
Log in
Intentions 1 total
Create
All (1)
Allow (1)
Deny (0)
Search
Source
Destination
Precedence
Actions
All Services (*)
All Services (*)
1
© 2020 HashiCorp
Consul 1.8.0+ent
Documentation