Interaction Server Private Edition Guide

Deploy Interaction Server

4/5/2026

Contents

Learn how to deploy Interaction Server (IXN) into a private edition environment.

Related documentation:

-
-
-

RSS:

- For private edition

Assumptions

- The instructions on this page assume you are deploying the service in a service-specific namespace, named in accordance with the requirements on Creating namespaces. If you are using a single namespace for all private edition services, replace the namespace element in the commands on this page with the name of your single namespace or project.
- Similarly, the configuration and environment setup instructions assume you need to create namespace-specific (in other words, service-specific) secrets. If you are using a single namespace for all private edition services, you might not need to create separate secrets for each service, depending on your credentials management requirements. However, if you do create service-specific secrets in a single namespace, be sure to avoid naming conflicts.

Deploy

> **Important**
> Make sure to review Before you begin for the full list of prerequisites required to deploy Interaction Server.

Environment setup

GKE

1. Log in to the gke cluster.

   ```
   gcloud container clusters get-credentials gke1
   ```

2. Create ixn project in the GKE cluster using the following manifest file:

```
Create Interaction Server project in gke cluster using following manifest file:
'''create-ixn-namespace.json'''
{
"apiVersion": "v1",
"kind": "Namespace",
"metadata": {
"name": "ixn",
"labels": {
"name": "ixn"
}
}
}
kubectl apply -f apply create-ixn-namespace.json
```

3. Confirm the namespace creation.

```
kubectl describe namespace ixn
```

4. (Optional step) Create a secret for docker-registry in order to pull image from JFrog.

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

AKS

1. Log in to the AKS cluster

```
az aks get-credentials --resource-group $RESOURCE_GROUP --name $AKS_CLUSTER_NAME
```

2. Create ixn project in the AKS cluster using the following manifest file:

```
{
"apiVersion": "v1",
"kind": "Namespace",
"metadata": {
"name": "ixn",
"labels": {
"name": "ixn"
}
}
}
kubectl apply -f apply create-ixn-namespace.json
```

3. Confirm the namespace creation.

```
kubectl describe namespace ixn
```

Prepare cluster resources

To prepare your resources, create secrets and a default pull secret for the cluster.

Create secrets

Create Kubernetes (K8s) secrets for Redis and Kafka access in the IXN namespace:

```
 kubectl delete secret redis-ors-secret --ignore-not-found

 kubectl create secret generic redis-ors-secret \
 --from-literal='voice-redis-ors-
stream={"password":"PaSSword","port":"1234","rejectUnauthorized":"false","servername":"redis-
cluster.namespace.svc.cluster.local"}'

 kubectl delete secret kafka-shared-secret --ignore-not-found

 kubectl create secret generic kafka-shared-secret \
 --from-literal='kafka-secrets={"bootstrap": "infra-kafka-cp-
kafka.infra.svc.cluster.local:9092"}'
```

The following is a case when username and password are needed for Kafka authentication. A Kubernetes secret creation command will look like this:

```
kubectl create secret generic kafka-shared-secret \

--from-literal='kafka-secrets={"bootstrap": "kafka-service.kafka.svc.cluster.local:9092",
"username":"…", "password":"…"}'
```

Service account

Either create a service account and set it in Helm values file or just modify an existing one after Helm is installed and service account is created.

Here is an example of created service account, it must be named as ixn-server- for consul injection working.

```
kubectl get serviceaccounts
NAME                                          SECRETS   AGE
ixn-server-
```

Deploy IXN via Helm

To deploy IXN via Helm, follow these steps:

1. Download the latest version of Interaction Server installation Helm Charts from the artifactory. See the JFrog Platform Artifactory.

2. Extract parameters from the chart to see multiple (default) values used to fine-tune the installation.

```
$ helm show values /ixn > override_values.yaml
```

Configure the following key entries in the IXN override_values.yaml file:

IXN Server

Secrets:

```
ixnServer:
```

```
  secrets:
    db:
      # -- Enable Interaction Server database secret
      enabled: true
      # -- Interaction Server database secret name
      secretName: ixn-db-secret
      # -- Interaction Server database username to put in the secret
      username: ""
      # -- Interaction Server database password to put in the secret
      password: ""
```

Database:

```
db:
    # -- Interaction Server Database engine
    engine: "postgre"
    # -- Interaction Server Database name
    name: "ixn-db"
    # -- Interaction Server Database host
    host:
    # -- Interaction Server Database port
    port: 5432
    # -- Interaction Server Database connection string suffix
```

ixnNode

Database:

```
  db:
    # -- Interaction Server Node DB host
    host:
    # -- Interaction Server Node DB port
    port: 5432
    # -- Interaction Server Node DB name
    name: ixn-node
```

Redis:

```
  redis:
    # -- Interaction Server Node connects to Redis host
  - host:
    port:
    # -- Is Redis instance a Cluster or not
    is_redis_cluster: "true"
```

Secrets:

```
secrets:
    db:
      # -- Enable Interaction Server Node database secret
      enabled: true
      # -- Interaction Server Node database secret name
      secretName: ixn-node-db-secret
      # -- Interaction Server Node database username to put in the secret
      username: ""
      # -- Interaction Server Node database password to put in the secret
      password: ""
```

Tenant:

```
tenant:
  # -- Tenant UUID or GWS ID
  id: ""
  # -- Tenant short ID
  sid:
```

You can apply multiple override values to customize your setup. However, Genesys recommends using minimal overriding values in the installation.

The following is a sample override_values.yaml file. (Also, refer to Log storage, Consul connection, and Volume mounts.)

```
# -- Add labels to all pods
podLabels: {}

# -- Add annotations to all pods
podAnnotations: {}

image:
  # -- Images registry
  registry: "pureengage-docker-staging.jfrog.io"
  # -- Images pull policy
  pullPolicy: Always #IfNotPresent
  # -- imagePullSecrets must have the following format:

  # - name: pullSecret1

  # - name: pullSecret2
  imagePullSecrets:
    - name: pullsecret
    #- name: jfrog-stage-credentials

ixnService:
  image:
    ixnServer:  #see versions.yaml
      # -- Interaction Server repository
      #repository: "ixn/interaction_server"
      # -- Interaction Server tag
      #tag: "latest"

    logSidecar:
      # -- Enable Interaction Server logging sidecar
      enabled: true
      # -- Interaction Server logging sidecar docker image repository
      repository: "fluent/fluent-bit"
      # -- Interaction Server logging sidecar docker image tag
      tag: "1.8.5"

    ixnNode:  #see versions.yaml
      # -- Interaction Server Node docker image repository
      #repository: "ixn/ixn_node"
      # -- Interaction Server Node docker image repository tag
      #tag: "latest"

  service:
    # -- Enable Kubernetes service for Interaction Service
    enabled: true

  # -- Volumes provided to Interaction Service pod
  # Must be declared with starting `\|-`
```

```
  # since they are parsed as a template
  volumes: |-
    - name: redis-ors-secret
      secret:
        secretName: redis-ors-secret
    - name: kafka-shared-secret
      secret:
        secretName: kafka-shared-secret

  # -- Security Context
  # ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-
security-context-for-a-container
  # Containers should run as genesys user and cannot use elevated permissions
  # !!! THESE OPTIONS SHOULD NOT BE CHANGED UNLESS INSTRUCTED BY GENESYS !!!
  securityContext: {}

  # -- Priority Class
  # ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
  priorityClassName: ""

  # -- Node labels for assignment.
  # ref: https://kubernetes.io/docs/user-guide/node-selection/
  nodeSelector: {}

  # -- Extra labels
  # ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/
  labels: {}

  # -- Extra Annotations
  # ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/
  # Must be declared with starting `\|-`
  # since they are parsed as template
  annotations: |-
      "consul.hashicorp.com/connect-inject": "true"
      "consul.hashicorp.com/connect-service": {{ include "ixn.consulIxnServerName" . | quote
}}
      "consul.hashicorp.com/connect-service-port": server-default
      "consul.hashicorp.com/connect-service-upstreams": |-
        voice-config:{{ .Values.ixnService.ixnNode.configNode.port }},
        {{ printf "tenant-%s:%d" .Values.tenant.id (int
.Values.ixnService.ixnServer.confServer.port) }}
      consul.hashicorp.com/service-tags: 'service-ixn'
      consul.hashicorp.com/service-meta-tenant-id: {{ .Values.tenant.id }}
      consul.hashicorp.com/service-meta-tenant-sid: {{ .Values.tenant.sid | quote }}

  prometheus:
    monitoringService:
      # -- Enable a service with Prometheus annotations for metrics scraping
      enabled: true

  ixnServer:

    serviceAccount:
      # -- Create service account for Interaction Server
      create: true
      # -- The name of the ServiceAccount to use.
      # If not set and create is true, a name is generated using the fullname template
      name:

    ports:
      # -- Interaction Server default port
      default: 7120
      # -- Interaction Server health port
```

```
  health: 9100

secrets:
  db:
    # -- Enable Interaction Server database secret
    enabled: true
    # -- Interaction Server database secret name
    secretName: ixn-db-secret-${TENANT_ID}
    # -- Interaction Server database username to put in the secret
    username: "${POSTGRES_USER}"
    # -- Interaction Server database password to put in the secret
    password: "${POSTGRES_PASSWORD}"

confServer:
  # -- Interaction Server connects to Configuration Server host
  host: "localhost"
  # -- Interaction Server connects to Configuration Server port
  port: 8888
  # -- Interaction Server application name in Configuration Server
  appName: InteractionServer

db:
  # -- Interaction Server Database engine
  engine: "postgre"
  # -- Interaction Server Database name
  name: "ixn-${TENANT_ID}"
  # -- Interaction Server Database host
  host: ${POSTGRES_ADDR}
  # -- Interaction Server Database port
  port: 5432
  # -- Interaction Server Database connection string suffix
  connectionString: "KeepaliveInterval=1;KeepaliveTime=60;"
  # -- Interaction Server Database Blob Chunk Size. Can be left empty
  optionBlobChunkSize:
  # -- Interaction Server Database Reconnect Pause. Can be left empty
  optionReconnectPause:
  # -- Interaction Server Database schema name. Can be left empty
  schemaName:

dbinit:
  enabled: true

logStorage:
  # -- Interaction Server logs mount path
  mountPath: "/mnt/logs"
  # -- Interaction Server log storage size. Used for PVC, can be left empty
  storageSize: 1Gi
  # -- Interaction Server log storage class name. Used for PVC, can be left empty
  storageClassName:
  # -- A volume definition to be inserted into Interaction Server container definition
  volume:
    emptyDir: {}

livenessProbe:
  # -- Interaction Server liveness probe initial delay
  initialDelaySeconds: 15
  # -- Interaction Server liveness probe check period
  periodSeconds: 30
  # -- Interaction Server liveness probe timeout
  timeoutSeconds: 3
  # -- Interaction Server liveness probe failure threshold
  failureThreshold: 3
```

```
    readinessProbe:
      # -- Interaction Server readiness probe initial delay
      initialDelaySeconds: 15
      # -- Interaction Server readiness probe check period
      periodSeconds: 30
      # -- Interaction Server readiness probe timeout
      timeoutSeconds: 3
      # -- Interaction Server readiness probe failure threshold
      failureThreshold: 3

    startupProbe:
      # -- Interaction Server startup probe check period
      periodSeconds: 30
      # -- Interaction Server startup probe failure threshold
      failureThreshold: 120

    resources:
      requests:
        # -- Interaction Server Kubernetes CPU request
        cpu: "100m"
        # -- Interaction Server Kubernetes memory request
        memory: "512Mi"
      limits:
        # -- Interaction Server Kubernetes CPU limit
        cpu: "200m"
        # -- Interaction Server Kubernetes memory limit
        memory: "2Gi"

    jvmOptions:
      1: "-XX:+UnlockExperimentalVMOptions"
      2: "-XX:+UseCGroupMemoryLimitForHeap"
      3: "-XX:+UseG1GC"
      4: "-XX:MinHeapFreeRatio=5"
      5: "-XX:MaxHeapFreeRatio=10"
      6: "-XX:GCTimeRatio=4"
      7: "-XX:AdaptiveSizePol"

    # -- Volumes mounted into an Interaction Server container
    volumeMounts:
      kafka-shared-secret:
        readOnly: true
        mountPath: "/mnt/env-secrets/kafka-secrets"

  ixnNode:

    settings:
      # -- Interaction Server Node settings mount path
      mountPath: "/mnt/settings"
      # -- Interaction Server Node settings file name
      file: "settings.json"

    storingSessions:
      # -- Enable storing Interaction Server Node sessions in database
      enabled: true
    db:
      # -- Interaction Server Node DB host
      host: ${POSTGRES_ADDR}
      # -- Interaction Server Node DB port
      port: 5432
      # -- Interaction Server Node DB name
      name: ixn-node-${TENANT_ID}
      options:
        # -- Keep Interaction Server Node db connection alive
```

```
        keepAlive: true
        # -- Keep Interaction Server Node db connection alive: initial delay in milliseconds
        keepAliveInitialDelayMillis: 300000
        # -- Interaction Server Node db connection ssl options.
        # Details: https://nodejs.org/docs/latest-v14.x/api/
tls.html#tls_new_tls_tlssocket_socket_options
        # If no any specific TLS options are required, yet connection should be established
via TLS,
        # just "ssl" property with empty object as value is required, like ssl: {}
        # here we made empty ssl: to disable it since ssl disabled for OC Postgres
        ssl:

    ports:
      # -- Interaction Server Node default port
      default: 6120

    configNode:
      # -- Interaction Server Node connects to config server host
     host: "localhost"
      # -- Interaction Server Node connects to config server port
      port: 11100

    redis:
      # -- Interaction Server Node connects to Redis host
    - host: ${REDIS_ADDR}
      port: ${REDIS_PORT}
      # -- Is Redis instance a Cluster or not
      is_redis_cluster: "true"

    secrets:
      db:
        # -- Enable Interaction Server Node database secret
        enabled: true
        # -- Interaction Server Node database secret name
        secretName: ixn-node-db-secret-${TENANT_ID}
        # -- Interaction Server Node database username to put in the secret
        username: ${POSTGRES_USER}
        # -- Interaction Server Node database password to put in the secret
        password: "${POSTGRES_PASSWORD}"

    dbinit:
      enabled: true

    redisOptions:
      tls:
        # -- Enable TLS mode for Interaction Server Node to Redis connection
        enabled: false
        # -- Reject unauthorized hostnames when using TLS connection
        rejectUnauthorized: false

    consul:
      # -- A consul host is either a string literal or valueFrom definition to be inserted
into pod definition
      host:
        valueFrom:
          fieldRef:
            fieldPath: status.hostIP
      # -- Consul HTTP port
      port: 8500
      # -- Connect to Consul using SSL mode: true or false
      sslMode: false

    livenessProbe:
```

```
      # -- Interaction Server Node liveness probe initial delay
      initialDelaySeconds: 15
      # -- Interaction Server Node liveness probe check period
      periodSeconds: 30
      # -- Interaction Server Node liveness probe timeout
      timeoutSeconds: 3
      # -- Interaction Server Node liveness probe failure threshold
      failureThreshold: 3

    readinessProbe:
      # -- Interaction Server Node readiness probe initial delay
      initialDelaySeconds: 15
      # -- Interaction Server Node readiness probe check period
      periodSeconds: 30
      # -- Interaction Server Node readiness probe timeout
      timeoutSeconds: 3
      # -- Interaction Server Node readiness probe failure threshold
      failureThreshold: 3

    resources:
      requests:
        # -- Interaction Server Node Kubernetes CPU request
        cpu: "40m"
        # -- Interaction Server Node Kubernetes memory request
        memory: "128Mi"
      limits:
        # -- Interaction Server Node Kubernetes CPU limit
        cpu: "300m"
        # -- Interaction Server Node Kubernetes memory limit
        memory: "320Mi"

    # -- Volumes mounted into an Interaction Server Node container
    volumeMounts:
      redis-ors-secret:
        readOnly: true
        mountPath: "/mnt/env-secrets/redis-secrets"

    # env:
    #   DEBUG: ioredis:*

ixnVQNode:
  image:
    ixnVQNode:  #see versions.yaml
      # -- Interaction Server VQ Node docker image repository
      #repository: "ixn/ixn_vq_node"
      # -- Interaction Server VQ Node docker image tag
      #tag: "latest"

  serviceAccount:
    # -- Create service account for Interaction Server VQ Node
    create: true
    # -- The name of the ServiceAccount to use.
    # If not set and create is true, a name is generated using the fullname template
    name:

  ports:
    # -- Interaction Server VQ Node default port
    default: 7122
    # -- Interaction Server VQ Node health port
    health: 9102

  resources:
    requests:
```

```
      # -- Interaction Server VQ Node Kubernetes CPU request
      cpu: "30m"
      # -- Interaction Server VQ Node Kubernetes memory request
      memory: "128Mi"
    limits:
      # -- Interaction Server VQ Node Kubernetes CPU limit
      cpu: "300m"
      # -- Interaction Server VQ Node Kubernetes memory limit
      memory: "160Mi"

  # -- Security Context
  # ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-
security-context-for-a-container
  # Containers should run as genesys user and cannot use elevated permissions
  # !!! THESE OPTIONS SHOULD NOT BE CHANGED UNLESS INSTRUCTED BY GENESYS !!!
  securityContext: {}

  # -- Priority Class
  # ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
  priorityClassName: ""

  # -- Node labels for assignment.
  # ref: https://kubernetes.io/docs/user-guide/node-selection/
  nodeSelector: {}

  # -- Interaction Server VQ Node extra labels
  # ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/
  labels: {}

  # -- Extra Annotations
  # ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/
  # Must be declared with starting `\|-`
  # since they are parsed as a template
  annotations: |-
    "consul.hashicorp.com/connect-inject": "true"
    "consul.hashicorp.com/connect-service": {{ include "ixn.consulIxnVQNodeName" . | quote }}
    "consul.hashicorp.com/connect-service-port": vqnode-default

  prometheus:
    monitoringService:
      # -- Enable a service with Prometheus annotations for metrics scraping
      enabled: true

  livenessProbe:
    # -- Interaction Server VQ Node liveness probe initial delay
    initialDelaySeconds: 15
    # -- Interaction Server VQ Node liveness probe check period
    periodSeconds: 30
    # -- Interaction Server VQ Node liveness probe timeout
    timeoutSeconds: 3
    # -- Interaction Server VQ Node liveness probe failure threshold
    failureThreshold: 3

  readinessProbe:
    # -- Interaction Server VQ Node readiness probe initial delay
    initialDelaySeconds: 15
    # -- Interaction Server VQ Node readiness probe check period
    periodSeconds: 30
    # -- Interaction Server VQ Node readiness probe timeout
    timeoutSeconds: 3
    # -- Interaction Server VQ Node readiness probe failure threshold
    failureThreshold: 3
```

```
  # -- Volumes provided to Interaction Server VQ Node pod
  # Must be declared with starting `\|-`
  # since they are parsed as a template
  volumes : |-
    - name: kafka-shared-secret
      secret:
        secretName: kafka-shared-secret

  # -- Volumes mounted into an Interaction Server VQ Node container
  volumeMounts:
    kafka-shared-secret:
      readOnly: true
      mountPath: "/mnt/env-secrets/kafka-secrets"

tenant:
  # -- Tenant UUID or GWS ID
  id: "${TENANT_UUID}"
  # -- Tenant short ID
  sid: ${TENANT_ID}

# -- Replica count. Applied to both Interaction Service and Interaction Server VQ node pods.
# Can only be 1 or 0. Other values are not supported
replicaCount: 1
```

Replace some parameters-placeholders in this file with proper values. Adjust and copy/paste shell variables below (example, might be different in your environment):

```
export TENANT_ID=100
export TENANT_UUID=9350e2fc-a1dd-4c65-8d40-1f75a2e080dd
export POSTGRES_USER=postgres
export POSTGRES_PASSWORD=password
export REDIS_ADDR=infra-redis-redis-cluster.infra
export REDIS_PORT=6379
```

Now, substitute placeholders with these values in override_values.yaml:

```
envsubst  override_values.yaml_
```

Note: It creates a separate file “override_values.yaml_” that you will use in deployment. 3. Validate the Helm chart and provided values:

```
$ helm template ixn-{short-tenant-id} /ixn --version={version} -f override_values.yaml_
```

4. Install the Interaction Server chart, using the override values file:

```
$ helm upgrade --install ixn-{short-tenant-id} /ixn --version={version} -f
override_values.yaml_
```

5. Wait until all containers are ready. There should be 4/4 (5/5 if a logging sidecar enabled) for ixn-*-sts-0 and 3/3 containers for ixn-*-vqnode. If it is 1/1, it usually means something is wrong with the consul sidecar injection.

If the following error appeared: "*line 5: exec: /home/genesys/interaction_server/interaction_server_64: cannot execute: Permission denied*", ixn-{short-tenant-id}-sts-0 pod restart may be required if service account policy was applied after pod started). Refer to Service account.

```
kubectl get pods
NAME                                              READY   STATUS    RESTARTS   AGE
ixn-{short-tenant-id}-sts-0                       4/4     Running  0          8m1s
ixn-{short-tenant-id}-vqnode-deploy-6d8bc6846d-ml49d  3/3     Running  0          21m
```

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

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

```
$ helm show values /ixn > override_values.yaml_
```

Log storage

The following is a log storage example configuration in IXN Helm values:

```
ixnService:
  ixnServer:
    logStorage:
      mountPath: "/mnt/logs"
      storageSize: 1Gi
      storageClassName:
      volume:
        emptyDir: {}
```

Consul connection

Consul connection can be configured in several ways:

```
ixnService:
  ixnNode:
    consul:
      host:
        value:
      port:
      sslMode: false
```

```
ixnService:
  ixnNode:
    consul:
      host:
        valueFrom:
          fieldRef:
            fieldPath: status.hostIP
      port:
      sslMode: false
```

Connection to Configuration Server using Consul

```
ixnService:
  annotations: |-
      "consul.hashicorp.com/connect-inject": "true"
      "consul.hashicorp.com/connect-service": {{ include "ixn.consulIxnServerName" . | quote
}}
      "consul.hashicorp.com/connect-service-port": server-default
      "consul.hashicorp.com/connect-service-upstreams": |-
        voice-config:{{ .Values.ixnService.ixnNode.configNode.port }},
        {{ printf "tenant-%s:%d" .Values.tenant.id (int
.Values.ixnService.ixnServer.confServer.port) }}
      consul.hashicorp.com/service-tags: 'service-ixn'
      consul.hashicorp.com/service-meta-tenant-id: {{ .Values.tenant.id }}
      consul.hashicorp.com/service-meta-tenant-sid: {{ .Values.tenant.sid | quote }}
  ixnServer:
    confServer:
      host: "localhost"
```

```
      port: 8888
      appName: InteractionServer
  ixnNode:
    configNode:
      host: "localhost"
      port: 11100
ixnVQNode:
  annotations: |-
    "consul.hashicorp.com/connect-inject": "true"
    "consul.hashicorp.com/connect-service": {{ include "ixn.consulIxnVQNodeName" . | quote }}
    "consul.hashicorp.com/connect-service-port": vqnode-default
```

Volume mounts

Volume mounts example:

```
ixnService:
  volumes: |-
    - name: redis-ors-secret
      secret:
        secretName: redis-ors-secret
    - name: kafka-shared-secret
      secret:
        secretName: kafka-shared-secret
  ixnServer:
    volumeMounts:
      kafka-shared-secret:
        readOnly: true
        mountPath: "/mnt/env-secrets/kafka-secrets"
  ixnNode:
    volumeMounts:
      redis-ors-secret:
        readOnly: true
        mountPath: "/mnt/env-secrets/redis-secrets"
  ixnVQNode:
    volumes : |-
      - name: kafka-shared-secret
        secret:
          secretName: kafka-shared-secret
    volumeMounts:
      kafka-shared-secret:
        readOnly: true
        mountPath: "/mnt/env-secrets/kafka-secrets"
```

Configure monitoring and logging

To configure monitoring parameters in the Helm values file, see Monitoring.

To configure logging parameters in the Helm values file, see Logging.

Validate the deployment

There must be two pods. Each pod must be in a Running state and all READY checks should pass.