Digital Channels Private Edition Guide

Deploy Digital Channels

8/13/2025

Contents

Learn how to deploy Digital Channels 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.

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

Prepare your environment

To prepare your environment for the deployment, complete the steps in this section for either Google Kubernetes Engine (GKE) or Azure Kubernetes Service (AKS).

GKE

Log in to the GKE cluster from the host where you will run the deployment:

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

AKS

Log in to the GKE cluster from the host where you will run the deployment:

```
az aks get-credentials --resource-group  --name  --admin
```

Create a JSON file called **create-nexus-namespace.json** with the following content:

```
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "nexus",
    "labels": {
      "name": "nexus"
    }
  }
}
```

Use the JSON file to create a new namespace for Digital Channels:

```
kubectl apply -f apply create-nexus-namespace.json
```

Now, confirm the created namespace:

```
kubectl describe namespace nexus
```

Add Helm repo and execute Helm upgrade to create persistent volumes and persistent volume claims

```
helm repo add nexushelmrepo https://pureengage.jfrog.io/artifactory/helm-dev   --
username={jfrog_user} --password={jfrog_token}
```

Configure a secret to access JFrog

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

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

Deploy

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

You must override the following key sections in **values.yaml**:

- image.*
- nexus.fqdn

- nexus.redis.*
- nexus.db.*
- ingress.*

Here's an example of how your **values.yaml** file might look:

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

Run the following command to install Digital Channels:

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

Validate the deployment

To validate the deployment, send the following GET request:

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

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

The response should look like this:


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


The deployment was successful if `state="green"`. You can also confirm that `db.ready=true` and `cache.ready=true`.

Next steps

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

Provisioning overview for details.