Universal Contact Service Private Edition Guide

Configure UCS

12/9/2022

Contents

Learn how to configure Universal Contact Service (UCS).

Related documentation:

-
-
-

RSS:

- For private edition

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=
--docker-email=
```

Now map the secret to the default service account:

```
kubectl secrets link default  --for=pull
```

Override Helm chart values

Create a file named **values.yaml** and set the following values depending on your environment.

- Set the number of running PODs:

```
replicaCount: 2
```

- Set the repository for the images:

```
image:
  registry: base address of UCS image in artifactory.
  repository: ucsx/ucsx
  pullPolicy: IfNotPresent
  pullSecrets: if needed, set to the appropriate value for your environment.
```

- Set the ElasticSearch url, for example, http://ucsx-es-client-service.ucsx.svc.cluster.local:9200:

```
elasticsearch:
  url: Set URL to the ElasticSearch for data
```

- Set the Authentication service information

```
gauth:
  auth:
    url: URL to Genesys Auth service
  env:
    url: URL to GWS Environment service
```

- Set the memory and CPU limits to the values required for your deployment:

```
resources:
  requests:
    memory: "500Mi"
    cpu: "300m"
  limits:
    memory: "1000Mi"
    cpu: "2000m"
```

- Modify the DNS Configuration to match your environment:

```
dnsConfig:
  options:
    - name: ndots
      value: "3"
```

- UCS requires stickiness for some scenarios (from GWS/WWE). You can enable this on the Service level or create ingress rules to enable and configure them.
- The Ingress configuration requires the sticky sessions. You can enable this on the Service level or create ingress rules to enable and configure them. The cookie name should be set to *UCS_SESSIONID*.

If you want to use arbitrary UIDs in your OpenShift deployment, you must override the **securityContext** settings in the **values.yaml** file, so that no user or group IDs are specified. For details, see OpenShift security settings.

Configure Kubernetes

Create a Kubernetes ConfigMap named **ucsx-config** and save the database parameters under the following keys:

- `CMX_MASTER_DB_HOST` - the FQDN of the host where PostgresDB server is running
- `CMX_MASTER_DB_NAME` - the database name
- `CMX_MASTER_DB_PORT` - The port number of the PostgresDB server
- `CMX_MASTER_DB_USER` - the database user

Create a Kubernetes Secret named **ucsx-secret** and save the following secrets under the following keys:

- `CMX_MASTER_DB_PASSWORD` - the password for the database user to access the database
- `CMX_GWS_SERVICE_CREDENTIALS_CLIENT_ID` - the Client ID for GWS Auth
- `CMX_GWS_SERVICE_CREDENTIALS_CLIENT_SECRET` - the Client Secret for GWS Auth

The ConfigMap and Secret can also be created automatically from Helm Chart if the following values are empty:

- `existingSecret` - to create the Secret
- `existingConfig` - to create the ConfigMap

The following values will be added to the Secret:

- `db.password`
- `gauth.auth.clientId`
- `gauth.auth.clientSecret`

The following values will be added to the ConfigMap:

- `db.host`
- `db.name`
- `db.port`
- `db.user`

Configure security

Universal Contact Service (UCS) requires **clientId** and **clientSecret** registered in the Auth Service. These have to be provided during helmchart deployment.

Arbitrary UIDs in OpenShift

If you want to use arbitrary UIDs in your OpenShift deployment, you must override the **securityContext** settings, so that you do not define any specific IDs.

```
securityContext:
  runAsNonRoot: true
  runAsUser: null
  runAsGroup: 0
  fsGroup: null

containerSecurityContext: {}
```