



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Designer Private Edition Guide

Designer on GKE

7/27/2024

Contents

- 1 Configure a secret to access JFrog
- 2 Create a Designer secret with GWS
 - 2.1 GWS settings for auth
- 3 Checking logs

Learn more about specific settings that you have to configure when deploying Designer on Google Kubernetes Engine (GKE).

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Important

Configure and deploy Designer as described in the topics under the **Configure and deploy** section. Only additional information that is specific to deploying Designer on GKE is provided here.

Configure a secret to access JFrog

If you haven't done so already, create a secret for accessing the JFrog registry (for example, `jfrog-stage-credentials`):

```
kubectl create secret docker-registry jfrog-stage-credentials \
--docker-server=pureengage-docker-staging.jfrog.io \
--docker-username= \
--docker-password= \
--docker-email=
```

Now map the secret to the default service account:

```
kubectl secrets link default jfrog-stage-credentials --for=pull
```

Create a Designer secret with GWS

To create a Designer secret with GWS, update the following values to your Environment in the GWS service:

- `internalApi`: Set to the internal API of GWS
- `contactCenterIds`: Set to your tenant ID

-
- `redirectURIs`: Set to the URL(s) to be used for Designer
 - `:`: Set to the domain address for the environment

And, optionally, update the following:

- `-u opsAdmin: opsPass` (this is the default delivered for tenants)
- `client_secret`: Set to any value (used in secret)
- `name`: Set to anything
- `client_id`: Set to your client ID (used in secret)

GWS settings for auth

In the Designer flowsettings override file, update the following options with these values:

- `htccserver`: `gws-service-proxy.gws.svc.cluster.local`
- `gwsenvurl`: `http://gauth-environment.gauth.svc.cluster.local:80`
- `gwsauthurl`: `http://gauth-auth.gauth.svc.cluster.local:8`
- `ssoLoginUrl`: `https://gauth.apps.`

Checking logs

After deploying Designer, you check the logs using the following commands:

Designer

```
kubectl get pods
```

```
kubectl logs
```

DAS

```
kubectl get pods
```

```
kubectl logs
```