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.

Workspace Web Edition Private Edition Guide

Deploy Workspace Web Edition

2/15/2026

Contents

Learn how to deploy Workspace Web Edition (WWE) 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 Workspace Web Edition.

Deploying in GKE

Prerequisites for GKE

Secret configuration for pulling image

Connect to the cluster using Cloud SDK

Use the following command to connect to the cluster from the deployment host:

```
gcloud container clusters get-credentials  --zone  --project
```

Create the secret for accessing the jfrog registry

Use the following command to create the secret:

```
kubectl create secret docker-registry mycred
--docker-
--docker-username=
--docker-password=
--docker-email= -n wwe
```

Environment Preparation for GKE

Download the Helm charts

1. Download the WWE Helm charts from following repository: https://pureengageuse1.jfrog.io/ui/login/
2. Create the **override_values.yaml** with appropriate overrides from the following sample file for a sample deployment:

   ```
   context:
    envs:
    optimizedConfig: false
    gwsUrl: 'https://'
   ```

3. Enable and set Ingress with the appropriate hostname.
4. Set the value for **gwsUrl** applying the external gws url.

WWE installation on GKE

Log in to GKE cluster

Use the following command to connect to the GKE cluster using Cloud SDK from the deployment host:

```
gcloud container clusters get-credentials  --zone  --project
```

Create Namespace for WWE

Use the following command to create a new namespace for WWE:

```
kubectl create namespace wwe
```

Render the templates

To verify whether resources are getting created without issue, execute the following command to render templates without installing:

```
helm template --debug wwe ./wwe-nginx-9.0.5.tgz -f override_values.yaml -n wwe
```

Review the displayed Kubernetes descriptors. The values are generated from Helm templates and are based on settings from the **values.yaml** and **values-test.yaml** files. Ensure that no errors are displayed. Later, you will apply this configuration to your Kubernetes cluster.

Deploy WWE

Use the following command to deploy WWE:

```
helm install wwe ./wwe-nginx-9.0.5.tgz -f override_values.yaml -n wwe
```

This process takes several minutes. Wait until all objects are created and allocated, and the Kubernetes descriptors applied to the environment appear.

Verify the installation

Use the following command to check the installed Helm release:

```
helm list –all-namespaces
```

Use the following command to check the WWE objects created by Helm:

```
kubectl get all -n wwe
```

Verify that you can now access WWE at the following URL:

http://wwe.

Provisioning WWE Ingress on GKE

Create or download the wwe-ingress.yaml file

Use the following example template to create the **wwe-ingress.yaml** Ingress file for WWE. In this example template, the namespace is set specifically to **wwe**. Adjust the values needed for your deployment.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: wwe-ingress
  namespace: wwe
  annotations:
    # add an annotation indicating the issuer to use.
    cert-manager.io/cluster-issuer: "selfsigned-cluster-issuer"
    # Custom annotations for NGINX Ingress Controller
    kubernetes.io/ingress.class: "nginx"
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
    nginx.ingress.kubernetes.io/use-regex: "true"
spec:
  rules:
  - host: wwe.test.dev
    http:
      paths:
        - path: /.*
          backend:
            serviceName:  wwe-wwe-nginx
            servicePort: 80
  tls:
  - hosts:
    - wwe.test.dev
    secretName: wwe-ingress-cert
```

Apply the yaml file to your namespace

Use the following command to apply the yaml file to your namespace:

```
kubectl apply -f wwe-ingress.yaml -n wwe
```

Deploy in AKS

Prerequisites

Secret configuration for pulling image

Use the following commands to create the Secret for accessing the jfrog registry and map the secret to the default account:

```
kubectl create secret docker-registry mycred --docker-server=pureengageuse1-docker-multicloud.jfrog.io --docker-username= --docker-password= --docker-email=
```

Install the azure-cli based in you OS environment

Follow the instructions found in the following website to install the Azure CLI:

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

Login to AKS cluster

```
$ az login
```

Connect to cluster

Use the following command to log in to the cluster from the deployment host:

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

Environment preparation

Create Namespace for WWE

Use the following command to create a new namespace for WWE:

```
$ kubectl create namespace wwe
```

Download the Helm charts

Download the WWE Helm charts from following repository: https://pureengageuse1.jfrog.io/ui/login/

Create the override file

Create the **override_values.yaml** file using the appropriate overrides based on the following sample:

```
context:
          envs:
          optimizedConfig: false
          gwsUrl: 'https://'
```

Note: Enable **ingress** and set it with an appropriate hostname. The value for **gwsUrl** must be set with the external gws url.

WWE Installation

Render the templates

To verify whether resources are getting created without issue, execute the following command to render templates without installing:

```
$ helm template --debug wwe ./wwe-nginx-9.0.5.tgz -f override_values.yaml -n wwe
```

Review the displayed Kubernetes descriptors. The values are generated from Helm templates and are based on settings from the **values.yaml** and **values-test.yaml** files. Ensure that no errors are displayed. Later, you will apply this configuration to your Kubernetes cluster.

Deploy WWE

Use the following command to deploy WWE:

```
$ helm install wwe ./wwe-nginx-9.0.5.tgz -f override_values.yaml -n wwe
```

This process takes several minutes. Wait until all objects are created and allocated, and the Kubernetes descriptors applied to the environment appear.

Verify the installation

Use the following command to check the installed Helm release:

```
$ helm list –all-namespaces
```

Use the following command to check the WWE objects created by Helm:

```
$ kubectl get all -n wwe
```

Execute the following helm status command:

```
$ kubectl status wwe -n wwe

LAST DEPLOYED: Mon Jun 20 10:21:25 2022

NAMESPACE: wwe

STATUS: deployed

REVISION: 2

TEST SUITE: None
```

Verify that you can now access wwe at the following URL:

http://wwe.