**Bold BI on Kubernetes**

Bold BI can be deployed on Kubernetes cluster. You can create Kubernetes cluster on either cloud or on-premise infrastructure. After completing cluster creation and connect to it, you can download the configuration files [here](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy). This directory includes configuration YAML files which contains all the configuration settings needed to deploy Bold BI on Kubernetes cluster. The following links explain Bold BI Kubernetes deployment in a specific cloud and on-premise environments.

* [Google Kubernetes Engine (GKE)](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/google-gke.md)
* [Amazon Elastic Kubernetes Service (EKS)](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/amazon-eks.md)
* [Azure Kubernetes Service (AKS)](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/microsoft-aks.md)
* [On-premise](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/on-premise.md)

**Bold BI on Google Kubernetes Engine**

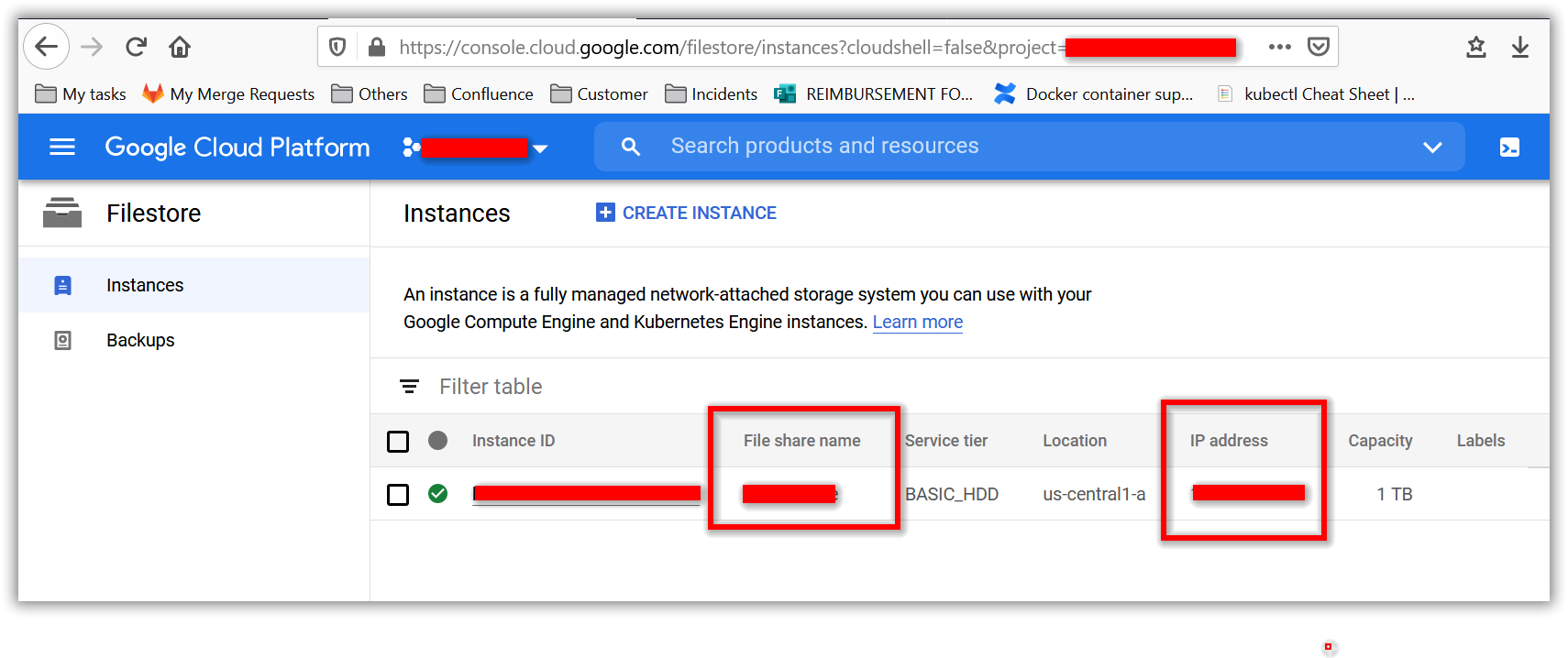
Please follow the below steps to deploy Bold BI On-Premise in Google Kubernetes Engine (GKE).

1. Download the following files for Bold BI deployment in GKE,
   * [pvclaim\_gke.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/pvclaim_gke.yaml)
   * [deployment.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/deployment.yaml)
   * [hpa\_gke.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/hpa_gke.yaml)
   * [service.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/service.yaml)
   * [ingress.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/ingress.yaml)
2. Create a Kubernetes cluster in Google Cloud Platform (GCP) to deploy the Bold BI On-Premise application.

<https://console.cloud.google.com/kubernetes>

1. Create a Google filestore instance to store the shared folders for applications’ usage.

<https://console.cloud.google.com/filestore>

1. Note the **File share name** and **IP address** after creating filestore instance, [](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/gke_file_share_details.png)
2. Open **pvclaim\_gke.yaml** file, downloaded in **Step 1**. Replace the **File share name** and **IP address** noted in above step to the <file\_share\_name> and <file\_share\_ip\_address> places in the file. You can also change the storage size in the YAML file. Save the file once you replaced the file share name and file share IP address.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/gke_pvclaim.png)

1. Set your project and newly created cluster in Google cloud shell,

<https://cloud.google.com/kubernetes-engine/docs/quickstart>

1. Deploy the latest Nginx ingress controller to your cluster using the following command,

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.41.2/deploy/static/provider/cloud/deploy.yaml

1. Navigate to the folder where the deployment files were downloaded from **Step 1**.
2. If you have a DNS to map with the application you can continue with the following steps, else skip to **Step 14**.
3. Open the **ingress.yaml** file. Uncomment the host value and replace your DNS hostname with example.com and save the file.
4. If you have the SSL certificate for your DNS and need to configure the site with your SSL certificate, follow the below step or you can skip to **Step 14**.
5. Run the following command to create a TLS secret with your SSL certificate,

kubectl create secret tls boldbi-tls --key <key-path> --cert <certificate-path>

1. Now uncomment the tls section and replace your DNS hostname with example.com in ingress spec and save the file.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_yaml.png)

1. Run the following command for applying the Bold BI ingress to get the IP address of nginx ingress,

kubectl apply -f ingress.yaml

1. Now run the following command to get the ingress IP address,

kubectl get ingress

Repeat the above command till you get the IP address in ADDRESS tab like in below image. [](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_address.png)

1. Note the ingress IP address and map it with your DNS if you have added the DNS in **ingress.yaml** file. If you do not have the DNS and want to use the application, you can use the ingress IP address.
2. Open the **deployment.yaml** file from the downloaded files on **Step 1**. Replace your DNS or ingress IP address in <application\_base\_url> place.

Ex: http://example.com, https://example.com, http://<ingress\_ip\_address>

1. Read the optional client library license agreement from the following link,

[Consent to deploy client libraries](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/consent-to-deploy-client-libraries.md)

1. Note the optional client libraries from the above link as comma separated names and replace in <comma\_separated\_library\_names> place. Save the file after the required values has been replaced.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/deployment_yaml.png)

1. Now run the following commands one by one,

kubectl apply -f pvclaim\_gke.yaml

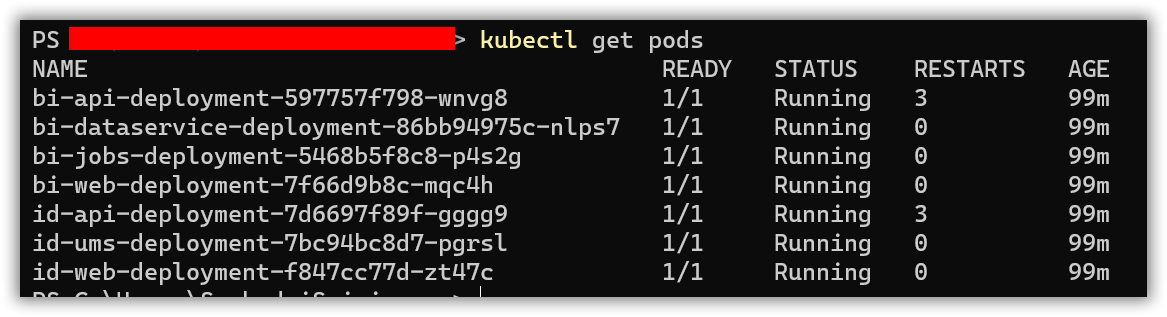
kubectl apply -f deployment.yaml

kubectl apply -f hpa\_gke.yaml

kubectl apply -f service.yaml

1. Now wait for some time till the Bold BI On-Premise application deployed to your Google Kubernetes cluster.
2. Use the following command to get the pods’ status,

kubectl get pods

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/pod_status.png)

1. Wait till you see the applications in running state. Then use your DNS or ingress IP address you got from **Step 15** to access the application in browser.
2. Configure the Bold BI On-Premise application startup to use the application. Please refer the following link for more details on configuring the application startup,

<https://help.boldbi.com/embedded-bi/application-startup>

**Bold BI on Amazon Elastic Kubernetes Service**

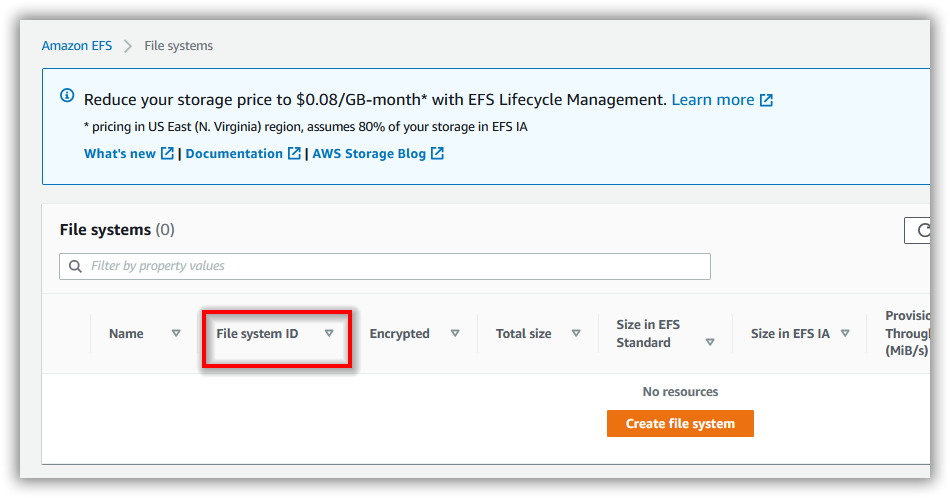
Please follow the below steps to deploy Bold BI On-Premise in Amazon Elastic Kubernetes Service (Amazon EKS).

1. Download the following files for Bold BI deployment in Amazon EKS,
   * [pvclaim\_eks.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/pvclaim_eks.yaml)
   * [deployment.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/deployment.yaml)
   * [hpa.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/hpa.yaml)
   * [service.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/service.yaml)
   * [ingress.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/ingress.yaml)
2. Create a Kubernetes cluster in Amazon EKS to deploy the Bold BI On-Premise application.

<https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html>

1. Create an Amazon Elastic File System (EFS) volume to store the shared folders for applications’ usage.

<https://docs.aws.amazon.com/eks/latest/userguide/efs-csi.html>

1. Note the **File system ID** after creating EFS file system, [](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/aws-efs.png)
2. Open **pvclaim\_eks.yaml** file, downloaded in **Step 1**. Replace the **File system ID** noted in above step to the <efs\_file\_system\_id> place in the file. You can also change the storage size in the YAML file.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/eks_pvclaim.png)

1. Connect with your Amazon EKS cluster.
2. You can skip this step if your cluster already has a CNI (Container Network Interface) running. However, if your cluster does not have any CNI or if you face any CNI related issues when deploying, you can install the Calico CNI using the following command in your EKS cluster.

<https://docs.projectcalico.org/about/about-calico>

kubectl apply -f https://docs.projectcalico.org/v3.11/manifests/calico.yaml

1. Deploy the EFS CSI Driver to manage the lifecycle of Amazon EFS file systems in kubernetes containers,

<https://github.com/kubernetes-sigs/aws-efs-csi-driver>

kubectl apply -k "github.com/kubernetes-sigs/aws-efs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.0"

1. Deploy the latest Nginx ingress controller to your cluster using the following command,

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.41.2/deploy/static/provider/aws/deploy.yaml

1. Navigate to the folder where the deployment files were downloaded from **Step 1**.
2. If you have a DNS to map with the application, you can continue the with following steps, else skip to **Step 16**.
3. Open the **ingress.yaml** file. Uncomment the host value and replace your DNS hostname with example.com and save the file.
4. If you have the SSL certificate for your DNS and need to configure the site with your SSL certificate, follow the below step or you can skip to **Step 16**.
5. Run the following command to create a TLS secret with your SSL certificate,

kubectl create secret tls boldbi-tls --key <key-path> --cert <certificate-path>

1. Now uncomment the tls section and replace your DNS hostname with example.com in ingress spec and save the file.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_yaml.png)

1. Run the following command for applying the Bold BI ingress to get the IP address of nginx ingress,

kubectl apply -f ingress.yaml

1. Now run the following command to get the ingress IP address,

kubectl get ingress

Repeat the above command till you get the IP address in ADDRESS tab like in below image. [](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_address.png)

1. If you face any issues related to webhook while applying the Bold BI ingress, you can run the following command to remove the webhook validation in nginx ingress,

kubectl delete -A ValidatingWebhookConfiguration ingress-nginx-admission

1. Note the ingress IP address and map it with your DNS if you have added the DNS in **ingress.yaml** file. If you do not have the DNS and want to use the application, you can use the ingress IP address.
2. Open the **deployment.yaml** file from the downloaded files on **Step 1**. Replace your DNS or ingress IP address in <application\_base\_url> place.

Ex: http://example.com, https://example.com, http://<ingress\_ip\_address>

1. Read the optional client library license agreement from the following link,

[Consent to deploy client libraries](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/consent-to-deploy-client-libraries.md)

1. Note the optional client libraries from the above link as comma separated names and replace in <comma\_separated\_library\_names> place. Save the file after the required values has been replaced.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/deployment_yaml.png)

1. Now run the following commands one by one,

kubectl apply -f pvclaim\_eks.yaml

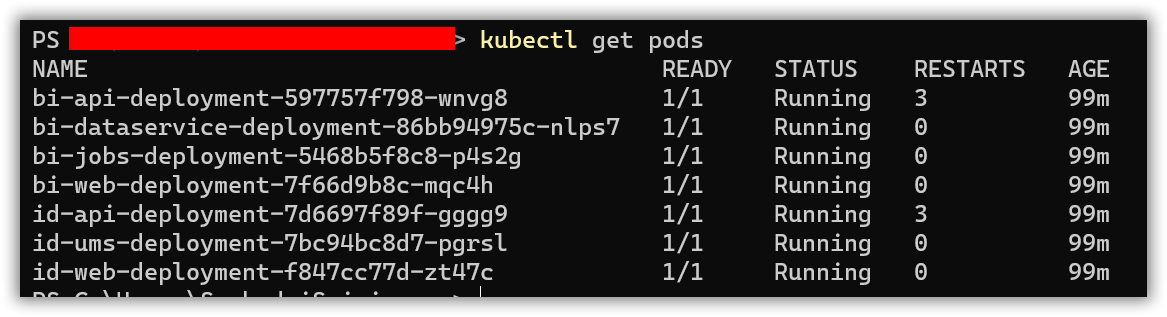
kubectl apply -f deployment.yaml

kubectl apply -f hpa.yaml

kubectl apply -f service.yaml

1. Now wait for some time till the Bold BI On-Premise application deployed to your Amazon EKS cluster.
2. Use the following command to get the pods’ status,

kubectl get pods

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/pod_status.png)

1. Wait till you see the applications in running state. Then use your DNS or ingress IP address you got from **Step 17** to access the application in browser.
2. Configure the Bold BI On-Premise application startup to use the application. Please refer the following link for more details on configuring the application startup,

<https://help.boldbi.com/embedded-bi/application-startup>

**Bold BI on Microsoft Azure Kubernetes Service**

Please follow the below steps to deploy Bold BI On-Premise in Microsoft Azure Kubernetes Service (AKS).

1. Download the following files for Bold BI deployment in AKS,
   * [pvclaim\_aks.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/pvclaim_aks.yaml)
   * [deployment.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/deployment.yaml)
   * [hpa.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/hpa.yaml)
   * [service.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/service.yaml)
   * [ingress.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/ingress.yaml)
2. Create a Kubernetes cluster in Microsoft Azure Kubernetes Service (AKS) to deploy the Bold BI On-Premise application.
3. Create a File share instance in your storage account and note the File share name to store the shared folders for applications’ usage.
4. Encode the storage account name and storage key in base64 format.
5. Open **pvclaim\_aks.yaml** file, downloaded in **Step 1**. Replace the **base64 encoded storage account name**, **base64 encoded storage account key** and **File share name** noted in above steps to <base64\_azurestorageaccountname>, <base64\_azurestorageaccountkey> and <file\_share\_name> places in the file respectively. You can also change the storage size in the YAML file.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/aks_pvclaim.png)

1. Open your Azure CLI and connect with your cluster.
2. Deploy the latest Nginx ingress controller to your cluster using the following command,

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.41.2/deploy/static/provider/cloud/deploy.yaml

1. Navigate to the folder where the deployment files were downloaded from **Step 1**.
2. If you have a DNS to map with the application you can continue with the following steps, else skip to **Step 14**.
3. Open the **ingress.yaml** file. Uncomment the host value and replace your DNS hostname with example.com and save the file.
4. If you have the SSL certificate for your DNS and need to configure the site with your SSL certificate, follow the below step or you can skip to **Step 14**.
5. Run the following command to create a TLS secret with your SSL certificate,

kubectl create secret tls boldbi-tls --key <key-path> --cert <certificate-path>

1. Now uncomment the tls section and replace your DNS hostname with example.com in ingress spec and save the file.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_yaml.png)

1. Run the following command for applying the Bold BI ingress to get the IP address of nginx ingress,

kubectl apply -f ingress.yaml

1. Now run the following command to get the ingress IP address,

kubectl get ingress

Repeat the above command till you get the IP address in ADDRESS tab like in below image. [](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_address.png)

1. Note the ingress IP address and map it with your DNS if you have added the DNS in **ingress.yaml** file. If you do not have the DNS and want to use the application, you can use the ingress IP address.
2. Open the **deployment.yaml** file from the downloaded files on **Step 1**. Replace your DNS or ingress IP address in <application\_base\_url> place.

Ex: http://example.com, https://example.com, http://<ingress\_ip\_address>

1. Read the optional client library license agreement from the following link,

[Consent to deploy client libraries](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/consent-to-deploy-client-libraries.md)

1. Note the optional client libraries from the above link as comma separated names and replace in <comma\_separated\_library\_names> place. Save the file after the required values has been replaced.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/deployment_yaml.png)

1. Now run the following commands one by one,

kubectl apply -f pvclaim\_aks.yaml

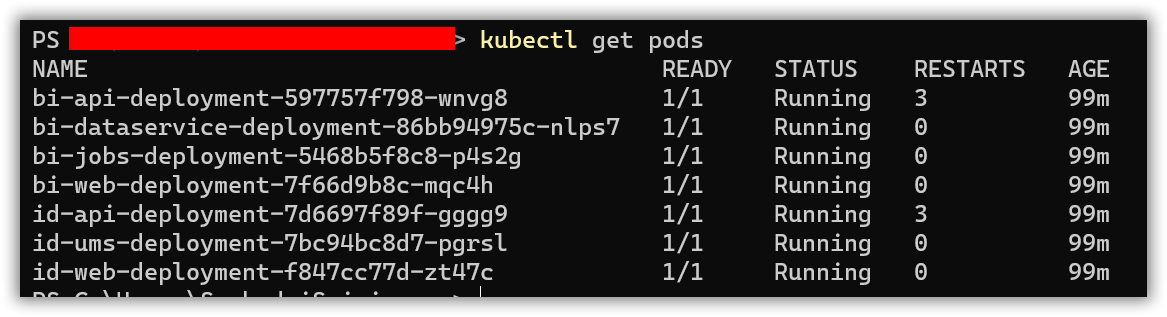
kubectl apply -f deployment.yaml

kubectl apply -f hpa.yaml

kubectl apply -f service.yaml

1. Now wait for some time till the Bold BI On-Premise application deployed to your Microsoft AKS cluster.
2. Use the following command to get the pods’ status,

kubectl get pods

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/pod_status.png)

1. Wait till you see the applications in running state. Then use your DNS or ingress IP address you got from **Step 15** to access the application in browser.
2. Configure the Bold BI On-Premise application startup to use the application. Please refer the following link for more details on configuring the application startup,

<https://help.boldbi.com/embedded-bi/application-startup>

**Bold BI on On-Premise Kubernetes Cluster**

Please follow the below steps to deploy Bold BI application in your On-Premise machine kubernetes cluster.

1. Download the following files for Bold BI deployment in On-Premise,
   * [pvclaim\_onpremise.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/pvclaim_onpremise.yaml)
   * [deployment.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/deployment.yaml)
   * [hpa.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/hpa.yaml)
   * [service.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/service.yaml)
   * [ingress.yaml](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/deploy/ingress.yaml)
2. Create a folder in your machine to store the shared folders for applications’ usage.

Ex: D://app/shared

1. Open **pvclaim\_onpremise.yaml** file, downloaded in above **Step 1**. Replace the shared folder path in your host machine to the <local\_directory> place in the file. You can also change the storage size in the YAML file.

Ex: D://app/shared -> /run/desktop/mnt/host/**d/app/shared**

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/onpremise_pvclaim.png)

1. Deploy the latest Nginx ingress controller to your cluster using the following command,

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.41.2/deploy/static/provider/cloud/deploy.yaml

1. Map the DNS to your machine IP address in which you want to access the application.
2. Navigate to the folder where the deployment files were downloaded from **Step 1**.
3. Open the **ingress.yaml** file. Uncomment the host value and replace your DNS hostname with example.com and save the file.
4. If you have the SSL certificate for your DNS and need to configure the site with your SSL certificate, run the following command to create a TLS secret with your SSL certificate,

kubectl create secret tls boldbi-tls --key <key-path> --cert <certificate-path>

1. Now uncomment the tls section and replace your DNS hostname with example.com in ingress spec and save the file.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/ingress_yaml.png)

1. Open the **deployment.yaml** file from the downloaded files on **Step 1**. Replace your DNS in <application\_base\_url> place.

Ex: http://example.com, https://example.com

1. Read the optional client library license agreement from the following link,

[Consent to deploy client libraries](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/consent-to-deploy-client-libraries.md)

1. Note the optional client libraries from the above link as comma separated names and replace in <comma\_separated\_library\_names> place. Save the file after the required values has been replaced.

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/deployment_yaml.png)

1. Now run the following commands one by one,

kubectl apply -f pvclaim\_onpremise.yaml

kubectl apply -f deployment.yaml

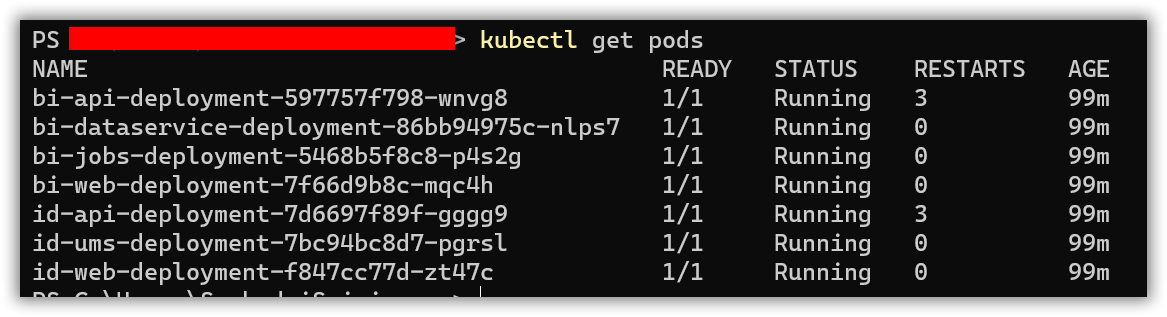
kubectl apply -f hpa.yaml

kubectl apply -f service.yaml

kubectl apply -f ingress.yaml

1. Now wait for some time till the Bold BI On-Premise application deployed to your On-Premise kubernetes cluster.
2. Use the following command to get the pods’ status,

kubectl get pods

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/pod_status.png)

1. After deployment wait for some time to Horizontal Pod Autoscaler (HPA) gets the metrics from pods. Use the following command to get HPA status,

kubectl get hpa

If you get <unknown>/80% instead of actual CPU and memory usage of pods, then you do not have any metrics server running inside your cluster. Use the following command to deploy metrics server in your cluster to enable the autoscaling feature by HPA.

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.3.7/components.yaml

1. Use your DNS hostname to access the application in browser.
2. Configure the Bold BI On-Premise application startup to use the application. Please refer the following link for more details on configuring the application startup,

<https://help.boldbi.com/embedded-bi/application-startup>

# Consent to deploy client libraries

By giving consent to install client libraries to connect with Oracle, PostgreSQL, MySQL, MongoDB, InfluxDB and Snowflake.Data, you can use the following libraries in your kubernetes pods. Bold BI uses these client libraries to connect with their respective SQL database variants. Read about the licenses of each library to give consent for usage.

## mongo-csharp-driver

* MongoDB

[Apache License, Version 2.0](https://github.com/mongodb/mongo-csharp-driver/blob/master/License.txt)

## Snowflake.Data

* Snowflake.Data

[Apache License, Version 2.0](https://github.com/snowflakedb/snowflake-connector-net/blob/master/LICENSE)

## Oracle.ManagedDataAccess

* Oracle

[Oracle License](https://www.oracle.com/downloads/licenses/distribution-license.html)

## Npgsql 4.0.0

* PostgreSQL
* Amazon Redshift
* Google Cloud - PostgreSQL
* Amazon Aurora - PostgreSQL

[PostgreSQL License](https://github.com/npgsql/npgsql/blob/main/LICENSE)

## MySQLConnector 0.45.1

* MySQL
* MemSQL
* MariaDB
* Google Cloud – MySQL
* Amazon Aurora - MySQL
* CDATA

[MIT License](https://github.com/mysql-net/MySqlConnector/blob/master/LICENSE)

## InfluxData.Net

* InfluxDB

[MIT License](https://github.com/pootzko/InfluxData.Net/blob/master/LICENSE)

## PhantomJS WebKit

PhantomJS is a headless WebKit scriptable with JavaScript. It is a free software/open source that may contain MIT, BSD, LGPL, GPL, or other similar licenses. It contains third-party code. This executable file is necessary to achieve Image and PDF export functionalities in dashboard, widgets, and schedules. Without this file, the Image and PDF export options in dashboard, widgets, and schedules will no longer be available. It is your decision if you choose to download Phantom JS, but you must accept all of its terms and conditions if you want to use it with Syncfusion’s products.

You can read the [License](https://github.com/ariya/phantomjs/blob/master/LICENSE.BSD) and [Third-Party](https://github.com/ariya/phantomjs/blob/master/third-party.txt) documents.

# Client library names as arguments for Bold BI deployment in Kubernetes

Find the names of client libraries which needs to be passed as a comma separated string for an environment variable in **deployment.yaml** file.

| **Library** | **Name** |
| --- | --- |
| mongo-csharp-driver | mongodb |
| Snowflake.Data | snowflake |
| Oracle.ManagedDataAccess | oracle |
| Npgsql 4.0.0 | npgsql |
| MySQLConnector 0.45.1 | mysql |
| InfluxData.Net | influxdb |
| PhantomJS WebKit | phantomjs |

If you want to use all client libraries in the Bold BI application, then pass the following string as value for INSTALL\_OPTIONAL\_LIBS environment variable. You need to add the names only for the libraries which you are consenting to use with Bold BI application.

phantomjs,mongodb,mysql,influxdb,snowflake,oracle,npgsql

[](https://github.com/boldbi/boldbi-kubernetes/blob/3.3.40_dev/docs/images/client-library.png)