

# Portainer Web GUI - Environment Connection

With my K8 cluster now set up, I'm going to connect this cluster to my Portainer container running in my local Docker engine on another VM I have in Proxmox. For more details about my Portainer set up, check out my [Docker Containers](#) book. Portainer is a Web GUI container you can use for Kubernetes and Docker management, and it'll allow me to manage and deploy containers and clusters. See below to view how I set up my connection:

## Portainer Agent

To connect this cluster, I'll be deploying a Portainer Agent onto the cluster. To do so, run the following command in any node of your cluster:

```
kubectl apply -f https://downloads.portainer.io/ce2-19/portainer-agent-k8s-nodeport.yaml
```

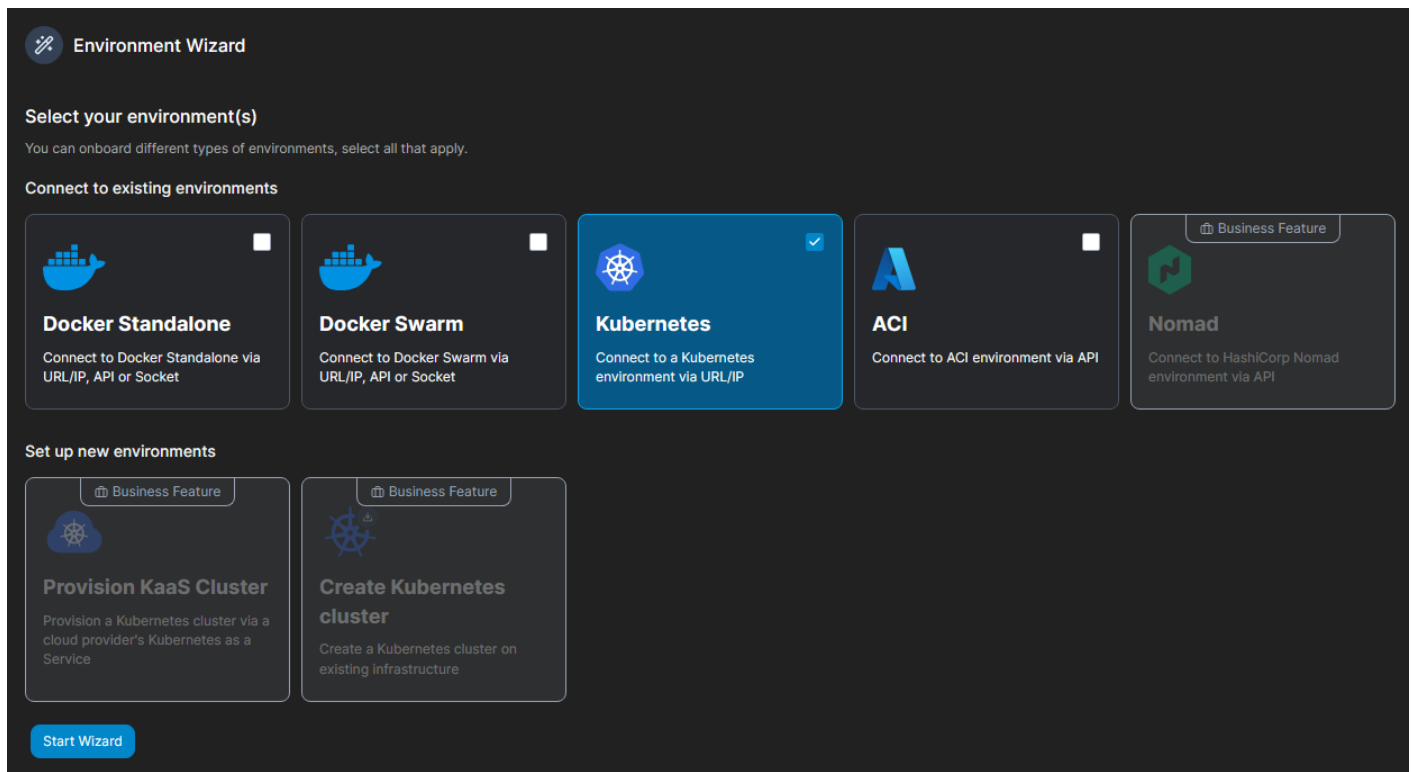
```
austin@k-ctrlr:~$ kubectl apply -f https://downloads.portainer.io/ce2-19/portainer-agent-k8s-nodeport.yaml
namespace/portainer created
serviceaccount/portainer-sa-clusteradmin created
clusterrolebinding.rbac.authorization.k8s.io/portainer-crb-clusteradmin created
service/portainer-agent created
service/portainer-agent-headless created
deployment.apps/portainer-agent created
austin@k-ctrlr:~$
```

With this installed, check the status of the agent by running the following command:

```
kubectl get pods --namespace=portainer
```

```
austin@k-ctrlr:~$ kubectl get pods --namespace=portainer
NAME                                READY   STATUS    RESTARTS   AGE
portainer-agent-548d57f7b5-2d6cj    1/1     Running   0           77s
austin@k-ctrlr:~$
```

Now, open up your Portainer GUI in the web browser and navigate to Environment ---> Add. Select Kubernetes and start the wizard:



Next, enter your node IP address and specify port 30778. I'm using NordPort so 30778 applies; if you're using load balancer then use port 9001:

Environment Wizard

1Kubernetes

Connect to your Kubernetes environment

Agent

Edge Agent Standard

Business Feature

Import

Import an existing Kubernetes config

Information

Ensure that you have deployed the Portainer agent in your cluster first. Refer to the platform related command below to deploy it.

Kubernetes via load balancer

Kubernetes via node port

```
kubectl apply -f https://downloads.portainer.io/ce2-19/portainer-agent-k8s-nodeport.yaml
```

Copy command

Name\*

k8-dev

Environment address\*

192.168.2.39:30778

> More settings

Connect

Previous

Close

Your cluster should now connect, load, and be added to your dashboard:

portainer.io  
COMMUNITY EDITION

Home

local

Dashboard

App Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

Settings

Users

Environments

Registries

Authentication logs

Notifications

Settings

Home

Latest News From Portainer

Environments

Search by name, group, tag, status, URL...

Refresh

Click on an environment to manage

Platform

Connection Type

Status

Tags

Groups

Agent Version

Clear all

Sort By

Items per page 10

local

Up

2024-04-04 21:38:14

Standalone 26.0.0

/var/run/docker.sock

Group: Unassigned

No tags

Local

0 stacks

3 containers

2

1

0

0

2 volumes

6 images

56 CPU

67.4 GB RAM

Disconnect

Connected

k8s-dev

Up

2024-04-04 21:38:14

Kubernetes v1.29.3

192.168.2.39:30778

Group: Unassigned

No tags

Agent

2.19.4

10 CPU

16.4 GB RAM

4 nodes

Live connect

Disconnected

Select "Live Connect" to view your cluster resources:

The cluster is now connected to Portainer and can be managed via the Web GUI!

---

Revision #3

Created 5 April 2024 02:29:13 by Austin

Updated 7 April 2024 23:21:25 by Austin