

ABLESTACK

# ABLESTACK Kubernetes Cluster와 Portainer 플랫폼 연동

# ABLESTACK Kubernetes Cluster와 Portainer 플랫폼 연동

ABLESTACK은 Koral을 통해 Kubernetes Cluster를 자동으로 생성하는 클러스터 관리 기능을 제공합니다. 복잡한 쿠버네티스 클러스터를 빠르게, 자동으로 배포하고 관리할 수 있다는 측면에서 많은 편리함을 제공하지만, 순수한 Kubernetes 클러스터이기 때문에 사용자 입장에서 대부분의 작업을 kubectl CLI 도구를 사용해야 하는 어려움이 있습니다. 때문에 ABLESTACK Koral 뿐 아니라 대부분의 순수한 Kubernetes 클러스터 사용자는 보다 편리하고 직관적인 관리도구와 플랫폼을 통해 애플리케이션 배포 작업을 할 수 있는 환경을 고민합니다.

본 문서는 Kubernetes Cluster를 사용하는 사용자가 보다 직관적인 UI를 통해 애플리케이션을 관리할 수 있도록 도와주는 오픈소스 플랫폼인 "Portainer"와의 연동을 통해 편리하게 클러스터를 관리할 수 있는 방법을 소개합니다.

## Portainer 소개

Portainer는 컨테이너 오케스트레이션의 운영 복잡성을 줄이는 데 도움이 되는 Full Stack 쿠버네티스 관리 플랫폼입니다.

직관적인 사용자 인터페이스와 강력한 기능을 갖춘 Portainer를 사용하면 조직에 필요한 제어 및 거버넌스를 유지하면서 쿠버네티스 클러스터를 빠르고 안정적으로 프로비저닝, 배포 및 관리 할 수 있습니다.

Portainer는 Kubernetes 외에도 Docker, Podman, Docker Swarm 등을 관리할 수 있으며 Community Edition의 경우 Kubernetes 클러스터에 대해 다음의 기능을 제공합니다.

- 멀티 클러스터 관리
- 볼륨 관리
- 네트워크 관리
- CPU/Memory 할당 관리
- API 게이트웨이
- 커스텀 템플릿 관리
- Helm Chart 자동 배포

위와 같은 기능을 제공하는 Portainer는 ABLESTACK Koral의 쿠버네티스 클러스터와 연계하면 편리하게 원하는 애플리케이션이나 플랫폼을 배포할 수 있습니다.

본 문서는 Portainer를 Koral 쿠버네티스 클러스터에 배포하고, 멀티 클러스터를 관리하며, 다양한 애플리케이션을 배포하는 과정을 설명합니다. 배포 순서는 다음과 같습니다.

1. Kubernetes 클러스터용 네트워크 생성
2. Kubernetes Cluster 생성
3. Persistent Volume용 NFS 스토리지 생성
4. kubectl을 통한 연결 확인
5. NFS CSI 드라이버 설치
6. 디폴트 스토리지 클래스 생성
7. Portainer 배포 및 연결 확인
8. 샘플 컨테이너 배포 및 테스트
9. 멀티 클러스터 관리

사용자의 컴퓨터에 Kubernetes CLI(kubectl)을 설치한 후 각 단계를 진행합니다.

## Kubernetes용 네트워크 생성

위와 같은 기능을 제공하는 Portainer는 ABLESTACK Koral의 쿠버네티스 클러스터와 연계하면 편리하게 원하는 애플리케이션이나 플랫폼을 배포할 수 있습니다.

본 문서는 Portainer를 Koral 쿠버네티스 클러스터에 배포하고, 멀티 클러스터를 관리하며, 다양한 애플리케이션을 배포하는 과정을 설명합니다. 배포 순서는 다음과 같습니다.

"쿠버네티스 서비스에 대한 기본 네트워크오퍼링"은 ABLESTACK이 쿠버네티스 클러스터 생성을 위해 기본적으로 제공하는 네트워크 오퍼링입니다. 새로운 네트워크 오퍼링을 만드는 경우 반드시 다음의 조건을 만족하는 오퍼링을 생성해야 합니다.

- Isolated Network
- Virtual Router를 통해 다음의 서비스를 지원
  - Vpn, Dhcp, Dns, Firewall, Lb, UserData, SourceNat, StaticNat, PortForwading
- 기본 송신 정책: 허용

Mold의 "네트워크 > 가상머신용 네트워크"를 선택한 후 "네트워크 추가"를 눌러 네트워크 추가 대화상자를 표시한 후 다음과 같이 설정 값을 입력합니다.

네트워크 추가 ?
✕

---

Isolated
L2
shared

**\* 이름** ?

**설명** ?

**\* Zone** ?

**소유자 유형**

**\* 도메인**

**\* 계정**

**네트워크 도메인** ?

**\* 네트워크 오퍼링** ?

**외부 아이디** ?

**게이트웨이** ?

**넷마스크** ?

**DNS 1** ?

**DNS 2** ?

**서비스용 네트워크의 라우터에 대한 IPv4 주소** ?

"네트워크 추가" 대화 상자에서 다음의 항목을 확인합니다.

- 네트워크 오퍼링 : 쿠버네티스 서비스를 위한 네트워크 오퍼링, 또는 새로 생성한 쿠버네티스용 오퍼링 선택

설정 정보를 모두 입력한 후 "확인" 버튼을 클릭해서 네트워크를 생성합니다.

## Kubernetes Cluster 생성

"Mold에서 "컴퓨터 > 쿠버네티스"를 선택한 후 "쿠버네티스 클러스터 생성" 버튼을 클릭하여 클러스터 생성을 위한 설정 정보를 다음과 같이 입력합니다.

쿠버네티스 클러스터 생성 ?
✕

**\* 이름** ⓘ

**설명** ⓘ

**\* Zone** ⓘ

**\* 쿠버네티스 버전** ⓘ

**\* 컴퓨터 오퍼링** ⓘ

**노드 루트 디스크 크기(GB)** ⓘ

**네트워크** ⓘ

HA 활성화 됨

\* 마스터 노드 ⓘ

3

외부 로드 밸런서 IP 주소 ⓘ

external load balancer IP address while using shared network with Kubernetes HA cl...

\* 클러스터 크기 ⓘ

3

SSH 키 쌍 ⓘ

사실 레지스트리

취소 확인

"쿠버네티스 클러스터 생성" 대화상자에서 다음의 설정 항목을 확인합니다.

- 컴퓨트 오퍼링 : 클러스터를 구성하는 가상머신 노드의 컴퓨트 크기를 결정
- 노드 루트 디스크 크기 : 컨테이너는 일반적으로 노드의 루트디스크를 오버레이 디스크로 사용하므로 적절한 크기를 할당
- 네트워크 : 쿠버네티스 클러스터를 위해 생성한 네트워크 선택
- 마스터 노드 : HA가 활성화 되는 경우 [3], 그렇지 않은 경우 [1]
- 클러스터 크기 : 워커 노드의 수를 지정

모든 설정을 입력한 후 "확인" 버튼을 클릭하여 쿠버네티스 클러스터를 생성합니다.

## Persistent Volume용 NFS 스토리지 생성

Mold에서 "스토리지 > 공유 파일 시스템" 메뉴를 선택한 후 "공유 파일시스템 생성" 버튼을 클릭하여 "공유 파일시스템 생성" 대화상자를 열고 다음과 같이 설정 정보를 입력합니다.

공유 파일시스템 생성 ?
✕

**소유자 유형**

계정 ▼

\* 도메인

ROOT ▼

\* 계정

admin ▼

\* 이름

K8S Cluster 001 Storage

**설명**

K8S Cluster 001 Storage

\* Zone ⓘ

ABLESTACK-PoC ▼

**파일 시스템 ⓘ**

XFS ▼

\* 네트워크 ⓘ

쿠버네티스 서비스를 위한 격리 네트워크 ▼

\* 디스크 오퍼링 ⓘ

Custom-WB ▼



\* 크기 ⓘ

500

\* 가상머신용 컴퓨터 오퍼링 ⓘ

2Core-4GB

취소 확인

"공유 파일시스템 생성" 대화상자에서 다음의 항목을 확인합니다.

- 네트워크: 쿠버네티스 클러스터에 연결한 네트워크와 동일한 네트워크 선택
- 크기: 스토리지의 저장 공간 크기를 입력
- 가상머신용 컴퓨터 오퍼링: 스토리지 컨트롤러의 컴퓨터 크기를 선택

대화상자의 모든 항목을 입력한 후 "확인" 버튼을 클릭하여 공유 파일시스템을 생성합니다.

## Kubectl을 통한 클러스터 연결 확인

Mold에서 "컴퓨터 > 쿠버네티스"를 선택하고 목록에서 위에서 생성한 쿠버네티스 클러스터를 클릭하여 상세 화면으로 이동한 후 "액세스" 탭을 클릭합니다.

ABLESTACK K8S Cluster 001

상세

액세스

가상머신

방화벽

포트 포워딩

부하 분산

이벤트

코멘트

상태: 실행중

아이디: 443fc245-c317-4463-b002-90ac0dd5fde1

IP: 10.10.31.41

메모리: 24576 MB 메모리

템플릿: fb64f092-9beb-4103-ae5b-471f51728cf2

컴퓨터 오퍼링: 2Core-4GB

Zone: ABLESTACK-PoC

계정: admin

쿠버네티스 클러스터 구성

```
apiVersion: v1
clusters:
- cluster:
certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0LS0kck1JSURCVENDQWUyZ0F3SUJBZ0U0RmNua21kK0pkTGd3RFFZSktvWkl
```

쿠버네티스 클러스터 구성 다운로드

CLI 사용

- 클러스터 용 kubeconfig 다운로드
- kubectl 명령 줄 도구는 kubeconfig 파일을 사용하여 클러스터를 선택하고 API 서버와 통신하는 데 필요한 정보를 찾습니다.
- 클러스터의 쿠버네티스 버전 용 kubectl 도구 다운로드
- Linux: <https://storage.googleapis.com/kubernetes-release/release/v1.30.3/bin/linux/amd64/kubectl>
- MacOS: <https://storage.googleapis.com/kubernetes-release/release/v1.30.3/bin/darwin/amd64/kubectl>
- Windows: <https://storage.googleapis.com/kubernetes-release/release/v1.30.3/bin/windows/amd64/kubectl.exe>
- 클러스터에 액세스하기위한 kubectl 및 kubeconfig 파일

위와 같이 표시된 화면에서 "쿠버네티스 클러스터 구성 다운로드" 버튼을 클릭하여 구성 정보 파일을 원하는 위치에 다운로드 합니다. kube.conf 가 지정된 위치에서 Shell 파일을 열어 다음과 같이 명령을 실행하여 클러스터에 정상적으로 연결되는지 확인합니다.

Bash

```
kubectl --kubeconfig ./cluster1.conf get nodes
```

## NFS CSI 드라이버 설치

셸 화면에서 다음의 명령을 실행하여 NFS CSI 드라이버를 설치합니다.

Auto(Bash)

```
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/rbac-csi-nfs.yaml
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/csi-nfs-driverinfo.yaml
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/csi-nfs-controller.yaml
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/csi-nfs-node.yaml
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/crd-csi-snapshot.yaml
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/rbac-snapshot-controller.yaml
kubectl --kubeconfig ./cluster1.conf apply -f
https://raw.githubusercontent.com/kubernetes-csi/csi-driver-
nfs/v4.11.0/deploy/csi-snapshot-controller.yaml
```

명령이 정상적으로 실행되었다면 다음 단계를 수행합니다.

## 디폴트 스토리지 클래스 생성

Portainer나 Helm과 같은 관리 도구를 이용해 쿠버네티스 애플리케이션을 배포하기 위해서는 스토리지 클래스를 생성하고 해당 스토리지 클래스를 디폴트 스토리지로 설정해야 합니다. 디폴트 스토리지를 설정하면 애플리케이션은 기본적으로 해당 스토리지를 PV, PVC로 사용합니다.

Mold의 "스토리지 > 공유 파일 시스템"을 선택하고 위에서 생성한 파일 시스템을 선택해 상세 정보를 확인하여 액세스 포인트를 다음과 같이 확인합니다.

The screenshot shows the Portainer interface for configuring an NFS share. The left sidebar has a menu with '상세' (Details), '액세스' (Access), '네트워크' (Network), '메트릭' (Metrics), and '이벤트' (Events). The main content area is titled 'NFS를 통해 공유 파일 시스템 마운트' (Mount shared file system via NFS). Under the '액세스' tab, there is a section '쿠버네티스 서비스를 위한 격리 네트워크' (Isolated network for Kubernetes service). Below this, two commands are listed:

- `mount -t nfs 10.1.1.156:/export [local_mount_path]`  
*(Mount the NFS share. Additional mount options can be given as required.)*
- `showmount -e 10.1.1.156`  
*(Check the status of the NFS server using the showmount command.)*

액세스 포인트에서 IP 주소와 경로를 확인하여 관련 정보를 메니페스트에 적용하고 다음의 명령을 실행합니다.

Bash

```
kubectl --kubeconfig ./cluster1.conf apply -f - <<EOF
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-default-cluster
  annotations:
    storageclass.kubernetes.io/is-default-class:
"true"
provisioner: nfs.csi.k8s.io
parameters:
  server: 10.1.1.185
  share: /export
  # csi.storage.k8s.io/provisioner-secret is only
needed for providing mountOptions in DeleteVolume
  # csi.storage.k8s.io/provisioner-secret-name:
"mount-options"
  # csi.storage.k8s.io/provisioner-secret-namespace:
"default"
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
mountOptions:
  - nfsvers=4.1
EOF
```

스토리지 클래스 실행이 정상적으로 완료되면 다음의 단계를 수행합니다.

## Portainer 배포 및 연결 확인

Portainer를 Koral이 생성한 쿠버네티스 클러스터에 설치하고 사용하는 것은 매우 간단합니다. 모든 과정이 자동으로 연동되기 때문에 바로 사용자 웹 브라우저에서 연결하여 사용할 수 있습니다.

다음의 명령을 실행하여 Portainer를 설치합니다.

### Bash

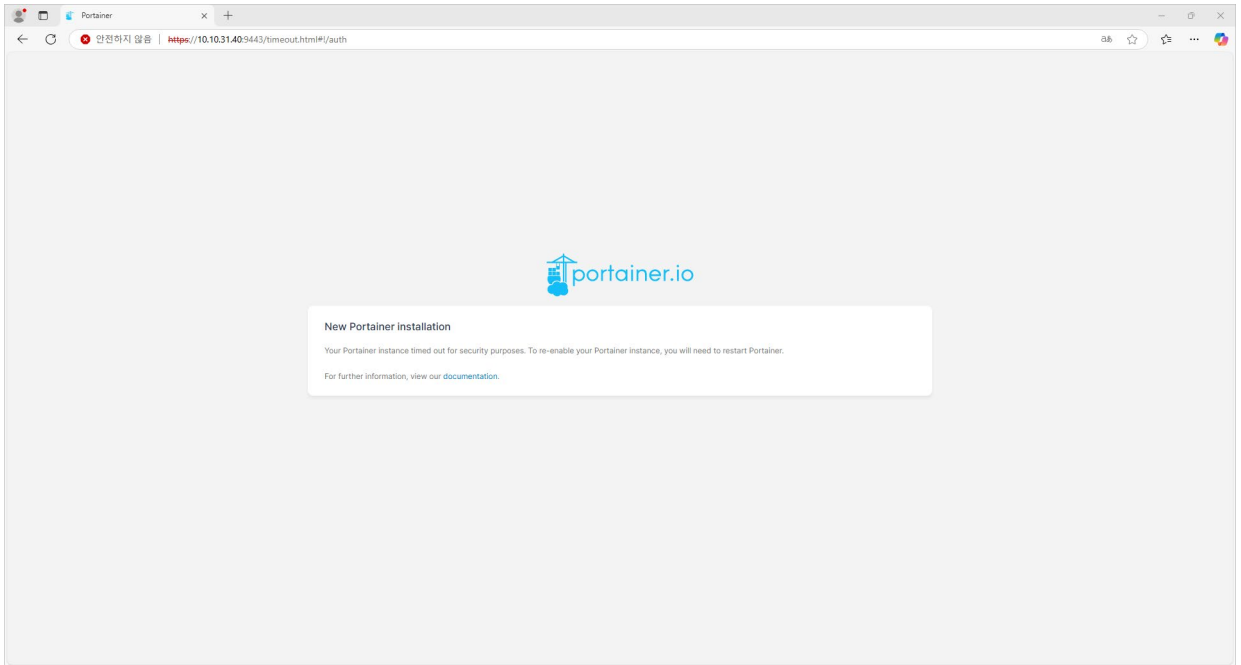
```
kubectl --kubeconfig ./cluster1.conf apply -n
portainer -f https://downloads.portainer.io/ce-
lts/portainer-lb.yaml
```

명령이 실행된 후 다음의 명령을 실행하여 외부 접속 주소(EXTERNAL-IP)와 연결되어 있는 포트를 확인합니다.

### Auto

```
kubectl --kubeconfig ./cluster1.conf get services -n
portainer
```

연결되어 있는 포트 중 9443이 https 포트입니다. 다음과 같이 브라우저에서 주소를 입력하여 Portainer 대시보드에 연결합니다.



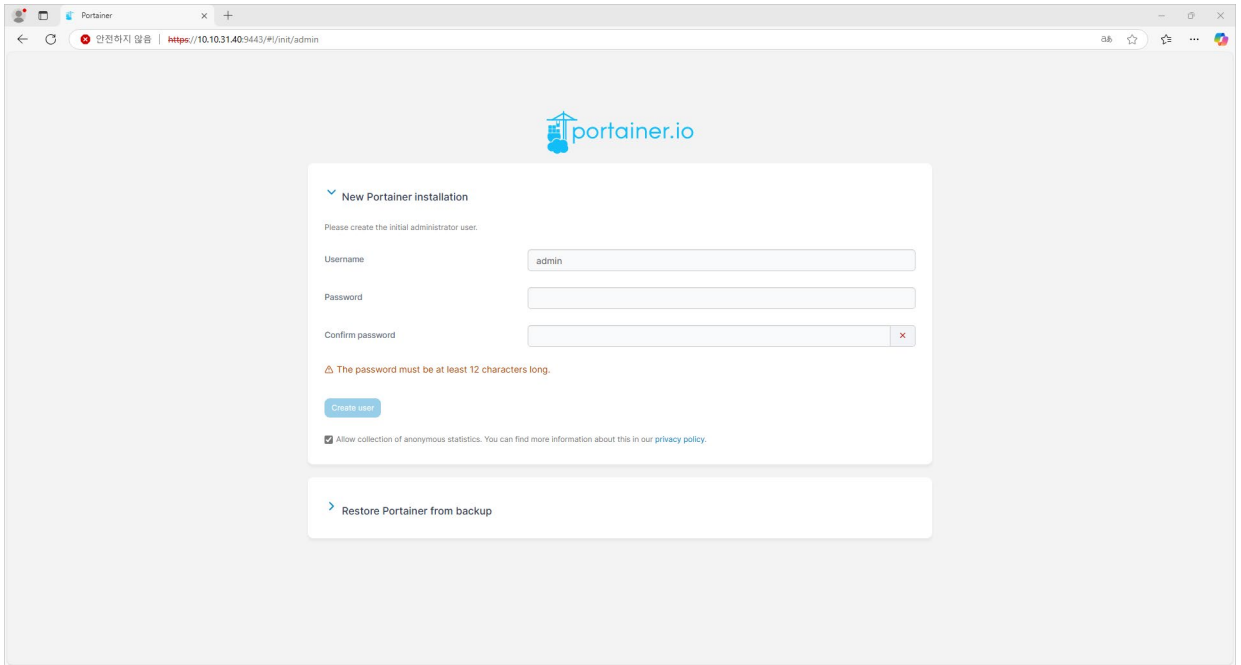
처음 접속하면 위와 같이 보안 목적으로 인스턴스를 실행할 수 없으니 재활성화를 위해 인스턴스를 재시작하라는 메시지가 표시됩니다.

다음의 명령을 실행하여 인스턴스를 재시작합니다.

#### Bash

```
kubectl --kubeconfig ./kube.conf rollout restart  
deployment portainer -n portainer
```

인스턴스가 재시작하면 다음과 같이 관리자 비밀번호를 변경하라는 화면이 표시되며 12자 이상의 비밀번호를 입력하면 Portainer 대시보드에 접속할 수 있습니다.



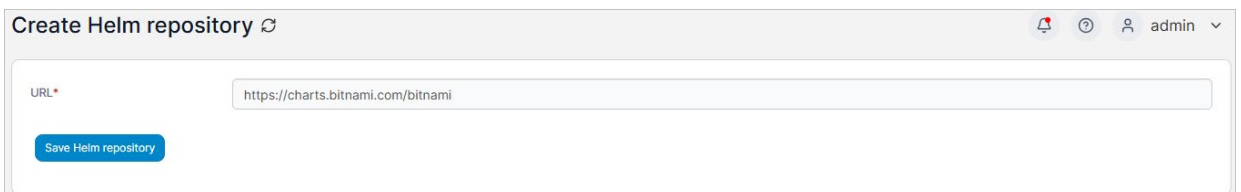
처음 대시보드에 접속하면 표시되는 화면에서 "Get Started"를 선택하면 현재 Portainer가 설치되어 있는 클러스터를 로컬 클러스터로 하여 쿠버네티스 클러스터 관리 화면이 표시됩니다.

## 워드프레스 샘플 컨테이너 배포 및 테스트

Portainer 대시보드에서 샘플 컨테이너를 배포하는 가장 간단한 방법은 Helm Chart를 사용하는 것입니다. Helm Chart는 Artifact Hub에서 필요한 애플리케이션을 검색할 수 있습니다.

가장 많은 오픈소스 자료를 보유하고 있는 Helm Chart 레지스트리는 Bitnami 레지스트리입니다. 해당 레지스트리를 Portainer에 등록하면 언제든지 컨테이너를 배포할 수 있습니다.

Portainer 대시보드에서 오른쪽 상단의 "admin"을 선택하여 My Account를 선택합니다. 표시된 User settings 화면에서 맨 아래의 Helm repositories 항목에서 "Add Helm repository" 버튼을 클릭합니다. 표시된 Create Helm repository 화면에서 다음과 같이 주소를 입력합니다.



"Save Helm repository" 버튼을 클릭하여 저장합니다. 이제 애플리케이션을 자동으로 배포할 준비가 되었습니다.

워드프레스를 배포해 보겠습니다. 관리의 편의를 위해 먼저 네임스페이스를 생성합니다. 네임스페이스 생성을 위해 클러스터 메뉴에서 "Namespaces" 메뉴를 선택합니다. 표시된 "Namespace list" 화면에서 "Add with form" 버튼을 클릭하여 다음과 같이 입력합니다.

The screenshot shows the 'Create a namespace' form with the following details:

- Name:** wordpress
- Annotations:** Add annotation button
- Resource Quota:** A resource quota sets boundaries on the compute resources a namespace can use. It's good practice to set a quota for a namespace to manage resources effectively. Alternatively, you can disable assigning a quota for unrestricted access (not recommended). Resource assignment is turned on.
- Registries:** Define which registries can be used by users who have access to this namespace. No registries available. Head over to the registry view to define a container registry.
- Storage:** Quotas can be set on each storage option to prevent users from exceeding a specific threshold when deploying applications. You can set a quota to 0 to effectively prevent the usage of a specific storage option inside this namespace. Enable quota is turned on.
- Summary:** Portainer will execute the following Kubernetes actions.
  - Create a **Namespace** named `wordpress`
- Actions:** Create namespace button

"Create namespace" 버튼을 클릭하여 네임스페이스를 만들고 다시 클러스터 메뉴에서 "Applications"를 선택하면 표시되는 "Application list" 화면에서 "Create from code"를 선택합니다. 표시된 "Create from code" 화면에서 Deploy from 항목의 "Helm chart"를 선택합니다.

Deploy to 항목에서 Namespace를 wordpress로 선택하고 Helm chart에서 wordpress를 검색하면 다음과 같습니다.



The screenshot shows the 'Create from code' interface. At the top, there is a blue 'Deploy' button and a 'Logs' link. Below this, the 'Deploy from' section has five options: Repository, Web editor, URL, Custom template, and Helm chart. The 'Helm chart' option is selected with a blue checkmark. The 'Deploy to' section has a 'Namespace' dropdown set to 'wordpress' and a 'Name\*' input field containing 'e.g. my-app'. A red warning triangle indicates 'This field is required.' Below this, the 'Helm chart' section has a search input with 'wordpress' and a 'Select a category' dropdown. A disclaimer box states: 'At present Portainer does not support OCI format Helm charts. Support for OCI charts will be available in a future release. If you would like to provide feedback on OCI support or get access to early releases to test this functionality, please get in touch.' The search results show two items: 'wordpress' (Helm) and 'wordpress-intel' (Helm). The 'wordpress' item is highlighted.

위 화면에서 첫번째 wordpress 항목을 선택한 후 name 항목에 wordpress를 입력하고 Install 버튼을 클릭하면 wordpress가 인스톨 되고 시작됩니다.

수초~분 내에 모든 배포가 완료됩니다. 다음과 같이 Application list 화면에서 wordpress 애플리케이션이 정상적으로 배포된 것을 확인할 수 있습니다.

### Application list

Applications    Stacks

Search...    Remove    + Add with form    + Create from code

System resources are hidden, this can be changed in the table settings

Helm option: From 2.20 and on, the Helm menu sidebar option has moved to the **Create from code screen** - accessed via the button above.

Name	Stack	Namespace	Image	Application Type	Status	Published	Created
cadvisor External	-	cadvisor	gcr.io/cadvisor/cadvisor:v0.45.0	DaemonSet	Global 3 / 3	No	2025-03-22
dashboard-metrics-scraper External	-	kubernetes-dashboard	kubernetesui/metrics-scraper:v1.0.8	Deployment	Replicated 1 / 1	Yes	2025-03-22
kubernetes-dashboard External	-	kubernetes-dashboard	kubernetesui/dashboard:v2.7.0	Deployment	Replicated 1 / 1	Yes	2025-03-22
prometheus External	-	prometheus	prom/prometheus:v3.2.1	Deployment	Replicated 1 / 1	Yes	2025-03-22
wordpress	-	wordpress	docker.io/bitnami/wordpress:6.7.2-debian-12-r8	Helm	Ready	Yes	2025-03-23

Name	Stack	Namespace	Image	Application Type	Status	Published	Created
wordpress	-	wordpress	docker.io/bitnami/wordpress:6.7.2-debian-12-r8	Deployment	Replicated 1 / 1	Yes	2025-03-23 02:10:41
wordpress-mariadb	-	wordpress	docker.io/bitnami/mariadb:11.4.5-debian-12-r4	StatefulSet	Replicated 1 / 1	Yes	2025-03-23 02:10:41

Items per page: 10

배포된 애플리케이션에 접속하기 위해서는 노출된 외부 IP와 포트를 알아야 합니다. 클러스터 메뉴에서 "Networking > Services"를 선택하면 표시되는 "Service list" 화면에서 wordpress 항목에 표시된 External IP를 확인합니다. 다음의 화면과 같습니다.

### Service list

Services

Search...    Remove    + Create from code

System resources are hidden, this can be changed in the table settings

Name	Application	Namespace	Type	Ports	Target Ports	Cluster IP	External IP	Created
dashboard-metrics-scraper External	dashboard-metrics-scraper	kubernetes-dashboard	ClusterIP	8000/TCP	8000	10.97.87.93	-	2025-03-22 23:51
kubernetes External	-	default	ClusterIP	443/TCP	6443	10.96.0.1	-	2025-03-22 23:51
kubernetes-dashboard External	kubernetes-dashboard	kubernetes-dashboard	ClusterIP	443/TCP	8443	10.108.74.22	-	2025-03-22 23:51
prometheus-nodeport External	prometheus	prometheus	NodePort	9090:30090/TCP	9090	10.111.93.86	-	2025-03-22 23:51
wordpress	wordpress	wordpress	LoadBalancer	80:32051/TCP 443:31043/TCP	http https	10.97.8.73	<a href="#">10.10.31.43</a>	2025-03-23 02:10
wordpress-mariadb	wordpress-mariadb	wordpress	ClusterIP	3306/TCP	mysql	10.105.16.104	-	2025-03-23 02:10
wordpress-mariadb-headless	wordpress-mariadb	wordpress	ClusterIP	3306/TCP	mysql	None	-	2025-03-23 02:10

Items per page: 10

표시된 "ExternalIP"를 클릭하면 바로 설치된 wordpress의 샘플 화면이 웹 브라우저에 표시됩니다. 관리자 페이지는 /admin으로 접속합니다.

관리자 아이디와 비밀번호는 애플리케이션 속성에서 확인할 수 있습니다. Applications 메뉴를 선택한 후 wordpress 항목에서 wordpress 파드를 선택하면 "Application details" 화면이 표시되며 하단의 "Environment variables, ConfigMaps or Secrets" 항목을 확인하면 WORDPRESS\_USERNAME과, WORDPRESS\_PASSWORD 항목을 확인할 수 있으며 해당 아이디와 비밀번호로 관리자 로그인 가능합니다.

더 자세한 사용법은 Portainer의 공식문서와 Helm의 공식문서를 참조하십시오.

## 멀티 클러스터 관리

Portainer는 ABLESTACK Koral과 연동해서 멀티 클러스터를 관리할 수 있습니다. 여러 개의 클러스터를 단일 대시보드에서 관리할 수 있다는 것은 매우 편리한 기능입니다.

쿠버네티스 클러스터를 앞에서 설명한 방식으로 동일하게 만든 후 이미 만들어진 Portainer에 클러스터로 등록하여 통합 관리하는 방법은 다음과 같습니다.

1. 쿠버네티스를 위한 네트워크와 클러스터를 생성합니다.
2. 쿠버네티스 클러스터에서 사용할 NFS 파일 시스템을 생성합니다.
3. NFS CSI 드라이버를 설치합니다.
4. 쿠버네티스 클러스터에 디폴트 스토리지 클래스를 등록합니다.
5. 쿠버네티스 클러스터에 Portainer 에이전트를 설치합니다.
6. 에이전트 Endpoint IP와 포트를 이용해 클러스터를 등록합니다.

1~4번까지의 과정은 앞서 설명한 과정이므로 동일하게 실행합니다. 여기서는 5번부터 설명합니다.

모든 설정을 완료한 새로운 쿠버네티스 클러스터에 Portainer 에이전트를 설치하기 위해 다음의 명령을 실행합니다. kube.conf 파일은 클러스터에 따라 다르므로 경우에 따라 다른 파일 이름을 사용해야 합니다.

## Auto(Bash)


```
kubectl --kubeconfig cluster2.conf apply -f
https://downloads.portainer.io/ce2-27/portainer-
agent-k8s-lb.yaml
```

에이전트를 설치한 후 Mold의 해당 클러스터를 위한 네트워크를 확인(네트워크 > 가상머신용 네트워크)하여 Public 주소의 부하분산 규칙을 확인 하면 다음과 같습니다.

The screenshot shows the Portainer interface for configuring a load balancer. The left sidebar shows the IP address 10.10.31.37 and the zone ABLESTACK-PoC. The main area is titled '부하 분산' (Load Balancing) and contains a table of load balancers. The table has columns for Name, Public Port, Actual Port, Algorithm, Source CIDR, Protocol, Health Check, and Status. One load balancer is listed with ID 'a018cc39d5e88482e81dd34e37cf5f20-tcp-9001', Public Port 9001, Actual Port 31982, Round-robin algorithm, TCP protocol, and Active status. There is a '+ 추가' (Add) button next to the table.


에이전트가 설치된 쿠버네티스 클러스터를 Portainer에 등록하기 위해 Portainer의 Administration > Environment-related > Environments 메뉴를 선택합니다. 표시된 "Environments" 화면에서 "Add environment" 버튼을 클릭합니다.

표시된 Quick Setup 화면에서 "Kubernetes"를 선택하고 Start Wizard 버튼을 클릭합니다. Environment Wizard 화면에서 Agent를 선택한 후 "Name"을 입력한 후 "Environment address"에 위에서 확인한 주소와 포트를 다음의 그림과 같이 입력합니다.


 Environment Wizard

**1**  
Kubernetes


Connect to your Kubernetes environment



**Agent**



**Edge Agent Standard**




**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/ce2-27/portainer-agent-k8s-lb.yaml
```

 Copy command

Name\*

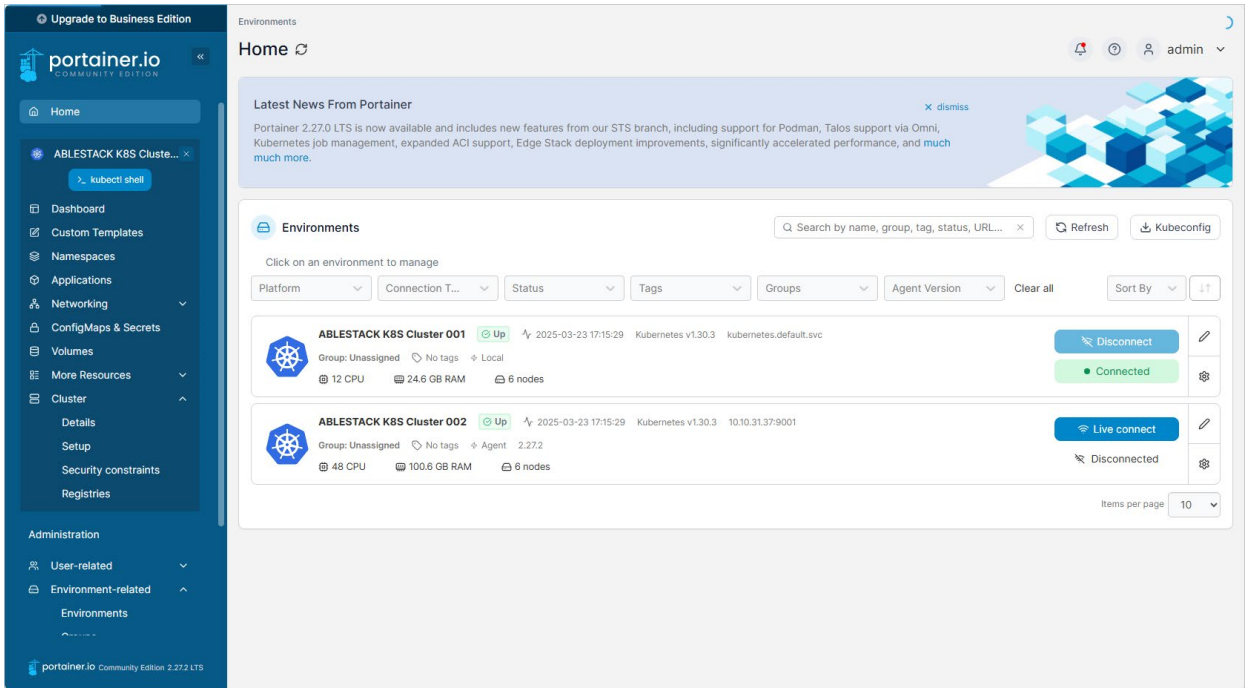
Environment address \* 

▼ More settings

 Connect

Connect 버튼을 클릭하여 클러스터를 추가합니다. 이제 단일 대시보드에서 여러 개의 쿠버네티스 클러스터를 관리할 수 있게 되었습니다.

Portainer에서 Home 메뉴를 클릭하면 등록된 클러스터 목록이 표시됩니다. 작업하고자 하는 클러스터를 선택해서 컨테이너를 배포하거나 관리할 수 있습니다.



ABLECLOUD

All about data & cloud

주식회사 에이블클라우드 [www.ablestack.co.kr](http://www.ablestack.co.kr)

주소 | 서울시 영등포구 영신로 220, KnK디지털타워 1901호

연구소 | 대전시 대덕구 대화로 106번길 66, 펜타플렉스 810~812호

대표전화 | 02-456-7667

이메일 | [sales@ablestack.co.kr](mailto:sales@ablestack.co.kr)



공식홈페이지