

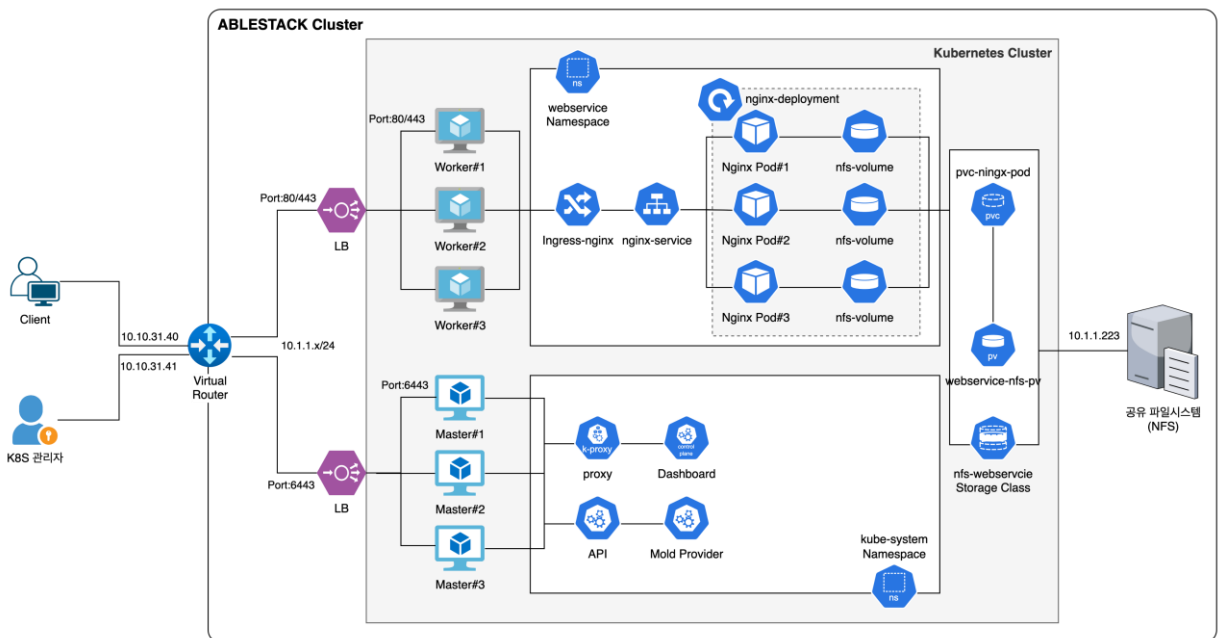
ABLESTACK

ABLESTACK Kubernetes Cluster 사용기초

ABLESTACK- Kubernetes Cluster 사용기초

ABLESTACK은 Koral이라는 내장된 쿠버네티스 클러스터 관리 기능을 제공합니다. 사용자는 원클릭으로 쿠버네티스 클러스터를 자동으로 배포할 수 있고 다양한 관리 도구를 이용해서 애플리케이션을 배포하고 서비스할 수 있습니다.

본 문서는 Koral을 이용해 쿠버네티스 클러스터를 배포하고, Nginx 웹 서버를 배포하여 웹 서비스를 구축하는 기초적인 구축 과정을 Mold 통합 관리 도구와 쿠버네티스 기본 관리도구인 Kubectl을 이용해 수행하는 과정을 설명합니다. 배포 아키텍처는 다음과 같습니다.



위와 같이 복잡한 아키텍처를 Mold를 통해 빠르게 구성하고 웹 서비스를 배포할 수 있습니다. 위의 아키텍처를 배포하는 내용을 쉽게 이름 붙여 보면 "로드밸런서로 부하 분산되는 Nginx 웹 서비스 구축하기" 입니다.

로드밸런서로 부하 분산되는 Nginx 웹 서비스 구축하기

클러스터 배포를 위해서 관리자는 자신의 컴퓨터에 Mold에 접속할 수 있는 웹 브라우저, 그리고 kubectl을 준비해야 하며, Mold 및 생성된 Kubernetes 클러스터의 퍼블릭 IP(예에서는 10.10.31.41)에 연결할 수 있어야 합니다. Nginx 기반의 웹 서비스를 구축하기 위해서는 다음과 같은 절차를 수행해야 합니다. 각각의 절차는 Mold를 통해서 또는 Kubernetes CLI(Kubectl)를 통해서 수행할 수 있습니다.

1. Kubernetes용 네트워크 생성
2. Kubernetes Cluster 생성
3. 공유볼륨 제공을 위한 공유 파일시스템 생성
4. Kubernetes Cluster에 연결 확인
5. Kubernetes Cluster에 NFS 볼륨(PV) 생성
6. NGINX에서 사용하기 위한 볼륨 청구(PVC) 생성
7. NGINX 배포
8. ingress-nginx를 이용해서 웹 서비스 로드밸런싱
9. 웹 서버 루트 디렉토리에 웹 페이지 배포
10. 브라우저로 웹 서비스에 접근

Kubernetes용 네트워크 생성

먼저 쿠버네티스에서 사용하기 위한 네트워크를 생성합니다.

네트워크 생성을 위해서는 적합한 네트워크 오퍼링이 있어야 합니다.

"서비스 오퍼링 > 네트워크 오퍼링" 메뉴를 선택해서 목록을 확인합니다.

ABLESTACK은 기본적으로 쿠버네티스에서 사용할 수 있는

"쿠버네티스 서비스에 대한 기본 네트워크오퍼링"을 제공하며 목록에서 확인할 수 있습니다.

만약 새로 만들고 싶다면 다음과 같이 진행합니다.

1. 네트워크 오퍼링 추가 버튼 클릭
2. 네트워크 오퍼링 추가 대화상자에서 다음과 같이 정보를 선택해서 "확인" 버튼 클릭

네트워크 오퍼링 추가 ?
✕

*** 이름** ?

설명 ?

네트워크 속도(Mb/s) ?

게스트 유형 ?

Isolated

L2

shared

인터넷 프로토콜 ?

[IPv6 support in ABLESTACK - Isolated networks and VPC tiers](#)을 제공하는 IPv6 지원 네트워크/VPC 생성에 대한 설명서를 참조하십시오.

IPv4 IPv4 + IPv6 (Dual Stack)

VLAN 지정 ?

VPC ?

영구적 ?

NSX ?

네트워크 모드 ?

Indicates the mode with which the network will operate. Valid option: NATTED or RO... ▼

<p>비규칙 모드 ⓘ</p> <p><input checked="" type="radio"/> 없음 <input type="radio"/> 승인 <input type="radio"/> 거부</p>	<p>Mac 변조 전송 ⓘ</p> <p><input checked="" type="radio"/> 없음 <input type="radio"/> 승인 <input type="radio"/> 거부</p>
<p>MAC 주소 변경 ⓘ</p> <p><input checked="" type="radio"/> 없음 <input type="radio"/> 승인 <input type="radio"/> 거부</p>	<p>MAC Learning ⓘ</p> <p><input checked="" type="radio"/> 없음 <input type="radio"/> 승인 <input type="radio"/> 거부</p>
<p>지원되는 서비스 ⓘ</p> <p><input checked="" type="checkbox"/> PortForwarding <input type="checkbox"/> SecurityGroup</p> <p><input type="checkbox"/> NetworkACL</p> <p><input type="checkbox"/> Connectivity</p> <p><input type="checkbox"/> BaremetalPxeService</p>	
<p>* 컴퓨터 오퍼링 ⓘ</p> <p><input type="text" value="System Offering For Software Router"/></p>	
<p>Redundant 라우터</p> <p><input type="checkbox"/></p>	
<p>지원되는 Source NAT 유형</p> <p><input checked="" type="radio"/> Per 계정 <input type="radio"/> Per Zone</p>	
<p>VM 오토스케일링 지원</p> <p><input checked="" type="checkbox"/></p>	
<p>절약 모드 ⓘ</p> <p><input checked="" type="checkbox"/></p>	

태그 ⓘ

the tags for the network offering.

기본 송신 정책 ⓘ

허용
거부

공개

Zone ⓘ

the ID of the containing zone(s), null for public offerings

네트워크 오퍼링 활성화 ⓘ

취소
확인

위의 대화상자에서 확인해야 할 중요한 설정 내용은 다음과 같습니다.

- VLAN 지정을 하지 않음(단, VLAN을 수동으로 지정해야 하는 환경에서는 해당 항목을 활성화)
- 지원되는 서비스 : VPN, Dhcp, Dns, Firewall, Lb, UserData, SourceNat, StaticNat, PortForwarding (모두 제공자는 Virtual Router 선택)
- 기본 송신 정책 : 허용
- 네트워크 오퍼링 활성화 : 선택

확인 버튼을 클릭하여 네트워크 오퍼링을 생성한 후 쿠버네티스를 위한 네트워크를 다음과 같이 생성합니다.

- 1.네트워크 > 가상머신용 네트워크 메뉴를 선택하고 "네트워크 추가" 버튼 클릭
- 2.네트워크 추가 대화상자에서 다음과 같이 항목을 입력하고 "확인" 버튼을 클릭하여 생성

네트워크 추가 ?
✕

Isolated
L2
shared

*** 이름** ?

설명 ?

*** Zone** ?

소유자 유형

*** 도메인**

*** 계정**

네트워크 도메인 ?

*** 네트워크 오퍼링** ?

VLAN/VNI ?

외부 아이디 ?

게이트웨이 ⓘ

the gateway of the network. Required for shared networks and isolated networks when i...

넷마스크 ⓘ

the netmask of the network. Required for shared networks and isolated networks when ...

DNS 1 ⓘ **DNS 2** ⓘ

the first IPv4 DNS for the network the second IPv4 DNS for the network

서비스용 네트워크의 라우터에 대한 IPv4 주소 ⓘ

IPv4 address to be assigned to the public interface of the network router. This address ...

위 대화 상자에서 확인해야 할 주요 설정은 다음과 같습니다.

•네트워크 오퍼링 : Kubernetes 클러스터에서 사용하기 위해 위에서 만든 네트워크 오퍼링을 선택

확인 버튼을 클릭하여 네트워크를 생성하고 다음 단계를 수행합니다.

Kubernetes Cluster 생성

Mold에서 Kubernetes Cluster를 다음과 같이 생성합니다.

1. 컴퓨터 > 쿠버네티스 메뉴를 선택하고 "쿠버네티스 클러스터 생성" 버튼을 클릭
2. "쿠버네티스 클러스터 생성" 대화상자에 다음과 같이 입력하고 "확인" 버튼을 클릭

쿠버네티스 클러스터 생성 ? ×

*** 이름** ⓘ

설명 ⓘ

*** Zone** ⓘ

*** 쿠버네티스 버전** ⓘ

*** 컴퓨트 오퍼링** ⓘ

노드 루트 디스크 크기(GB) ⓘ

네트워크 ⓘ

HA 활성화 됨

*** 마스터 노드** ⓘ

외부 로드 밸런서 IP 주소 ⓘ

*** 클러스터 크기** ⓘ

3

SSH 키 쌍 ⓘ

▼

사설 레지스트리

위의 대화상자에서 확인해야 할 중요한 설정 내용은 다음과 같습니다.

- 컴퓨트 오퍼링 : Kubernetes를 구성하는 노드(가상머신)의 컴퓨트 사이징을 선택, 모든 노드에 동일하게 적용됨
- 노드 루트 디스크 크기 : 가상머신의 루트 디스크는 컨테이너의 오버레이 디스크로 사용되므로 넉넉하게 할당
- 네트워크 : Kubernetes 클러스터를 위해 위에서 만든 네트워크를 선택
- HA 활성화 됨 : 마스터 노드를 무중단으로 운영하고자 할 때 선택
- 마스터 노드 : HA가 활성화 되면 3개가 기본, 활성화 되지 않으면 1개가 기본(2대로 HA 구성하는 것은 가능하나 권장하지 않음)
- 클러스터 크기 : Kubernetes 클러스터의 Worker 노드의 수를 설정

확인 버튼을 클릭하여 클러스터를 생성합니다. 마스터 노드 및 워커 노드, 그리고 각종 설정 적용으로 인해 클러스터 생성이 완료되는 데는 수분이 소요 됩니다.

그 사이 다음의 NFS 스토리지를 생성합니다.

공유볼륨 제공을 위한 공유 파일시스템 생성

Mold를 통해 Kubernetes 클러스터에서 사용할 수 있는 공유 볼륨을 위해 공유 파일시스템을 생성합니다.

1. 스토리지 > 공유 파일시스템 메뉴를 선택하고 "공유 파일시스템 생성" 버튼을 클릭
2. 공유 파일시스템 생성 대화상자에서 다음과 같이 입력하고 "확인" 버튼을 클릭

공유 파일시스템 생성 ?
✕

소유자 유형

계정 ▼

*** 도메인**

도 ROOT ▼

*** 계정**

admin ▼

*** 이름**

K8S-WebService-NFS

설명

K8S-WebService-NFS

*** Zone i**

ABLESTACK-PoC ▼

파일 시스템 i

XFS ▼

* 네트워크 ⓘ

Kubernetes-10Gb-Network(Isolated) ▼

* 디스크 오퍼링 ⓘ

Custom-WB ▼

* 크기 ⓘ

500

* 가상머신용 컴퓨터 오퍼링 ⓘ

2Core-4GB ▼

취소 확인

위의 대화상자에서 확인해야 할 중요한 설정 내용은 다음과 같습니다.

- 네트워크 : Kubernetes 클러스터가 사용하는 네트워크와 동일한 네트워크 선택
- 디스크 오퍼링 및 크기 : 스토리지 용량을 설정
- 가상머신용 컴퓨터 오퍼링 : NFS 서버의 컴퓨터 용량 선택

확인 버튼을 클릭하여 NFS 스토리지를 생성합니다.

Kubernetes Cluster 연결 확인

Kubernetes 클러스터 생성이 완료 되는 시간 동안 관리자는 사용자 컴퓨터에 Kubernetes CLI 도구(kubectl)를 설치하고 다음의 순서로 Kubernetes 클러스터에 연결할 수 있도록 준비합니다. kubectl 설치하는 컴퓨터 운영체제에 따라 다르므로 Kubernetes의 공식 문서를 참고합니다.

Kubernetes 클러스터가 생성되고 클러스터 상태가 "실행중"으로 표시되면 클러스터가 모두 준비된 것입니다. 다음의 순서로 연결을 확인합니다.

1. 컴퓨터 > 쿠버네티스 메뉴를 선택하고 표시된 목록에서 생성한 쿠버네티스 클러스터를 선택하여 상세 화면으로 이동
2. 상세 화면에서 "액세스" 탭을 선택하여 "쿠버네티스 클러스터 구성 다운로드" 버튼을 클릭하여 kube.conf 파일 다운로드하고 원하는 위치로 선택해 복사
3. kube.conf 파일이 있는 디렉토리로 이동한 후 kubectl을 이용해서 다음의 명령을 실행해서 정상적으로 실행 되는지 확인

Bash

#Pod 목록

```
kubectl --kubeconfig ./kube.conf get pods --all-namespaces
```

#Node 목록

```
kubectl --kubeconfig ./kube.conf get nodes --all-namespaces
```

#서비스 목록

```
kubectl --kubeconfig ./kube.conf get services --all-namespaces
```

4. 로컬에서 프록시를 실행하기 위해 다음의 명령을 실행

Bash

```
kubectl --kubeconfig ./kube.conf proxy
```

5. 사용자 컴퓨터의 브라우저에서 다음의 URL 조회

Auto (Bash)

```
http://localhost:8001/api/v1/namespaces/kubernetes-  
dashboard/services/https:kubernetes-dashboard:/proxy/
```

6. 대시보드에 로그인할 수 있도록 다음의 명령을 이용해서 토큰을 생성

Bash

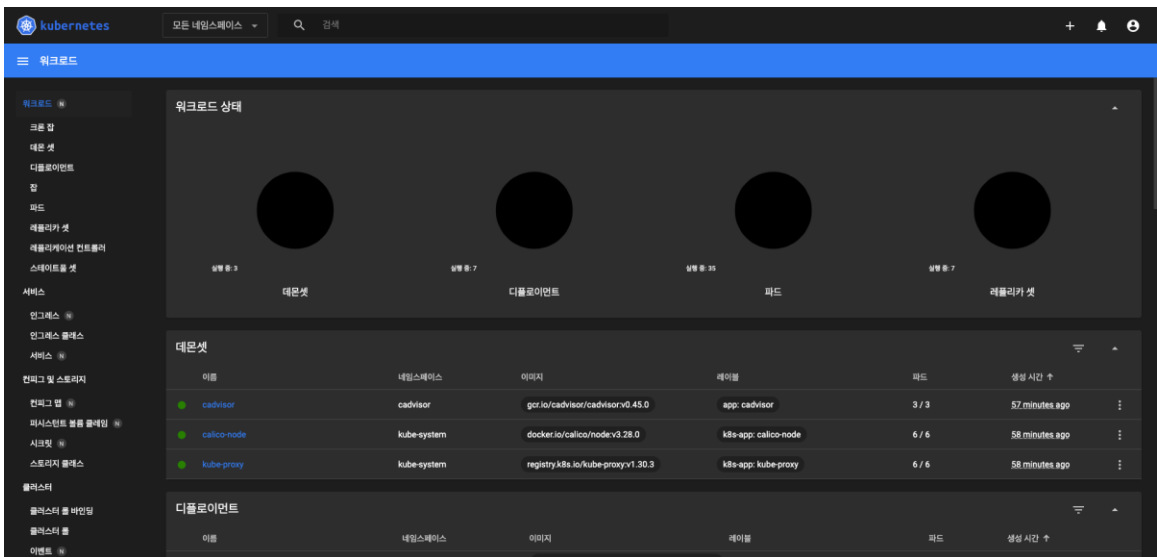
```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF  
apiVersion: v1  
kind: ServiceAccount  
metadata:  
  name: kubernetes-dashboard-admin-user  
  namespace: kubernetes-dashboard  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: ClusterRoleBinding  
metadata:  
  name: kubernetes-dashboard-admin-user  
roleRef:  
  apiGroup: rbac.authorization.k8s.io  
  kind: ClusterRole  
  name: cluster-admin  
subjects:  
- kind: ServiceAccount  
  name: kubernetes-dashboard-admin-user  
  namespace: kubernetes-dashboard  
---  
apiVersion: v1  
kind: Secret  
type: kubernetes.io/service-account-token  
metadata:  
  name: kubernetes-dashboard-token  
  namespace: kubernetes-dashboard  
  annotations:  
    kubernetes.io/service-account.name: kubernetes-  
dashboard-admin-user  
EOF
```

7. 토큰을 다음의 명령을 이용해 확인

Bash

```
kubectl --kubeconfig ./kube.conf describe secret $(kubectl --
kubeconfig ./kube.conf get secrets -n kubernetes-dashboard |
grep kubernetes-dashboard-token | awk '{print $1}') -n
kubernetes-dashboard
```

8. 조회된 토큰을 복사하여 웹 대시보드 토큰 입력에 입력하고 로그인



연결이 확인되면 웹 서비스를 위한 네임스페이스를 생성합니다.

Bash

```
kubectl --kubeconfig ./kube.conf create namespace webservice
```

Kubernetes Cluster에 NFS 볼륨(PV) 생성

나중에 생성할 Kubernetes의 Nginx Pod에 할당할 수 있도록 NFS 방식의 스토리지를 연결합니다. 다음의 명령을 실행하여 스토리지 클래스를 생성합니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: nfs-webservice
  namespace: webservice
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: Immediate
EOF
```

PV를 생성하기 위해서는 NFS의 액세스 경로를 알아야 합니다. Mold의 "스토리지 > 공유 파일 시스템" 메뉴에서 위에서 생성한 공유 파일 시스템을 선택하고 상세화면에서 "액세스" 탭의 내용을 확인하여 생성한 NFS 파일시스템 서비스의 접속 경로를 확인합니다.

The screenshot shows a configuration window for mounting an NFS share. The title is "NFS를 통해 공유 파일 시스템 마운트". On the left, there is a sidebar with tabs: "상세", "액세스", "네트워크", "메트릭", and "이벤트". The "액세스" tab is active. The main content area shows the following configuration:

- Network: **Kubernetes-10Gb-Network(Isolated)**
- Mount Command: `mount -t nfs 10.1.1.57:/export [local_mount_path]`
(Mount the NFS share. Additional mount options can be given as required.)
- Showmount Command: `showmount -e 10.1.1.57`
(Check the status of the NFS server using the showmount command.)

위의 그림의 예를 보면 NFS 서버의 IP 주소는 10.1.1.57이고 공유 경로는 /export 입니다. 다음의 명령을 실행하여 NFS 스토리지를 쿠버네티스 클러스터에 연결합니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
apiVersion: v1
kind: PersistentVolume
metadata:
  name: webservice-nfs-pv
  namespace: webservice
spec:
  capacity:
    storage: 500Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  storageClassName: nfs-webservice
  nfs:
    server: 10.1.1.57 #사용자 환경에 맞게 IP 변경
    path: /export
EOF
```

NGINX에서 사용하기 위한 볼륨 청구(PVC) 생성

Nginx Pod에서 사용할 수 있도록 파일 시스템에 PVC를 생성합니다. 다음의 명령을 실행합니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-nginx-pod
  namespace: webservice
spec:
  storageClassName: nfs-webservice
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 50Gi
EOF
```

NGINX 배포

다음의 명령을 이용해 NFS PVC를 사용하는 Nginx를 배포합니다. 다음의 구성 명령은 "Nginx 배포는 Deployment를 사용하며 replica 3 즉, Nginx Pod를 3개 배포"한다는 내용입니다. 따라서 아래의 명령은 사용자의 환경에 따라 수정해야 합니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: webservice
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
        volumeMounts:
        - name: nfs-volume
          mountPath: /usr/share/nginx/html
      volumes:
      - name: nfs-volume
        persistentVolumeClaim:
          claimName: pvc-nginx-pod
EOF
```

서비스를 배포하여 Pod에 접근할 수 있는 경로를 만듭니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: webservice
  labels:
    app: nginx
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
EOF
```

ingress-nginx를 이용해서 웹 서비스 로드밸런싱

ingress-nginx 컨트롤러를 다음의 명령을 이용해 설치합니다. 아래의 명령은 ingress-nginx의 변경에 따라 주소가 달라질 수 있습니다. 최신의 명령은 쿠버네티스 ingress-nginx 설치 가이드(<https://kubernetes.github.io/ingress-nginx/deploy/#bare-metal-clusters>)에서 확인해야 합니다.

Auto(Bash)

```
kubectl --kubeconfig ./kube.conf apply -f
https://raw.githubusercontent.com/kubernetes/ingress-
nginx/controller-
v1.12.0/deploy/static/provider/baremetal/deploy.yaml
```

Mold의 로드밸런서와 Kubernetes 클러스터 간의 Ingress 연동을 위해 ingress-nginx 서비스 및 컨피그 맵을 다음의 명령을 실행하여 패치합니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
---
kind: Service
apiVersion: v1
metadata:
  name: ingress-nginx
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
  annotations:
    service.beta.kubernetes.io/cloudstack-load-balancer-
proxy-protocol: enabled
spec:
  type: LoadBalancer
  selector:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
  ports:
    - name: http
      port: 80
      targetPort: http
    - name: https
      port: 443
      targetPort: https
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
data:
  use-proxy-protocol: "true"
EOF
```

Ingress를 생성하고 서비스를 매핑합니다.

Bash

```
kubectl --kubeconfig ./kube.conf apply -f - <<EOF
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-nginx
  namespace: webservice
spec:
  ingressClassName: nginx
  rules:
  - http:
    paths:
    - path: /
      pathType: Prefix
      backend:
        service:
          name: nginx-service
          port:
            number: 80
EOF
```

웹 루트 디렉토리에 웹 페이지 배포

nginx 서비스가 정상적으로 실행되고 있는지 확인하기 위해 nginx 루트 디렉토리(여에서는 /usr/share/nginx/html)에 index.html을 배포합니다.

파일을 기록하는 가장 쉬운 방법은 kubectl cp 명령을 사용하는 것입니다. (자세한 cp 명령 사용법은 Kubernetes 공식 문서를 참고)

먼저 서비스 구성을 위해 배포된 파드(컨테이너)를 확인하기 위해 다음의 명령을 실행합니다.

Bash

```
kubectl --kubeconfig ./kube.conf --namespace webservice get pods
```

파드 모두에 NFS 디렉토리가 매핑 되어 있기 때문에 파드 중 처음 파드의 이름을 확인하여 다음의 명령을 실행합니다.

Bash

```
kubectl --kubeconfig ./kube.conf cp ./index.html webservice/nginx-deployment-7f998d9995-7swvd:/usr/share/nginx/html/
```


테스트를 위한 index.html 파일은 간단하게 텍스트를 작성해서 만들면 됩니다.

브라우저로 웹 서비스에 접근

배포된 Kubernetes Cluster의 webservice를 접속하기 위한 IP 주소는 ABLESTACK Mold의 Kubernetes 네트워크 상세 정보에서 확인할 수 있습니다.





"네트워크 > 가상머신용 네트워크" 메뉴를 선택하여 Kubertenes Cluster에 연결된(예제에서는 Kubernetes-10Gb-Network(Isolated)) 네트워크를 선택하여 상세 정보를 확인합니다.

상세정보에서 "Public IP 주소" 탭을 선택하면 다음과 같습니다.

Public IP 주소	가상 라우터	VNF 기기	네트워크 권한	이벤트	코멘트
+ 새 IP 주소 가져오기					
IP 주소	상태	VM	네트워크		
<input type="checkbox"/> 10.10.31.41 <small>Source NAT</small>	● Allocated		Kubernetes-10Gb-Network(Isolated)		
<input type="checkbox"/> 10.10.31.43 <small>FW 규칙 정의</small>	● Allocated		Kubernetes-10Gb-Network(Isolated)		

전체 2 개 항목 < 1 > 10 / 쪽

위의 목록 중에 Source NAT으로 정의된 IP는 Kubernetes 클러스터에 접근하기 위한 IP이고, 웹 서비스 IP는 "FW 규칙 정의"로 표기되어 있는 IP입니다. 해당 주소(예에서는 10.10.31.43)를 클릭하면 상세 정보를 볼 수 있으며 ingress-nginx와 연동해서 방화벽과 부하분산 규칙이 다음과 같이 설정되어 있음을 확인할 수 있습니다.

방화벽	포트 포워딩	부하 분산	이벤트	코멘트	
Source CIDR	프로토콜	시작 포트	종료 포트	추가	
<input type="text"/>	TCP	<input type="text"/>	<input type="text"/>	<input type="button" value="추가"/>	
Source CIDR	프로토콜	시작 포트/ICMP 유형	종료 포트/ICMP 코드	상태	작업
<input type="checkbox"/> 0.0.0.0/0	TCP	443	443	Active	 
<input type="checkbox"/> 0.0.0.0/0	TCP	80	80	Active	 

전체 2 개 항목 < 1 > 10 / 쪽

상세 방화벽 포트 포워딩 **부하 분산** 이벤트 코멘트

* 이름
 * Public 포트
 * 사실 포트

Source CIDR 목록
 알고리즘 라운드 로빈
 프로토콜 TCP
 오토스케일 아니오
 가상머신 추가 추가

<input type="checkbox"/>	이름	Public 포트	사실 포트	알고리즘	Source CIDR 목록	프로토콜	상태	엄격성	가상머신 추가	오토 스케일	작업
<input type="checkbox"/>	a1b51d27a251f414e84a2a3478a337cb-tcp-80	80	31619	Round-robin		TCP	Active	Configure	+ 추가		
<input checked="" type="checkbox"/>	K8S-WebService-Cluster-node-195ae069381						10.1.1.114				
<input checked="" type="checkbox"/>	K8S-WebService-Cluster-node-195ae06b488						10.1.1.128				
<input checked="" type="checkbox"/>	K8S-WebService-Cluster-node-195ae06d9e5						10.1.1.171				
<input type="checkbox"/>	a1b51d27a251f414e84a2a3478a337cb-tcp-443	443	32331	Round-robin		TCP	Active	Configure	+ 추가		
<input checked="" type="checkbox"/>	K8S-WebService-Cluster-node-195ae069381						10.1.1.114				
<input checked="" type="checkbox"/>	K8S-WebService-Cluster-node-195ae06b488						10.1.1.128				
<input checked="" type="checkbox"/>	K8S-WebService-Cluster-node-195ae06d9e5						10.1.1.171				

전체 2 개 항목 < 1 > 10 / 쪽

확인한 IP(예에서는 10.10.31.43) 주소를 웹 브라우저에서 조회하면 등록된 index.html의 내용이 표시됩니다.

ABLECL^{UD}

All about data & cloud

주식회사 에이블클라우드 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



공식홈페이지