Kubernetes 로더 밸런서 - Service

대상 컨테이너 생성

kubectl create deploy nginx --image=nginx

```
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl create deploy nginx --image=nginx --namespace default
deployment.apps/nginx created
```

```
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl get all -n default
NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-6799fc88d8-bk7pp   1/1     Running   0          47s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   130m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   1/1     1            1           47s

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-6799fc88d8   1         1         1       48s
```

생성 확인

Service basic template

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: ClusterIP/NodePort/LoadBalancer
```

3개의 Type 중 하나로 생성할 수 있음

Kubernetes 서비스 타입 비교 (ClusterIP/NodePort/LoadBalancer)

Service : 쿠버네티스의 Pod로 접근하기 위해 네트워크로 노출시키는 것

ClusterIP	NodePort	LoadBalancer
-. 클러스터 내부에서만 접근할 수 있는 IP를 할당함. -. 외부에서는 접근할 수 없으로 port forwarding 또는 proxy를 통해 접근을 해야 함. -. default 설정	-. 노드의 특정 포트를 사용하여 접근하는 방식, 포트당 하나의 서비스만 사용 가능 -. 30000-32767범위 내 포트를 사용 -. 노드가 사라졌을 때 자동으로 다른 노드를 통해 접근이 불가능	-. 노드 포트 앞단에 특정 Load Balancer를 사용하여 접근하는 방식

Service 생성 - ClusterIP Type

kubectl expose deploy nginx --type=ClusterIP --port=80 --target-port=8080

```
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl expose deploy nginx --type=ClusterIP --port=80 --target-port=8080
service/nginx exposed
```

 # 생성된 ClusterIP 정보확인

```
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl get service
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.100.0.1      <none>        443/TCP   141m
nginx        ClusterIP   10.100.69.236   <none>        80/TCP    51s
root@labs-910775232:/home/project/container-orchestration/yaml# kubectl get service nginx -o yaml
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: "2022-03-17T04:41:29Z"
  labels:
    app: nginx
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:metadata:
        f:labels:
          .: {}
          f:app: {}
      f:spec:
        f:ports:
          .: {}
          k:{"port":80,"protocol":"TCP"}:
            .: {}
            f:port: {}
            f:protocol: {}
            f:targetPort: {}
        f:selector:
          .: {}
          f:app: {}
        f:sessionAffinity: {}
```

Service 생성 - NodePort Type

`kubectl expose deploy nginx --type=NodePort --port=80 --target-port=8080`

```
root@labs-910775232:/home/project/Kubernetes-basic# kubectl expose deploy nginx --type=NodePort --port=80 --target-port=80
service/nginx exposed
root@labs-910775232:/home/project/Kubernetes-basic# kubectl get service

NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.100.0.1     <none>        443/TCP        143m
nginx        NodePort    10.100.12.48   <none>        80:32449/TCP   5s
root@labs-910775232:/home/project/Kubernetes-basic#
root@labs-910775232:/home/project/Kubernetes-basic# kubectl get service nginx -o yaml

apiVersion: v1
kind: Service
metadata:
  creationTimestamp: "2022-03-17T04:44:19Z"
  labels:
    app: nginx
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:metadata:
        f:labels:
          .: {}
          f:app: {}
      f:spec:
        f:externalTrafficPolicy: {}
```

Service 생성 - LoadBalancer Type

kubectl expose deploy nginx --type=LoadBalancer --port=80 --target-port=8080

```
root@labs-910775232:/home/project/Kubernetes-basic# kubectl expose deploy nginx --type=LoadBalancer --port=80 --target-port=80
service/nginx exposed
root@labs-910775232:/home/project/Kubernetes-basic# kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-6799fc88d8-bk7pp   1/1     Running   0          35m
pod/siege                    1/1     Running   0          24m

NAME                 TYPE           CLUSTER-IP     EXTERNAL-IP                                                                PORT(S)        AGE
service/kubernetes   ClusterIP      10.100.0.1     <none>                                                                     443/TCP        165m
service/nginx        LoadBalancer   10.100.72.57   ab780d8a341ea419bbfccfe13b29fc35-1481889596.ap-northeast-2.elb.amazonaws.com   80:30160/TCP   22s

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   1/1     1            1           35m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-6799fc88d8   1         1         1       35m
```