

# 3주차- 쿠버네티스 첫 걸음 & 매니페스트와 파드

## 쿠버네티스 첫 걸음

### 1. 클러스터 구성확인

- a. K8s 클러스터 환경의 정보 출력(macOS 미니쿠베)

```
vagrant@minikube:~$ kubectl cluster-info
Kubernetes master is running at https://10.0.2.15:8443
CoreDNS is running at https://10.0.2.15:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
vagrant@minikube:~$ kubectl get node
NAME          STATUS    ROLES          AGE      VERSION
minikube     Ready    control-plane,master 6m12s   v1.21.2
```

- b. 싱글 노드 K8s 클러스터에서 구성 노드 출력

```
vagrant@minikube:~$ kubectl get node
NAME          STATUS    ROLES          AGE      VERSION
minikube     Ready    control-plane,master 6m12s   v1.21.2
```

- ii. Role : 책과 다르게 control-plane 이 나오는 이유?

### 2. 파드 실행

- a. 파드는 쿠버네티스에서 컨테이너를 실행하는 최소 단위.

- i. kubectl : K8s 클러스터를 조작하기 위해 사용되는 커맨드
- ii. run : 컨테이너 실행을 명령하는 서브 커맨드
- iii. hello-world : 쿠버네티스 오브젝트의 이름(파드나 컨트롤러 등)
- iv. --image=hello-world : 컨테이너의 이미지, 쿠버네티스에서는 파드 단위로 컨테이너가 기동되며 리포지터리명이 생략된 경우에는 도커 허브를 사용
- v. -it : 도커에서의 -it와 마찬가지로, -i는 키보드 표준 입력에 연결하고, -t는 유사터미널과 연결하여 대화 모드 설정. 옵션 --restart=Never인 경우에만 유효하며, 그 외에는 백그라운드로 실행.
- vi. --restart=Never : 이 옵션에 따라 파드의 기동 방법 변경. Never는 직접 파드가 기동되며 Always나 OnFailure는 컨트롤러를 통해 파드가 기동

```
vagrant@minikube:~$ kubectl run hello-world --image=hello-world -it --restart=Never

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

- b. 파드 실행

```
vagrant@minikube:~$ kubectl get pod
NAME          READY    STATUS    RESTARTS   AGE
hello-world   0/1     Completed 0           5m39s
```

- c. 파드목록 출력

## 매니페스트와 파드

### 1. 매니페스트

- a. 매니페스트란 쿠버네티스의 오브젝트를 생성하기 위한 메타 정보를 YAML이나 JSON으로 기술한 파일이다.
- b. 매니페스트는 kubectl run nginx --image=nginx:latest --restart=Never 를 실행한 것과 같은 의미를 가진다.

### 2. 파드API

- a. apiVersion : v1 설정
- b. Kind : Pod설정
- c. Metadata : 파드의 이름을 지정하는 name은 필수 항목이며, 네임스페이스 내에서 유일한 이름이어야 함.
- d. Spec : 파드의 사양을 기술

### 3. 파드의 사양

- a. Containers : 컨테이너의 사양을 배열로 기술

- b. initContainers : 초기화 전용 컨테이너의 사양을 배열로 기술.
  - c. nodeSelector : 파드가 배포될 노드의 레이블을 지정.
  - d. Volumes : 파드 내 컨테이너 간에 공유할 수 있는 볼륨을 설정
4. 컨테이너 설정
- a. Image : 이미지의 리포지터리명과 태그
  - b. Name : 컨테이너를 여러 개 기술할 경우 필수 항목
  - c. livenessProbe : 컨테이너 애플리케이션이 정상적으로 동작 주인지 검사하는 프로브
  - d. readinessProbe : 컨테이너 애플리케이션이 사용자의 요청을 받을 준비가 되었는지 검사하는 프로브
  - e. Ports : 외부로부터 요청을 전달받기 위한 포트 목록
  - f. Resources : cpu와 메모리 요구량과 상한치
  - g. volumeMounts : 파드에 정의한 볼륨을 컨테이너의 파일 시스템에 마운트하는 설정. 복수 개 기술 가능
  - h. Command : 컨테이너 가동 시 실행할 커맨드. args가 인자로 적용
  - i. Args : command의 실행 인자
  - j. Env : 컨테이너 내에 환경 변수를 설정
5. 매니페스트의 적용과 확인
6. 파드를 만드는 매니페스트 적용 & 파드의 상태 목록

```

daegyeom.kim ~/15_DandK/step07/manifest master ? ls
nginx-pod.json nginx-pod.yml
daegyeom.kim ~/15_DandK/step07/manifest master ? kubectl apply -f nginx-pod.yml
pod/nginx created
daegyeom.kim ~/15_DandK/step07/manifest master ? kubectl get po
NAME READY STATUS RESTARTS AGE
nginx 0/1 ContainerCreating 0 11s
daegyeom.kim ~/15_DandK/step07/manifest master ?

```

7. 파드의 IP 주소와 파드가 배포된 노드 표시

```

daegyeom.kim ~/15_DandK/step07/manifest master ? kubectl get po nginx -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
nginx 1/1 Running 0 2m7s 172.17.0.3 minikube <none> <none>
daegyeom.kim ~/15_DandK/step07/manifest master ?

```