

## 2주차 과제 상세 안내서

목적	해당 과제는 실제로 개발된 Application이 어떻게 Cloud환경에서 동작이 되는지 이해를 하고 Yaml파일 작성에 대한 이해를 위한 과제 입니다. 여러분이 Application을 개발하는 것이 아니라 Cloud를 관리 하시는 업무를 하시게 된다면 직접 수행 하시는 일은 없을 것입니다. 하지만 Cloud를 관리 하기 위해서는 해당 과정을 직접 해보시고 이해를 하셔야 합니다.
----	--

### Main 과제

※ 주제 : application을 만들어 해당 application을 container image로 제작하고 그것을 Kubernetes에 올려 정상 동작 하는지 확인 하는 과제입니다. 비교적 코딩 및 동작 확인이 쉬운 node.js를 통한 web application을 선정하였습니다.

- 1) node.js를 이용한 간단한 web application을 만드세요
- 2) Docker file을 작성하여 Docker build를 통해 application을 container image로 제작하세요
- 3) Docker image repository(Docker hub 추천)에 image를 올리세요
- 4) Docker repository에 올린 image를 다운 받아 Kubernetes Cluster에 설치하는 **Yaml파일 (Deployment)**을 작성하시고 실제로 적용 해보세요.
- 5) **Service object**를 이용해 Cluster가 구성되어 있는 pc 뿐만 아니라 다른 pc에서도 접속 할 수 있도록 해보세요.

### Expert 과제

※ 주제 : Socket 통신을 사용한 간단한 Server, Client Program을 제작 하시고 해당 Program을 Kubernetes Cluster에서 정상적으로 동작 할 수 있도록 구성하는 과제입니다.

프로그래밍 언어는 자유롭게 선택하시면 되며 각 Program의 역할은 다음과 같습니다.

Server: socket을 열고 Client의 신호를 기다립니다. 신호가 오면 특정 값을 1씩 증가 시켜 stdout(ex: printf) 으로 출력시킵니다.

Client: Server에게 주기적으로 신호를 전송합니다.

⇒ 고려 사항

- 1) 여러분이 Server Program을 배포 할 시 pod의 ip는 Kubernetes에서 랜덤으로 정해진 ip를 사용하게 됩니다. Client에서 Server에 신호를 전송 할 때 위 문제를 어떻게 해결할지 고려해 보셔야 합니다.

### 과제 작성 방법

※ 아래에 해당되는 파일들을 압축하여 전달 해주세요

Main 과제)

- 1) Container image를 만들기 위해 작성한 Dockerfile.
- 2) application을 Kubernetes Cluster에 올리기 위해 작성한 Yaml 파일
- 3) kubectl get pods -A -o wide, kubectl get svc -A 명령어의 결과
- 4) Application에 접속한 web 화면 Capture

Expert 과제)

- 1) Container image를 만들기 위해 작성한 Dockerfile.
- 2) 작성하신 Source Code
- 3) application을 Kubernetes Cluster에 올리기 위해 작성한 Yaml 파일
- 4) `kubectl get pods -A -o wide`, `kubectl get svc -A` 명령어의 결과
- 5) `kubectl logs ($server pods name)` 출력 결과