3주차 과제

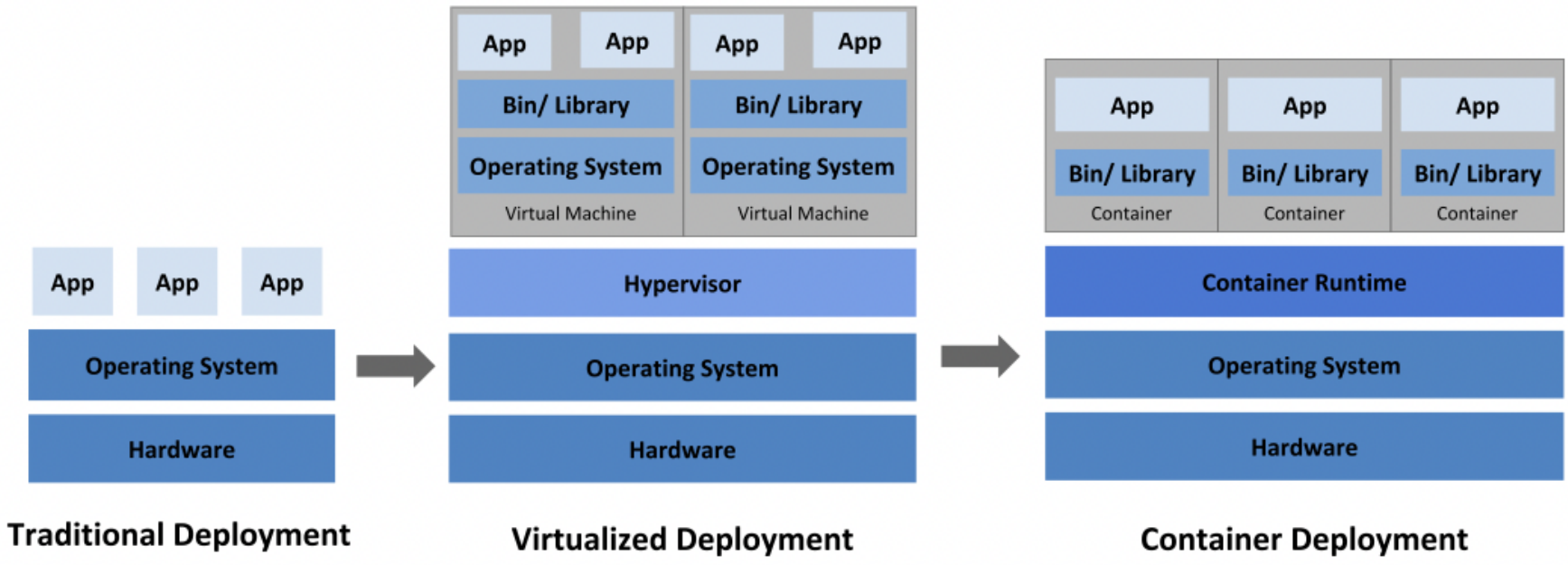
Kubernetes란? 사용하는 이유? 서비스 배포 방법

- ***Kubernetes(쿠버네티스) 란?***

  쿠버네티스는 컨테이너화된 workload와 서비스를 관리하기 위한 확장가능한 오픈소스 플랫폼이다. 이는 자동화를 간편하게 해주는 데에 큰 장점을 가지고 있다.

  Kubernetes → K8s

- ***사용하는 이유?***



(1) 전통적인 배포 시대: 물리 서버에서 실행

하지만 하나의 물리 서버에서 여러 application을 실행하기 되면 resource전체를 사용하게 되는 인스턴스가 있을 수 있으므로, 이로 인해 다른 applicaiton의 성능 저하가 될 수 있다.

→ 이를 해결하기 위해 다른 여러 대의 물리 서버에서 각각의 application을 실행하는 방법이 있겠다. 그러나 이렇게 사용하게되면 resource의 낭비가 발생할 수 있다는 점에서 확장성이 낮으며 또한 물리 서버를 많이 유지하기 위해서 많은 비용이 필요하다.

→ 해결책: 가상화

(2) 가상화된 배포 시대: 물리서버에 대한 해결책

단일 물리 서버의 CPU에서 여러 가상 머신(VM)을 실행할 수 있다. 가상화를 사용하면 VM간에서 application을 격리하여 application 간에서 데이터를 access할 수 없으므로 일정 수준의 보안성도 제공한다.

물리서버의 resource를 보다 효율적으로 활용가능하며 쉽게 application을 추가, 업데이트 할 수 있고 여러 대의 물리서버를 사용하는 것에 비해 비용 절감을 할 수 있으며 더 나은 확장성을 가지고 있다.

각각의 가상머신은 가상화된 하드웨어 상에서 자체적으로 OS를 포함한 모든 구성요소를 실행하는 하나의 완전한 머신이다.

(3) 컨테이너 개발 시대:

컨테이너는 가상머신과 유사하지만 격리 속성을 완화하여 application간에 OS를 공유한다. 그러므로 가상화에 비해서 컨테이너는 가볍다. 가상머신도 마찬가지로 컨테이너에는 자체 파일 시스템, CPU점유율, 메모리, 프로세스 공간 등이 있다. 기본 인프라와의 종속성을 끊어 클라우드나 OS배포본에 모두 적용가능하다.

⇒ 컨테이너의 추가적인 장점:

- 가상머신 이미지를 사용하는 것에 비해 컨테이너 이미지 생성이 보다 쉽고 효율적이다.
- 안정적이고 주기적으로 컨테이너 이미지를 빌드해서 배포가능하고 효율적으로 롤백이 가능하다
- 개발과 운영이 분리되어있다. 배포 시점이 아닌 빌드, release시점에서 application 컨테이너 이미지를 생성하기 때문
- 일관성: 개발환경과 같게 클라우드에서도 동일하게 작동한다
- 클라우드 및 OS간의 범용성: 어디에서든 작동한다
- 리소스 격리(성능 예측 가능), 자원을 고효율로 사용

- ***서비스 배포 방법 (아래 실습진행)***

https://kubernetes.io/

https://cloudacode.com/tutorials/cloud/aws/amazon-eks-setup/

AWS EKS는 무엇인지? https://aws.amazon.com/ko/eks/faqs/

- ***AWS EKS란 무엇인가?***

  자체 쿠버네티스 제어 영역이나 작업자 노드를 설치 및 운영할 필요 없이 AWS에서 쿠버네티스를 손쉽게 실행할 수 있도록 지원하는 관리형 서비스이다.

- ***EKS를 사용해야하는 이유?***

  EKS는 여러 AWS 가용영역에 걸쳐 API서버 및 백엔드 지속성 계층을 포함하여 쿠버네티스 제어 영역을 프로비저닝하고 확장한다.

  비정상적인 제어 영역 노드를 자동으로 감지하고 교체하며 제어 영역에 패치를 적용한다. AWS Fargate를 사용하여 EKS를 실행할 수 있다. Fargate에서는 서버를 프로비저닝하고 관리할 필요가 없어 application별로 resource를 지정하고 관련비용을 지불할 수 있고, application을 격리하여 보안 성능을 향상시킬 수 있다.

  다양한 AWS서비스와 통합되어 확장성 및 보안을 제공한다.

- ***EKS의 작동 방식***

  높은 수준에서 쿠버네티스 컨테이너를 실행하는 '작업노드' 클러스터와 제어영역이라는 2가지 주요 구성 요소로 이루어져 있다

  EKS가 없으면 쿠버네티스 제어영역과 작업자 노드 클러스터를 모두 직접 실행해야 한다. EKS를 사용함으로 EKS콘솔, CLI또는 API에서 단일 명령을 사용하여 작업자 노드를 포로비저닝한다. AWS는 고가용성의 보안 구성으로 쿠버네티스 제어영역의 프로비저닝, 크기 조정 및 관리작업을 처리한다. 운영 부담을 줄이고 AWS인프라를 관리하는 대신 application구축에 집중할 수 있다.

- ***EKS 운영체제***

> Amazon EKS는 Kubernetes와 호환되는 Linux x86, ARM 및 Windows Server 운영 체제 배포를 지원합니다. Amazon EKS는 Amazon Linux 2 및 Windows Server 2019에 최적화된 AMI를 제공합니다. Ubuntu와 같은 다른 Linux 배포를 위한, EKS에 최적화된 AMI는 각 공급업체에서 제공합니다.

(옵션) 쿠버네틱스에 배포한 웹 앱과 DB 앱이 정상 기동하는지 확인

#1. 쿠버네틱스에 배포한 웹 앱 작동 (DB: sqlite → 웹 앱과 합쳐져 있음)

과제2 진행하면서 생성된 도커 이미지를 사용하여 sqlite디비를 사용하는 flask앱을 EKS에 배포 (참고:https://chang12.github.io/eks-flask/)

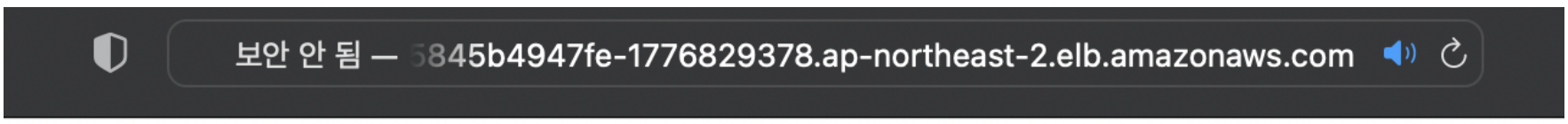
- ***EKS cluster 접속확인***

```
2022-01-27 22:23:11 [✔]  all EKS cluster resources for "cloud-eks-cluster" have been crea
ted
2022-01-27 22:23:11 [ℹ]  nodegroup "cloud-eks-workers" has 1 node(s)
2022-01-27 22:23:11 [ℹ]  node "ip-192-168-24-202.ap-northeast-2.compute.internal" is read
y
2022-01-27 22:23:11 [ℹ]  waiting for at least 1 node(s) to become ready in "cloud-eks-wor
kers"
2022-01-27 22:23:11 [ℹ]  nodegroup "cloud-eks-workers" has 1 node(s)
2022-01-27 22:23:11 [ℹ]  node "ip-192-168-24-202.ap-northeast-2.compute.internal" is read
y
2022-01-27 22:23:12 [ℹ]  kubectl command should work with "/Users/rim/.kube/config", try
'kubectl get nodes'
2022-01-27 22:23:12 [✔]  EKS cluster "cloud-eks-cluster" in "ap-northeast-2" region is re
ady
```

- ***flask todo app배포 완료***

```
rim@rim-ui-MacBookAir ~ % kubectl get services
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP
                                   PORT(S)          AGE
flask-app    LoadBalancer   10.100.30.44   a79d9b385217545efb6d15845b4947fe-1776829378.ap
-northeast-2.elb.amazonaws.com   5000:30399/TCP   14s
kubernetes   ClusterIP      10.100.0.1     <none>
                                   443/TCP          14m
```

- ***flask todo app 작동***



Task Master

Task	Added	Actions
work?	2022-01-27	Delete Update

Add Task

- ***delete cluster***

```
2022-01-27 22:40:08 [ℹ]  waiting for stack "eksctl-cloud-eks-cluster-addon-iam
2022-01-27 22:40:08 [ℹ]  waiting for CloudFormation stack "eksctl-cloud-eks-cl
2022-01-27 22:40:24 [ℹ]  waiting for CloudFormation stack "eksctl-cloud-eks-cl
2022-01-27 22:40:24 [ℹ]  deleted serviceaccount "kube-system/aws-node"
2022-01-27 22:40:25 [ℹ]  will delete stack "eksctl-cloud-eks-cluster-cluster"
2022-01-27 22:40:26 [✔]  all cluster resources were deleted
```

#2. 쿠버네틱스에 배포한 DB앱 작동 (DB: mysql → 웹 앱과 별도의 디비)

- ***Maria DB 설치 (mac OS)***

  https://mariadb.com/kb/ko/installing-mariadb-on-macos-using-homebrew/

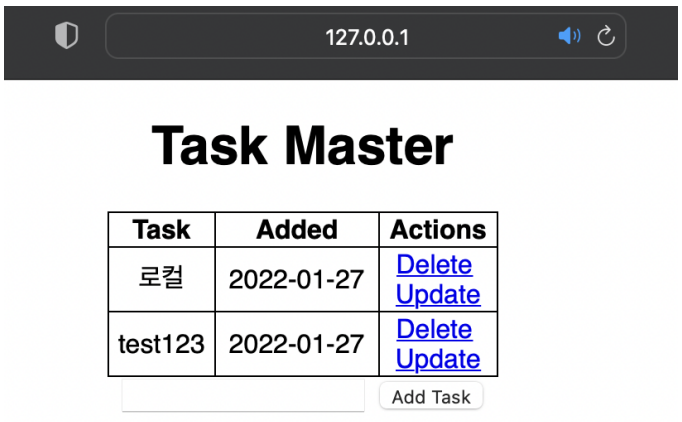
- ***Maria DB에 새로운 DB생성***

```
# todo 디비 생성
CREATE DATABASE todo;

# 생성된 디비 사용
USE todo;

# todo 테이블 생성
CREATE TABLE todo(
    id INT NOT NULL AUTO_INCREMENT,
    content VARCHAR(200) NOT NULL,
    date_created DATETIME DEFAULT(CURRENT_TIME),
    PRIMARY KEY (id)
);
```

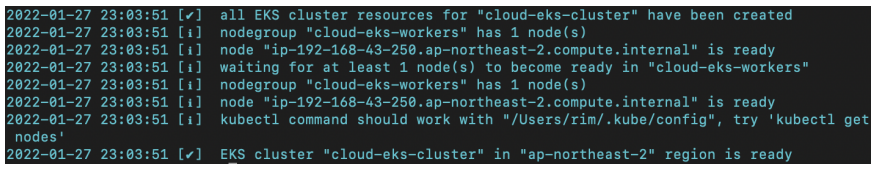
- ***로컬에서 maria DB와 연동 후 웹 앱 정상 작동***



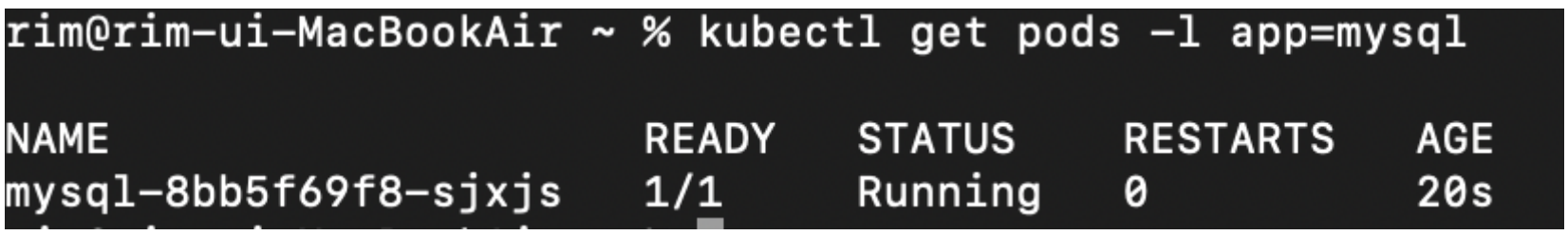
로컬에서는 maria DB생성하여 코드 수정하였고, 로컬에서 mysql DB와 연동하여 웹 앱이 잘 작동하는 것을 확인하였습니다. 그리고 수정된 코드 또한 도커 이미지로 빌드하여 도커 허브에 업로드하였습니다. (https://hub.docker.com/repository/docker/rim0703/flask-mysql)

실습을 진행하던 중 mysql을 도커에서 어떻게 작동하는지에 대해서 많은 자료를 찾아보았지만 이해가 잘 안 되어 코드를 따라 작성해보면서 **DB에 대한 도커 이미지도 생성해보려고 했지만 잘 해결되지 않았습니다**. 이에 아래의 실습은 멘토님께서 제공해주신 튜토리얼에 이미 업로드 되어 있는 이미지파일을 사용하여 EKS에 배포한 결과입니다.

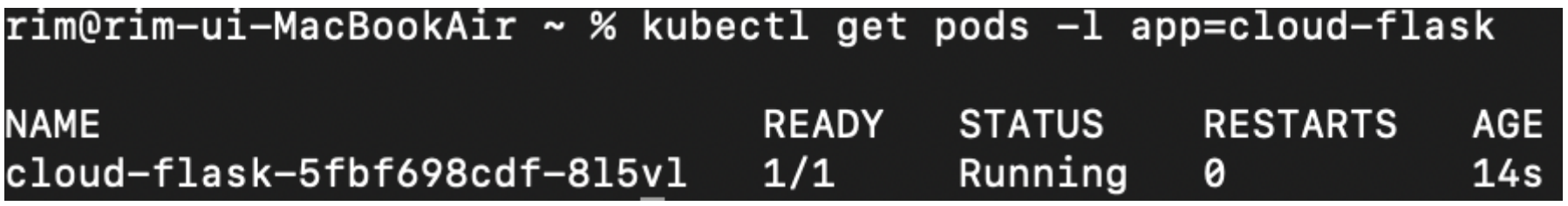
- ***EKS Cluster접속확인***



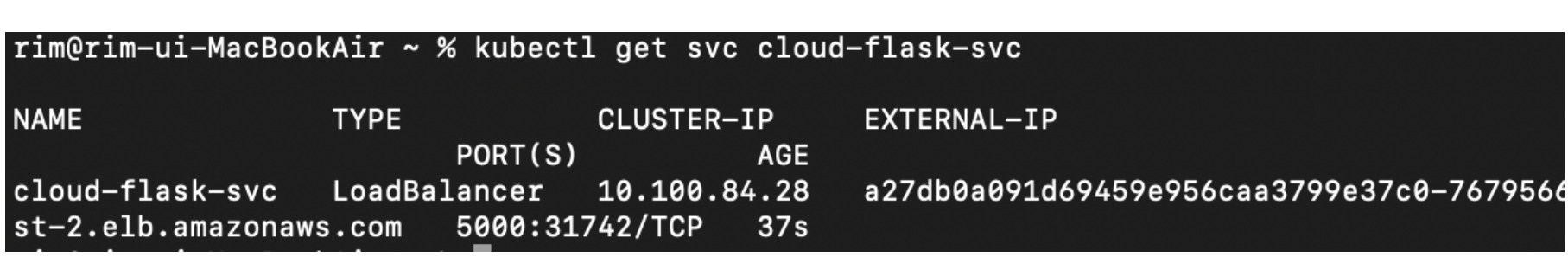
- ***DB 배포***



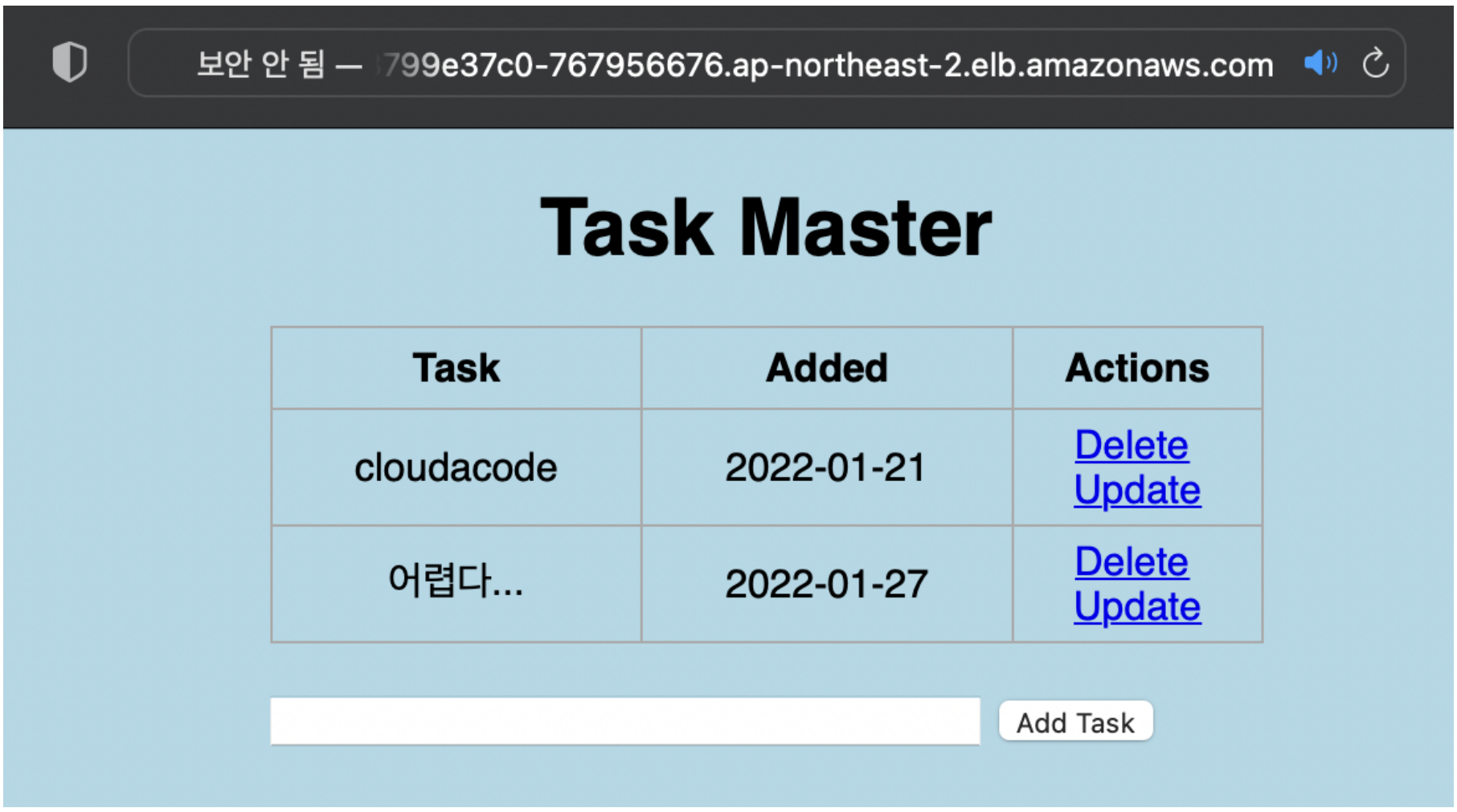
- ***Flask app배포***



- ***LB Endpoint***



- ***DB웹 앱 작동 캡쳐***



- ***delete cluster***

```
2022-01-27 23:42:42 [ℹ]  waiting for CloudFormation stack "eksctl-cloud-eks-c
ccount-kube-system-aws-node"
2022-01-27 23:43:00 [ℹ]  waiting for CloudFormation stack "eksctl-cloud-eks-c
ccount-kube-system-aws-node"
2022-01-27 23:43:00 [ℹ]  deleted serviceaccount "kube-system/aws-node"
2022-01-27 23:43:02 [ℹ]  will delete stack "eksctl-cloud-eks-cluster-cluster"
2022-01-27 23:43:02 [✔]  all cluster resources were deleted
```

질문

- 앞서 언급했듯이.. mysql DB를 도커에서 어떻게 빌드하는지에 대해서 질문드리고 싶습니다.
- EKS를 통해 배포를 진행완료 후 웹 앱이 정상적으로 작동하고 있는 백그라운드 로그(?)는 어떻게 확인할 수 있을까요?
- 위에 실습에서 볼 수 있다시피 EKS cluster 하나 만들어서 한번 실습하고 delete하고 다시 클러스터 생성하는 방식으로 실습을 진행하였습니다. 생성된 하나의 cluster를 사용하여 이전에 배포된 것(sqlite버전의 앱)을 새로 배포 될 것(app+mysql DB)으로 덮어씌우는 것도 진행이 가능할까요? (프리티어를 사용하는 계정이라고 해도 EKS는 따로 요금이 부과된다고 해서ㅠㅠ 많은 시도를 해보지는 못하였습니다.)