

애플리케이션 서비스가 생성 및 배포됐다. 실행되고 있는 컨테이너를 확인해보자.

```
(venv)$ docker ps
```

명령을 입력하면 [그림 9-4]와 같이 실행되고 있는 컨테이너가 포트 번호와 함께 표시된다.

A terminal window titled 'youngestdev@Abduls-MacBook-Air:~/Work/Building-Web-APIs-with-FastAPI-and-Py...' shows the command 'docker ps' being executed. The output is a table with columns: CONTAINER ID, IMAGE, COMMAND, CREATED, and STATUS. It lists two containers: 'planner_api_1' (event-planner-api:latest) and 'planner_database_1' (mongo).

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
84423100df3e	event-planner-api:latest	"python main.py"	2 minutes ago	Up 2 minutes
01646e18fbb2	mongo	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes

그림 9-4 현재 실행되고 있는 컨테이너 목록

GET 요청을 보내 애플리케이션이 제대로 실행되는지 확인해보자.

```
(venv)$ curl -X 'GET' \
'http://localhost:8000/event/' \
-H 'accept: application/json'
```

윈도우에서는 다음과 같이 실행한다.

```
curl -X GET "http://localhost:8000/event/" -H "accept: application/json"
```

요청에 대한 응답은 다음과 같다. 생성한 이벤트가 없으므로 빈 응답이 반환된다.

```
[ ]
```

배포한 애플리케이션이 제대로 실행되고 있다!

이번에는 계정을 생성해서 데이터베이스가 제대로 작동하는지 확인해보자.

```
(venv)$ curl -X 'POST' \
'http://localhost:8000/user/signup' \
```