



기업 서비스에서 R 활용하기

Docker와 Kubernetes로 분석 모델 실행하기

저자: 이남신

소속: 오피지지

2021-11-19

발표 개요

데이터 분석에서 R은 굉장히 유용한 언어입니다. 지원하는 패키지가 다양하고 사용법도 직관적이어서 수많은 데이터 분석가들이 R을 활용하고 있습니다.

하지만 실제로 제작한 분석 모델이 좋은 퍼포먼스를 보여주더라도 실제 서비스에 적용할 수 없거나 과한 공수가 필요하다면 서비스 관점에서 효율적인 분석 모델이라 할 수 없습니다.

따라서 실제 웹/앱 서비스에 연계되는 데이터 분석 모델을 만드는 데이터 분석가들은 언제나 서비스에 어떻게 연동해야 할지 고민해야 합니다.

이번 발표에서는 Docker와 Kubernetes를 활용하여 R 분석 모델을 서비스에 쉽게 적용하는 방법에 대해 말씀드리려 합니다.

1. Docker
2. Docker로 이미지 만들기
3. Kubernetes
4. Kubernetes로 분석하기

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

구동 중인 컴퓨터의 상태와 관계 없이
내가 프로세스를 실행하고 싶은 환경을 만드는 것

가상 머신과 유사하지만
실제로 OS를 설치하지 않고 프로세스만 격리해서 활용

MAC에서 Ubuntu container를 실행하면 OS를 ubuntu로 인식

개발자에게 그냥 코드만 전달해준다면...

코드 실행했는데 에러가 나요.

패키지 설치가 안 되어있는데요?

실행에 권한이 필요한데요?

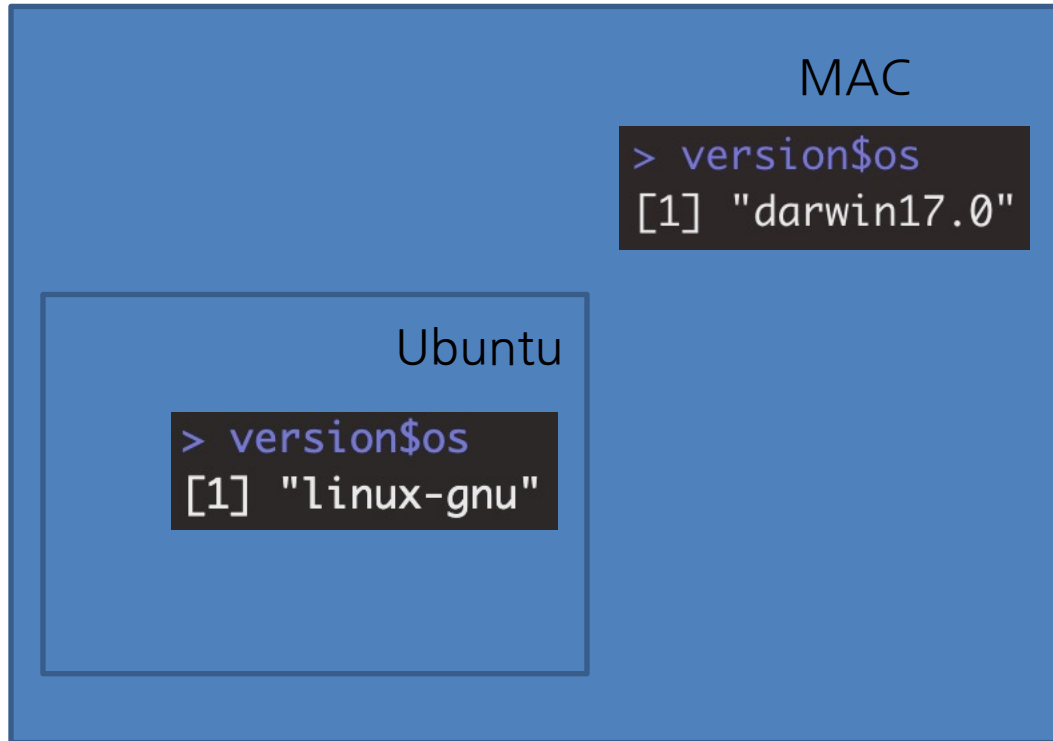
폰트가 없다는데...

패키지가 업데이트되면서 parameter가 바뀐 것 같아요.

환경 변수가 잘못 설정된 것 같은데...

인코딩이 달라요

아... 이거 다음에 반영할게요



코드 + 실행한 환경(container image)를 같이 전달

동일한 실행 결과 보장

Docker Image

특정 실행 환경을 image라 합니다.

Image에는 OS, 환경 변수, 프로그램, 파일 등이 포함됩니다.

사용할 image 만드는 방법

- Image registry에서 좋은 image 찾아쓰기
- Base image를 실행하고 수동으로 설정하기
- 내가 원하는 대로 설정할 Dockerfile을 작성해서 build하기

Docker Image

Image registry에서 찾기

- <https://hub.docker.com/> 에서 원하는 이미지 검색
- 예시
 - ubuntu에 R만 깔려있는 이미지
 - R과 Rstudio가 깔린 이미지
 - R, Rstudio 및 여러 패키지(tidyverse 등)가 설치된 이미지
 - 예전 버전 R(3.5.0)이 깔린 이미지

Docker Image

Base image에서 수동으로 설정하기

- <https://hub.docker.com/> 에서 원하는 이미지 검색
 1. Ubuntu만 깔린 base image 다운로드
 2. Ubuntu container 실행하기
 3. R 및 Rstudio, 필요한 패키지 설치하기(수동)
 4. 설치한 상태의 container를 새로운 image로 commit하기

다른 사용자가 해당 image에 대한 자세한 description을 받기 힘들다

Docker Image

Dockerfile 작성해서 build하기

- <https://hub.docker.com/> 에서 원하는 이미지 검색
 1. Ubuntu만 깔린 base image 다운로드
 2. 프로그램(R, Rstudio) 설치, 파일 추가, 환경 변수 설정하는 Dockerfile 작성
 3. Dockerfile을 build하여 새로운 image 생성

다른 사용자도 Dockerfile을 읽고 해당 image에 대한 정보를 얻을 수 있음

Dockerfile 예시

```

# Dockerfile

# base 이미지
FROM ubuntu:20.04

# Timezone 설정
ENV TZ Asia/Seoul
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

# 설치
RUN apt update
RUN apt install r-base -y

# 컨테이너가 실행될 때 실행할 기본 명령
CMD R

```

```

# 맥 터미널에서
# Dockerfile이 위치한 디렉토리에서 실행

$ docker build -t r-on-ubuntu:1.0.0 .

# 만든 image를 기반으로 container 실행하기
$ docker run -it r-on-ubuntu:1.0.0

```

Kubernetes

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

여러 개의 containers를 자동으로 관리하는 orchestration tool

- 끝난 프로세스 종료
- 실패한 프로세스 복구, 재실행
- 사용자가 설정해둔 규칙에 따라 리소스 자동 분배
- 서비스 상태에 따라 flexible하고 기민하게 변경
- Desired state만 설정하면 알아서 실행

Desired State

Kubernetes는 ‘어떻게’를 고민할 필요가 없습니다.

원하는 작업 상태만 알려주면 방법은 kubernetes가 알아서 찾습니다.

Manifest 작성하기

1. 어떤 리소스를 얼마나 사용하는지
2. 어떤 이미지를 이용해 컨테이너를 만들 것인지
3. 몇 개의 컨테이너를 사용하는지
4. 어떤 명령을 실행하는지
5. 주기적으로 실행되는지, 일회성인지
6. 종속성이 있는 서비스는 무엇인지
7. 에러가 났을 때 어떤 식으로 대응할 것인지
8. 부하가 커질 때 어떻게 대응할 것인지
9. 네트워크 설정은 어떤지 등

Cronjob 등록하기

주기적으로 R로 된 분석 코드를 실행하는 cronjob을 등록합니다.

```

apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: analysis-cron
spec:
  schedule: "0 * * * *" # 매 시간 실행되는 cronjob
  concurrencyPolicy: Forbid # 앞 job이 에러로 밀려서 동시에 실행되어야 할 때의 처리 정책
  jobTemplate:
    spec:
      template:
        spec:
          restartPolicy: OnFailure
          containers:
            - name: lol-analysis-cron
              # R 스크립트를 실행할 image
              image: *****.dkr.ecr.ap-northeast-2.amazonaws.com/analysis-r:latest
              imagePullPolicy: IfNotPresent
              resources:
                requests:
                  memory: "20Gi" # 최소로 확보해야 하는 RAM
              command: # 컨테이너가 실행할 command
                - sh
                - -c
                - |
                  /bin/sh << 'EOF'
                  Rscript /root/analysis_cron.R # 실행할 Rscript
                  EOF
          envFrom: # 컨테이너 내 환경 변수
            - secretRef:
                name: secret
  
```

경청해 주셔서
감사합니다.

오피지지