

[Project News] Hello K8s

- 진행: 황수재 컨설턴트
- 날짜: 2020.09.28
- 목차
 - [📄 Project enquete](#)
 - [🏠 사업이 도착 했습니다.](#)
 1. [K8s?](#)
 2. [K8s 등장 배경](#)
 3. [자동배포, 스케일링](#)
 4. [K8s 가 할 수 있는 일들](#)
 5. [추가 자료](#)

📄 Project enquete

설문조사를 통해 얻은 데이터와 추가의견들은 프로젝트 완성도를 높일 수 있을 뿐만 아니라, 최종 발표 때에도 근거 자료로 활용할 수 있습니다.

구미 2반 1팀 프로젝트 설문조사

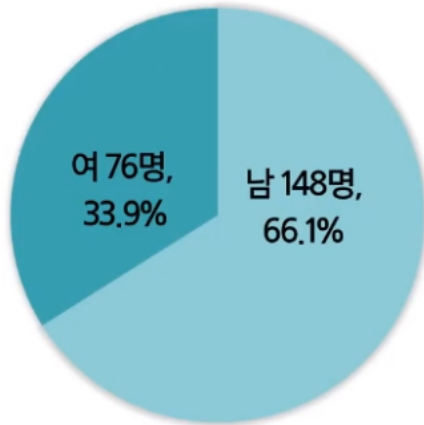
3기 뿐만 아니라 4기 교육생들 대상으로도 설문조사를 실시했습니다.

탄단지 팀 처럼 운영 프로님들의 도움을 받아 더 넓은 대상으로부터 설문조사를 할 수 있습니다.

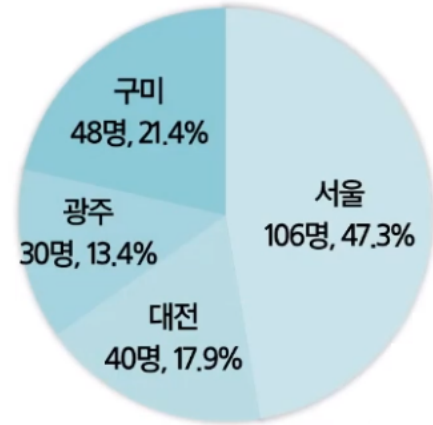
건강한 세끼

- ✓ 구미 2반 1팀 '탄단지' 팀에서 요청한 설문조사 입니다.
- ✓ '개인 입맛과 영양 성분에 따른 식단 추천, 근처 식당 추천 서비스'를 주제로 프로젝트를 진행하고 있습니다.
- ✓ 사람들이 좋아하는 음식 및 음식의 취향을 취합하여 데이터 셋을 만드는데 이용하고자 SSAFY 3기, 4기 교육생을 대상으로 설문조사를 실시했습니다.

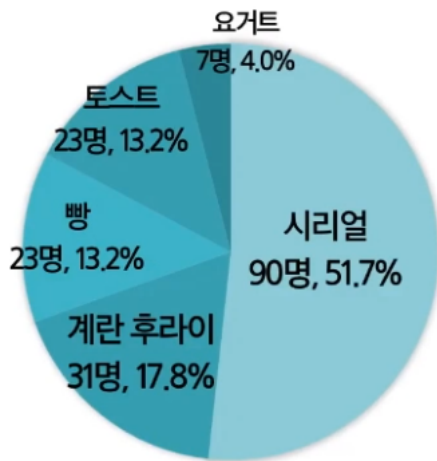
성별 비율



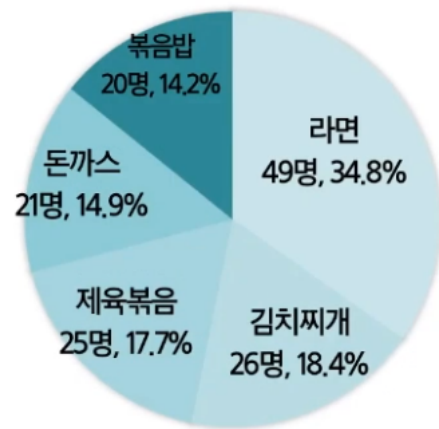
지역 비율



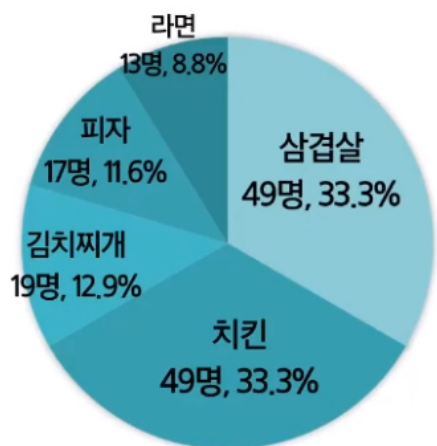
아침 Top5



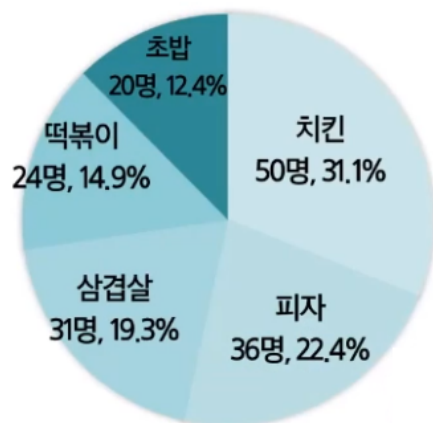
점심 Top5



저녁 Top5



최애음식 Top5



📧 사연이 도착 했습니다.

스윗한 익명 교육생님으로부터의 편지

오늘은 저희 팀을 소개해드리려고 사연을 보냅니다.
저희 팀에는 "육왕이 나르싸피" Merge왕, API왕, 알고리즘왕, 리액트왕, CSS왕, 배포왕이 있습니다.

"안 되면 되게 하라"는 정신으로
계획했던 기능을 구현하여, 만족스러운 결과를 낼 수 있을 것 같습니다.

웹엑스로만 만나다 보니 텐션이 떨어지는 날이면
랜선 화식으로 함께 맥주 한 캔 하며 기분이라도 내고 있구요.

빨리 코로나가 잠잠해지고 다시 만나는 그날까지 방역수칙 잘 지키고
개인 위생 관리하며 건강하게 지내보아요.

프로젝트 완성을 향해 달려가고 있는 SSAFY 3기 친구들,
모두 마지막까지 힘내시고 끝까지 포기하지 말고 성과를 거둔다면
우리 모두 취뽕할 수 있을 거예요. SSAFY 3기 화이팅!!



1. K8s?

인프라에 대한 이해를 해봅시다.

- Kubernetes Numeronym (i18n, ally)
 - i18n: Internationalization
 - ally: Accessibility
- 그리스어로 배의 조타수 또는 파일럿
- 구글의 15년 이상의 노하우가 담겨있는 인프라 기술로, 2014년 구글에서 오픈소스로 제공



쿠버네티스(K8s)는 컨테이너화된 애플리케이션을
자동으로 배포, 스케일링 및 관리해주는 오픈소스
시스템입니다. 🐳

2. K8s 등장 배경

1. 서버가 다운 된 경우



2. 일단 서버를 살리고, 문제를 분석합니다. 서버가 또 죽어서 서버를 또 늘립니다.



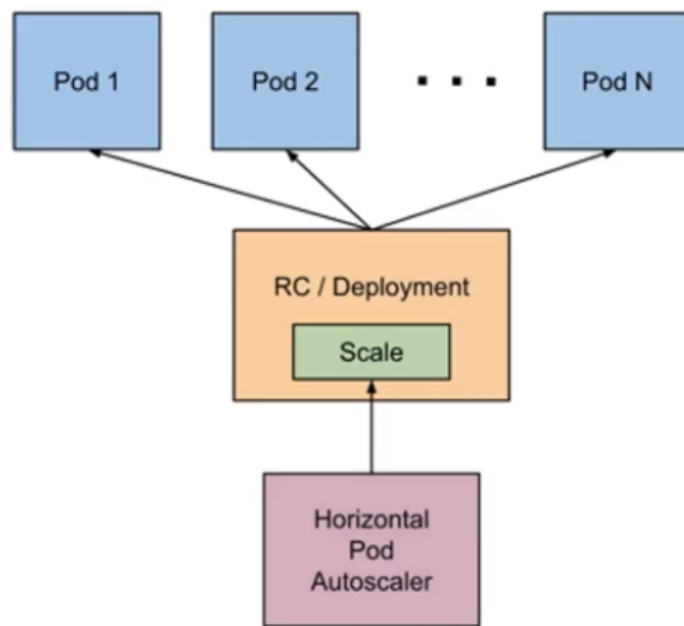
3. 비용이 많이 듭니다. 비용 감축을 위해 서버를 두개로 내립니다.

4. 그랬더니 서버가 또 다운됩니다.



5. 서버가 괜찮은지 계속해서 사람이 보고 있는 것은 쉽지 않습니다. 즉 자동화가 필요한 부분인 것입니다.

3. 자동배포, 스케일링



- Autoscaler 가 모니터링 하다가 Pod 를 하나 더 늘리거나 줄이는 일을 하는 것

파드란 무엇인가?

- [공식 문서](#)

참고: 도커가 가장 일반적으로 잘 알려진 런타임이지만, 쿠버네티스는 도커보다 컨테이너 런타임을 더 많이 지원하며, 도커의 일부 용어를 사용하면 파드를 설명하는 데 도움이 된다.

파드의 공유 컨텍스트는 리눅스 네임스페이스, 컨트롤 그룹(cgroup) 및 도커 컨테이너를 격리하는 것과 같이 잠재적으로 다른 격리 요소들이다. 파드의 컨텍스트 내에서 개별 애플리케이션은 추가적으로 하위 격리가 적용된다.

도커 개념 측면에서, 파드는 공유 네임스페이스와 공유 파일시스템 볼륨이 있는 도커 컨테이너 그룹과 비슷하다.

- 간단히 말해 Pod 하나 안에 Container 하나(App하나)가 들어있다고 생각할 수 있다.

컨테이너란 무엇인가?

정의

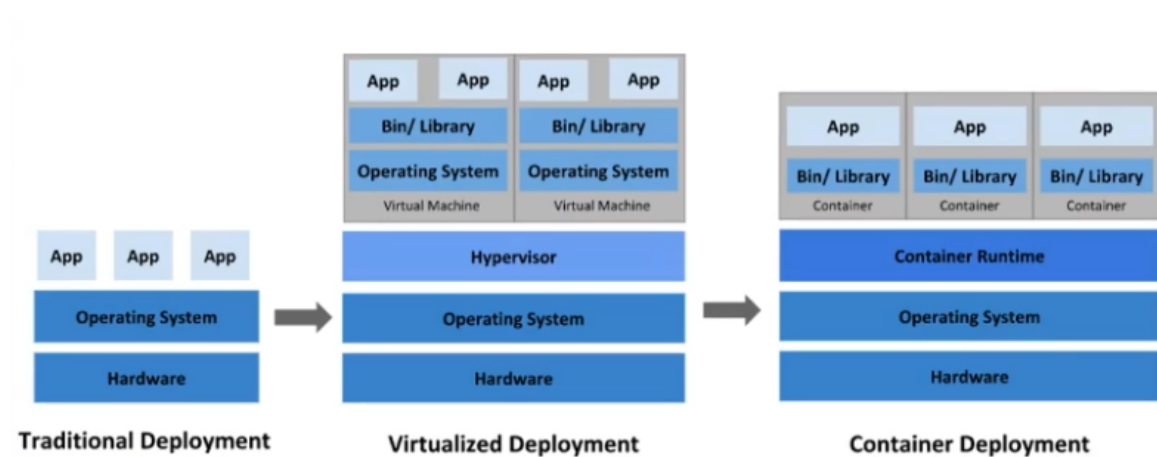
- 애플리케이션 실행에 필요한 코드와 모든 의존성들이 패키징 된 소프트웨어의 표준 단위
 - 코드, 시스템 라이브러리, 환경변수 등
 - ex) Docker

등장배경

- 로컬에서는 되는데 배포해서는 안 되는 경우가 있음

- 개발하던 환경을 배포 환경에도 그대로 적용하려는 시도로부터 등장

컨테이너 개념 정리



Traditional Deployment

- HW 와 OS 위에 APP 이 존재.
- 하나의 OS 내에서 동작하다보니 App 간 엉키거나, 의도한 바 대로 작동하지 않는 경우가 존재

Virtualized Deployment

- MacOS에서 Windows 를 설치한다거나, Windows 에서 Ubuntu 를 설치하는 등 하나의 운영체제 위에 다른 OS 를 설치하는 경우
- OS 를 다시 부팅하기 위한 시간 자체가 오래 걸림. 시스템 하나 하나가 크다는 뜻

Container Deployment

- 이전 방법을 경량화 하기 위해 꼭 필요한 Binary, Library 와 App (Code) 만 컨테이너에 담는다
- 이를 통해 컨테이너 별로 자동 배포를 할 수 있다.

4. K8s 가 할 수 있는 일들

1. 서비스 디스커버리와 로드 밸런싱 (부하 분산)
2. 스토리지 오케스트레이션 (하드디스크를 어디에서 쓸 지)
3. 자동화 된 롤아웃과 롤백
4. 자동화 된 빈 패킹
5. 자동화 된 복구
6. 시크릿과 구성 관리

5. 추가 자료

- 쿠버네티스 공식 [기초 학습](#)
 - 삼성 SDS 에서 작성 한 [초보자를 위한 쿠버네티스 소개](#)
 - Red Hat 에서 작성한 [쿠버네티스의 개념, 성능, 사용방법 및 차이점](#)
 - 배탈의 민족 [쿠버네티스를 이용해 테스트 환경 구현해보기](#)
-
- [도커와 쿠버네티스의 차이](#)
 1. 도커는 '기술적인 개념이자 도구'이고 쿠버네티스는 '도커를 관리하는 툴'
 2. 도커는 '한 개의 컨테이너'를 관리하는 데 , 쿠버네티스는 '여러 개의 컨테이너'를 서비스 단위로 관리하는 데 최적화되어있다.