



캡스톤 디자인의 최종 발표

AWS 활용 가상 머신 네트워크 시스템 구축 및 취약점 분석과 대응 솔루션 개발



#2025-06-11

#최종 발표

#태훈 업고 튀어

#20201738 서민재

#20222517 한하영

#20222021 김주령

- 프로로그
- 프로젝트 개요
- 시스템 구조
- 핵심 기능
- 주요 알고리즘
- 일정 및 계획
- 에필로그



한하영

프론트엔드
React, WebSocket,
Docker



김주령

백엔드
Spring Boot, Redis,
MySQL



서민재

가상머신 서버
Spring Boot,
Docker, K8S

무엇을 도와드릴까요?



프로로그

프로젝트 개요

시스템 구조

핵심 기능

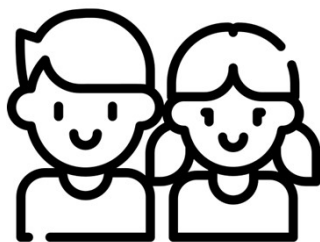
주요 알고리즘

일정 및 계획

에필로그

📌 현재 발생하는 불편 사항

- 사용자가 어떠한 **프로그램을 사용**하기 위해서 해당 프로그램을 지원하는 **운영체제를 설치**해야 하는 과정 **필연적**.
(예시 : gdb, pwndbg → 리눅스에서 작동)
- 비전공자/초보자들에게는 해당 프로그램을 사용하기 위한 운영체제 **설치 과정**에서 오류 등의 **어려움** 발생
- 주니어 개발자 / 시니어 개발자들 또한 개발환경과 운영환경 간의 차이로 인하여 서버/운영체제 학습 별도 필요
- 잠깐 사용 후 사용하지 않을 경우 **컴퓨터 자원 낭비**



비전공자, 초보자



gdb: GNU debugger, 리눅스의 대표적인 디버거
pwndbg : gdb의 플러그인



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

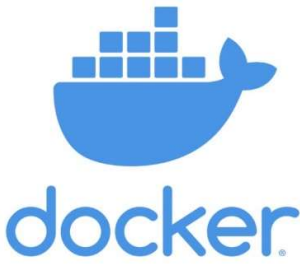
위 문제를 해결하기 위한 다른 도구들이 다수 존재하지 않은가?



가상화 기술
QEMU(Quick Emulator)

하드웨어 및 운영체제를
가상화하여 실행할 수 있는 도구

속도가 느리고 어려움



컨테이너 기술
Docker

하드웨어 리소스를 적게 사용하여
애플리케이션과 필요한 환경을 컨테이너로
독립적인 실행시켜 배포하는 도구

사전 지식 학습 필요 및 보안 위험



클라우드 기술
AWS

사용자가 하드웨어 구매/유지보수 필요 없이
필요한 만큼만 IT 자원을 빌려서 사용
인터넷을 통한 IT 인프라 제공 기술

사전 지식 요구 및 비용부담

각각 단점들이 존재



프로로그

프로젝트 개요

시스템 구조

핵심 기능

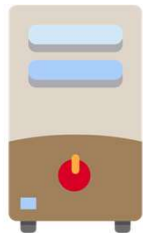
주요 알고리즘

일정 및 계획

에필로그

📌 현재 발생하는 불편 사항

하드웨어 적 한계, 기술의 어려움의 한계, 비용의 한계로 인하여, 새로운 초보 개발자, 비전공자 들의 프로그램 이용의 **진입장벽**이 높아지고 있음



저게 다 뭐지...? 저거를 알아야 해당 서비스를 이용할 수 있는 건가..? 뭘 팔아서 사용해야하네..



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

프로젝트 목표 및 비전

No Hardware Restrictions

Supports multiple OS

Unlimited space



People who use Computer

No time constraints

Secure and Reliable

사용자가 요청하는 **운영체제**를 짧은 기간동안 **1회용**으로 생성
편리하게 이용할 수 있는 웹 기반 서비스



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

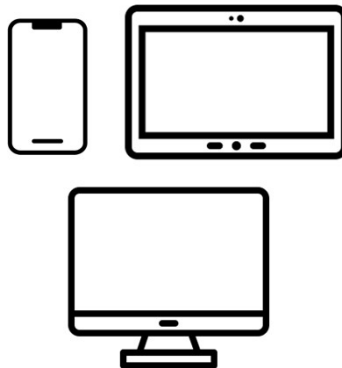
일정 및 계획

에필로그

기대 효과는?



불필요한 프로그램 설치
관리 미필요



어떤 디바이스 환경에서도
쉽고 빠른 접근



손쉬운 서버 관리 및
편리한 서버 상태 확인



실시간 악의적 이용
빠른 대응



프롤로그

프로젝트 개요

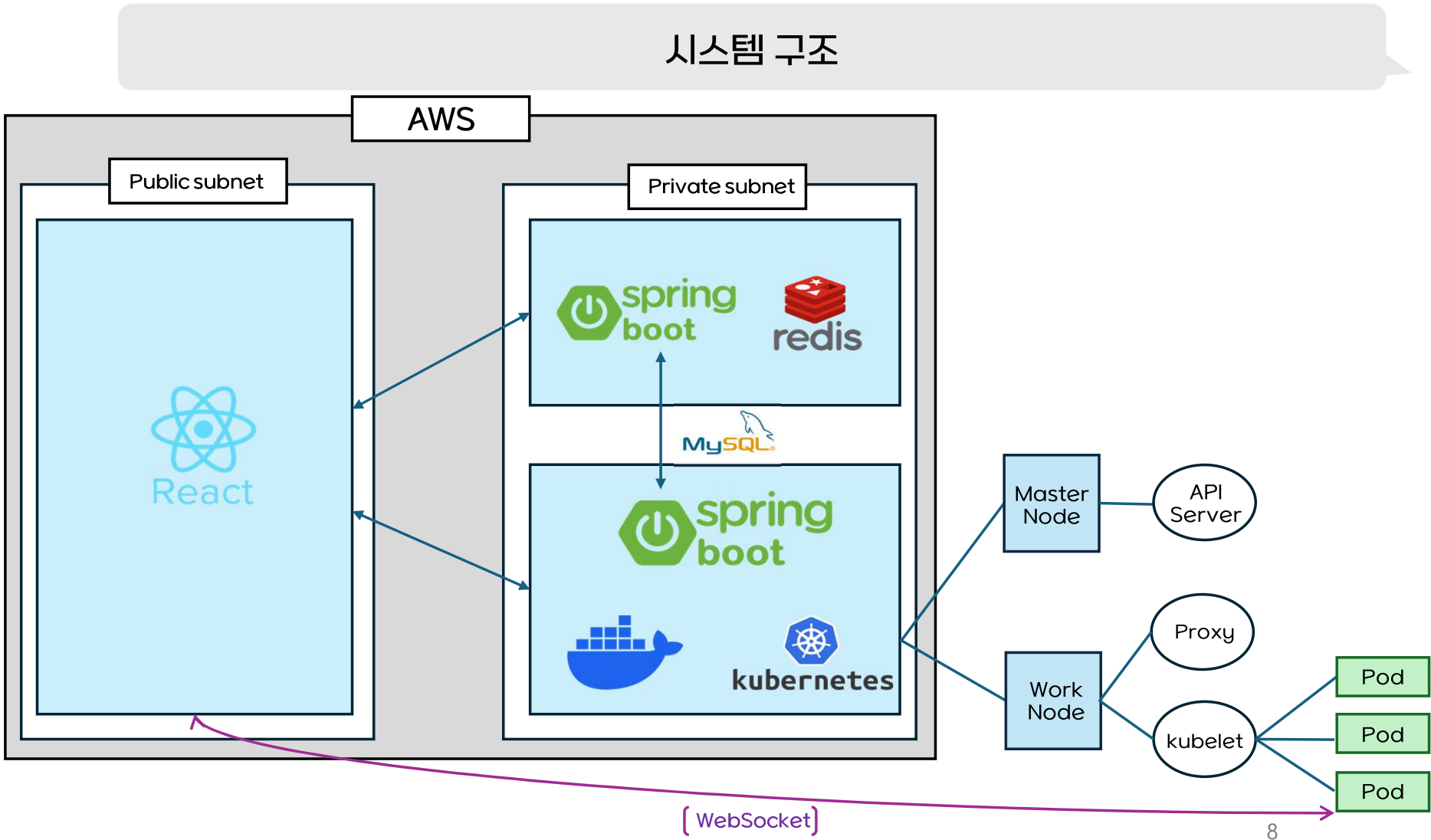
시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그



To Be Continue

📌 **핵심 기능 1: 가상 서버 생성 / 조회 / 삭제**

- 프로로그
- 프로젝트 개요
- 시스템 구조
- 핵심 기능**
- 주요 알고리즘
- 일정 및 계획
- 에필로그

서버 생성

서버 이름

서버 이름을 입력하세요

운영 체제 및 버전

Ubuntu

버전 선택

20.04

22.04

Create

Cancel

← 사용자가 서버 이름 생성

← 생성 가능 목록 불러오기

프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

가상서버 생성

Random podName
Business plan podNamespace

```
seo ~ k get pods
NAME                                                    READY   STATUS    RESTARTS   AGE
alertmanager-kube-prometheus-stack-alertmanager-0     2/2     Running   0           3d6h
kube-prometheus-stack-grafana-67bb9f4bc6-bkkgf         3/3     Running   0           3d6h
kube-prometheus-stack-kube-state-metrics-77678594d6-tl84n 1/1     Running   3 (2d17h ago) 3d6h
kube-prometheus-stack-operator-f99678f48-f4k5n         1/1     Running   0           3d6h
kube-prometheus-stack-prometheus-node-exporter-0zwnl   0/1     Pending   0           3d6h
pod-29f239f1                                            1/1     Running   2 (20d ago)   39d
pod-5f164071                                            1/1     Running   2 (20d ago)   39d
pod-673e4c1f                                            1/1     Running   0            20d
pod-c1212930                                            1/1     Running   0            128m
prometheus-kube-prometheus-stack-prometheus-0        2/2     Running   0           3d6h
seo ~
```

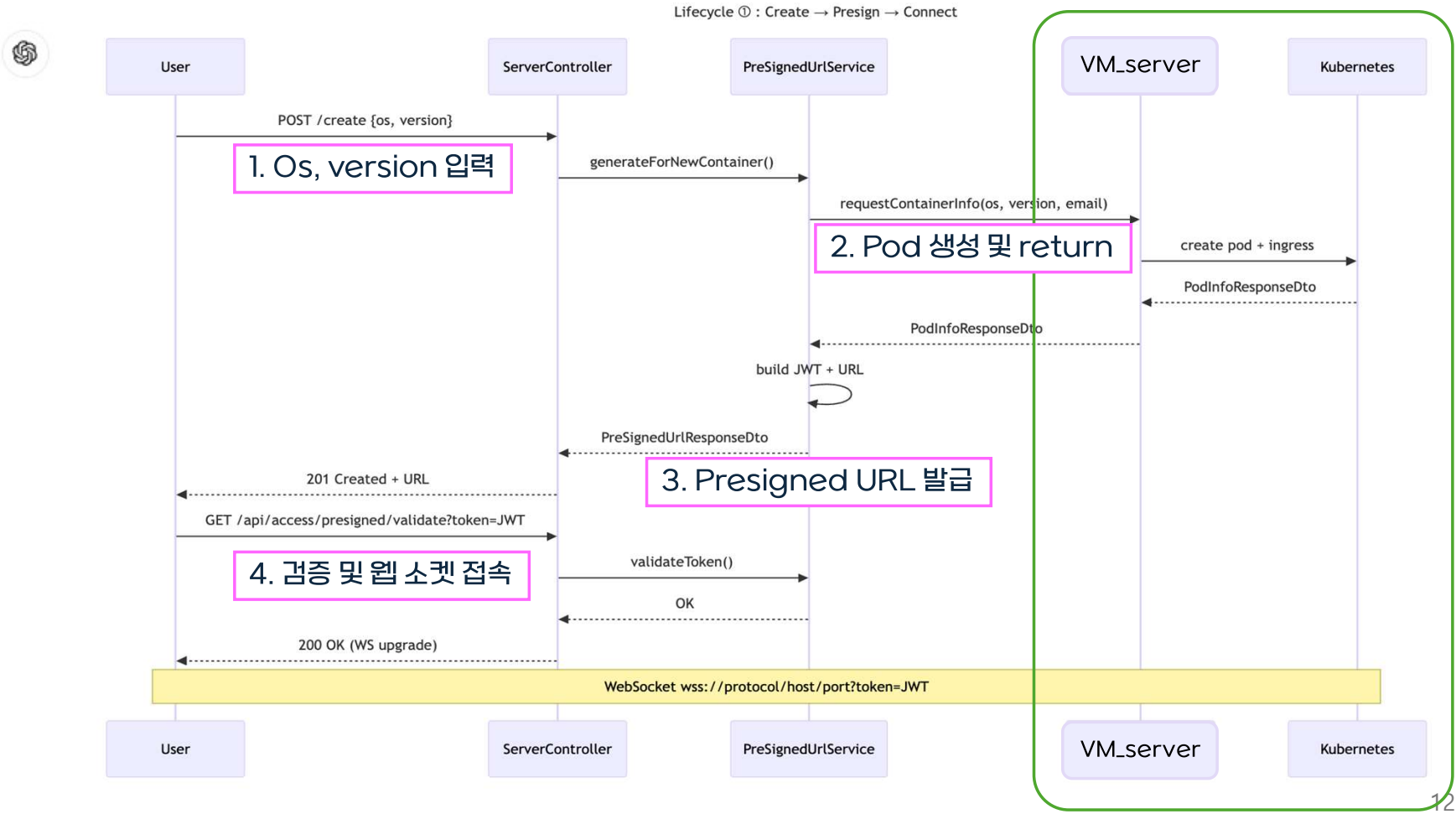
```
Database changed
mysql> select * from pod;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | created_at | os   | pod_name | pod_namespace | service_port | status | user_email | version | ingress | called_name |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 24 | 2025-04-30 | ubuntu | pod-29f239f1 | default | NULL | RUNNING | hey_minj@naver.com | 22.04 | tcar.admin.connection.com/default/pod-29f239f1 | NULL |
| 25 | 2025-04-30 | ubuntu | pod-5f164071 | default | NULL | RUNNING | hey_minj@naver.com | 22.04 | tcar.admin.connection.com/default/pod-5f164071 | NULL |
| 26 | 2025-05-19 | ubuntu | pod-673e4c1f | default | NULL | RUNNING | hanhy0219@naver.com | 22.04 | tcar.admin.connection.com/default/pod-673e4c1f | NULL |
| 29 | 2025-06-08 | ubuntu | pod-c1212930 | default | NULL | RUNNING | hanhy0219@naver.com | 22.04 | tcar.admin.connection.com/default/pod-c1212930 | NULL |
| 30 | 2025-06-09 | ubuntu | pod-0d83e12a | default | NULL | RUNNING | zooryeong@naver.com | 22.04 | tcar.admin.connection.com/default/pod-0d83e12a | hi2 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

해당 속성의 pod 생성 및 데이터베이스 저장

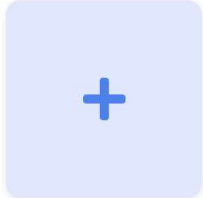
- 프로로그
- 프로젝트 개요
- 시스템 구조
- 핵심 기능
- 주요 알고리즘
- 일정 및 계획
- 에필로그

가상서버 생성 시나리오




📌 **핵심 기능 1: 가상 서버 생성 / 조회 / 삭제**

My Servers



Management


Server 1 

Status: running

TTL:

OS:

Version:

Server 2 

Status: stopped

TTL:

OS:

Version:

```
1  [
2      {
3          "podName": "pod-673e4c1f",
4          "namespace": "default",
5          "status": "RUNNING",
6          "ingressUrl": "tcar.admin.connection.com/default/pod-673e4c1f",
7          "calledName": null,
8          "accessType": "owner"
9      }
10 ]
```

서버 목록 조회 및 서버로부터 받는 **Pod의 JSON** 데이터

📌 **핵심 기능 1: 가상 서버 생성 / 조회 / 삭제**

프로로그

프로젝트 개요

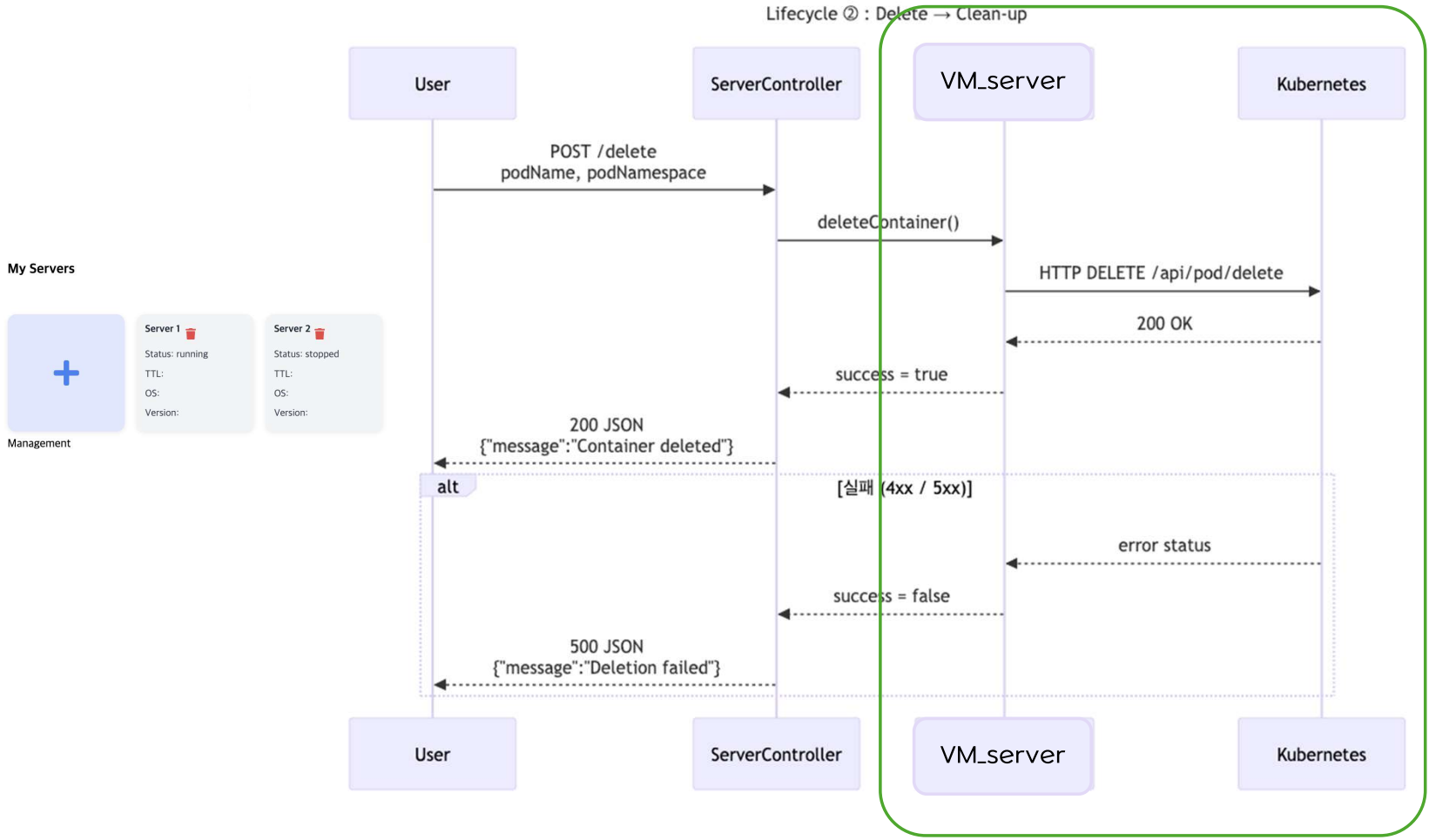
시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

📌 핵심 기능 2 : 웹 터미널 서버 접근

웹 터미널 테스트

🔍 사용자의 가상머신 CLI

```
[+] Connected to your pod
root@pod-5f164071:/# ls
bin  dev  home  media  opt  root  sbin  sys  usr
boot  etc  lib  mnt  proc  run  srv  var
root@pod-5f164071:/# ps aux | grep 'bash'
root      1  0.0  0.0  4116   308 pts/0    Ss+  Jun05   0:00 /bin/bash
root     557  0.0  0.0  4116   504 pts/2    Ss+  Jun06   0:00 bash
root    1483  0.0  0.0  4116   536 pts/1    Ss+  Jun07   0:00 bash
root    1511  0.0  0.0  4116   528 pts/3    Ss+  Jun07   0:00 bash
root    1530  0.0  0.0  4116   528 pts/4    Ss+  Jun07   0:00 bash
root    1631  0.0  0.0  4116   664 pts/5    Ss+  00:28   0:00 bash
root    2432  0.0  0.0  4116  3440 pts/6    Ss   06:50   0:00 bash
root    2494  0.0  0.0  2880  1284 pts/6    c+   06:50   0:00 grep --color=auto bash
root@pod-5f164071:/#
```

웹 터미널 영역



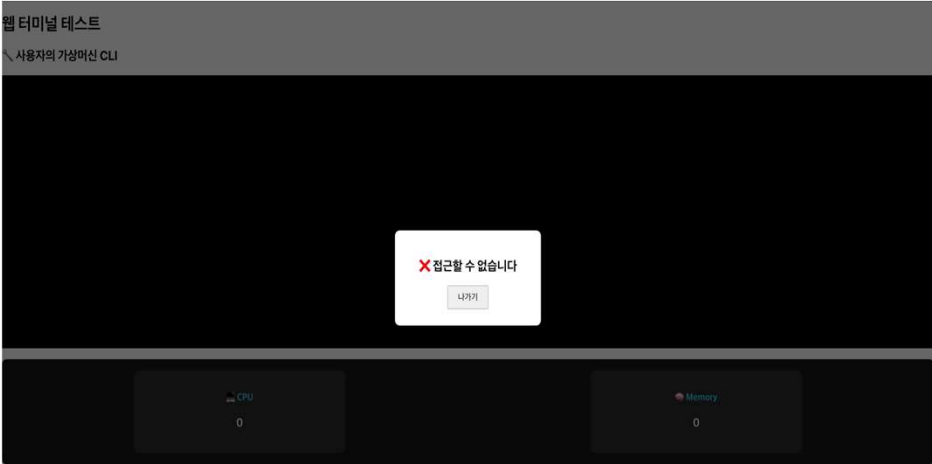
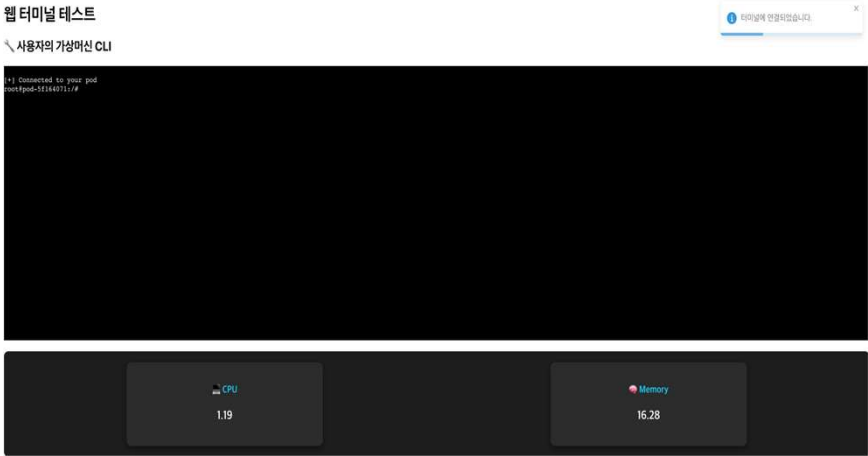
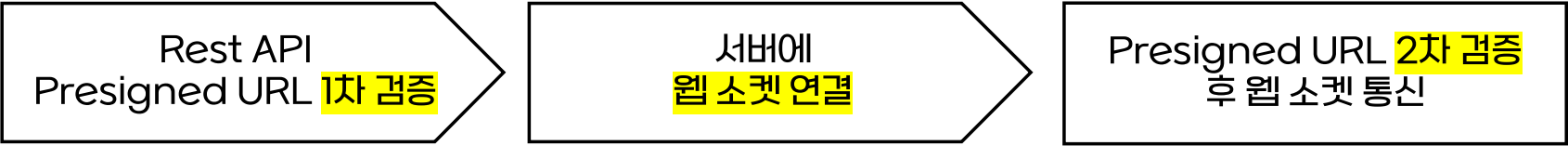
서버 상태 실시간 모니터링 영역

현재까지 개발된 웹 터미널 프로토타입

웹 터미널 상세 설명



웹 소켓 세션 연결 과정



현재까지 개발된 웹 터미널 프로토타입

🔗 📄 🌐 🔊

프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

웹 터미널 상세 설명

🌀 웹 소켓 메시지 타입 설정

```
RECV(JSON): TerminalComponent.js:80
▶ {type: 'WARNING', message: '장시간 활동이 감지되지 않아1분 뒤 터미널이 종료됩니다', currentTime: 1749377617349}


RECV(JSON): TerminalComponent.js:80
▶ {type: 'INPUT', message: 'exit\r\n\x1B[?2004l\r\nexit\r\n', currentTime: 1749377677367}


RECV(JSON): TerminalComponent.js:80
▶ {type: 'NOTICE', message: '연결이 종료되었습니다.\r\n', currentTime: 1749377677400}
```

NOTICE - 공지

WARNING - 경고

INPUT - 터미널 입력

 터미널에 연결되었습니다. ✕

 장시간 활동이 감지되지 않아1분 뒤 터미널이 종료됩니다 ✕

```
root@pod-5f164071:/# exit
exit

[+] Connection closed.
```

현재까지 개발된 웹 터미널 프로토타입



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

웹 터미널 상세 설명



터미널 Bash셸 관리

스케줄러를 이용한 미사용 셸 정리

20분 마다 스케줄러 작동

```
PodMetricsResponse(kind=PodMetrics, apiVers
PodMetricsResponse(kind=PodMetrics, apiVers
현재 시간 : 1749393754958, 스케줄링 시작
현재 시간 : 1749393814966, 스케줄링 시작
현재 시간 : 1749393874971, 스케줄링 시작
```

사용자의 마지막 입력으로부터
5분 지났을 때 : **warning** 전송
10분 지났을 때 : **exit** 명령어 전송

터미널 종료 시 exit 명령어 전송

세션 종료 시 강제 창 닫기

```
root@pod-5f164071:/# exit
exit
```

[+] Connection closed.

```
RECV(JSON):
▶ {type: 'INPUT', message: 'exit\r\n\x1B[?2004l\r\nexit\r\n'}
```

이미 접속중인 사람 추가 접속 막기

개발 예정

현재까지 개발된 웹 터미널 프로토타입



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

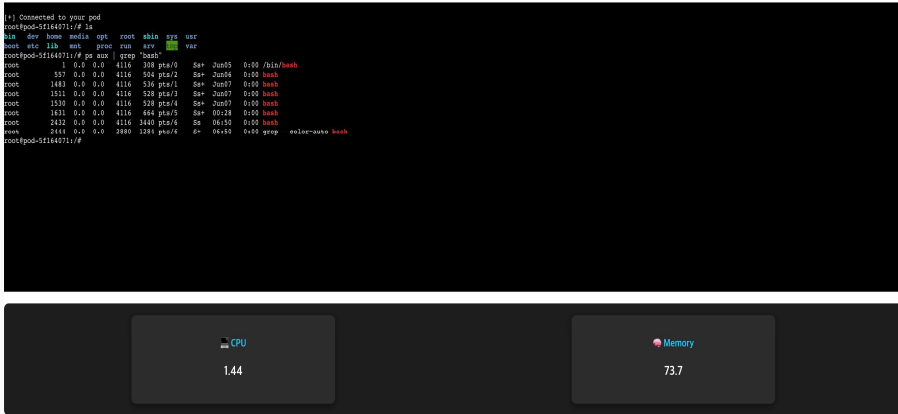
웹 터미널 상세 설명



터미널 실시간 모니터링

웹 터미널 테스트

사용자의 가상머신 CLI



seo ~ kubectl top pod -n default		
NAME	CPU(cores)	MEMORY(bytes)
alermanager-kube-prometheus-stack-alertmanager-0	3m	44Mi
kube-prometheus-stack-grafana-67bb9f4bc6-bkkgf	26m	419Mi
kube-prometheus-stack-kube-state-metrics-77678594d6-tl84n	3m	27Mi
kube-prometheus-stack-operator-f99678f48-f4k5n	1m	40Mi
pod-1e933f36	0m	0Mi
pod-29f239f1	0m	0Mi
pod-4aed852b	0m	0Mi
pod-5f164071	0m	20Mi
pod-673e4c1f	0m	0Mi
pod-8bdfdf3c	0m	0Mi
pod-c1212930	0m	0Mi
prometheus-kube-prometheus-stack-prometheus-0	26m	248Mi
seo ~		

Metrics-server를 통해 pod의 실시간 사용량 수집

Rest API를 통하여 5초에 한 번씩 cpu/memory 사용량 업데이트

현재까지 개발된 웹 터미널 프로토타입



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

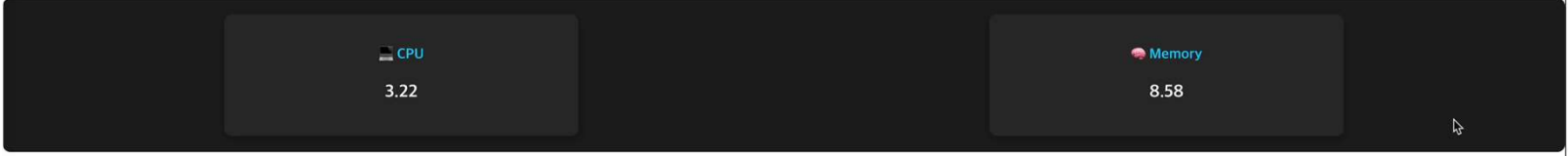
웹 터미널 데모



웹 터미널 테스트

🔧 사용자의 가상머신 CLI

```
[+] Connected to your pod
root@pod-5f164071:/#
```





프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

📌 핵심 알고리즘

Presigned URL

: 클라우드 스토리지나 서버 리소스에 일시적으로 접근할 수 있도록 만들어진 서명된 URL

- 매우 짧은 유효 시간
- URL에 서명(signature) 부여
- 해당 URL 보유 시, 추가 인증 없이 일정 시간 동안 접근

Backend Server 발급 / VM Server 검증

JWT

(JSON Web Token) : 인증 정보를 JSON 형태로 담고, 디지털 서명(Signature)을 포함한 토큰 기반 인증 방식

- HTTP의 무연결성(stateless)을 극복하기 위해, 사용자 검증하는 서명된 JSON 문자열
- 로그인 성공 시 JWT 발급
- localStorage에 jwt 저장, 이후 인증에 사용

Backend Server 사용자 인증 중계 / VM Server 인증 없이 서버 관련 작업



프로로그

프로젝트 개요

시스템 구조

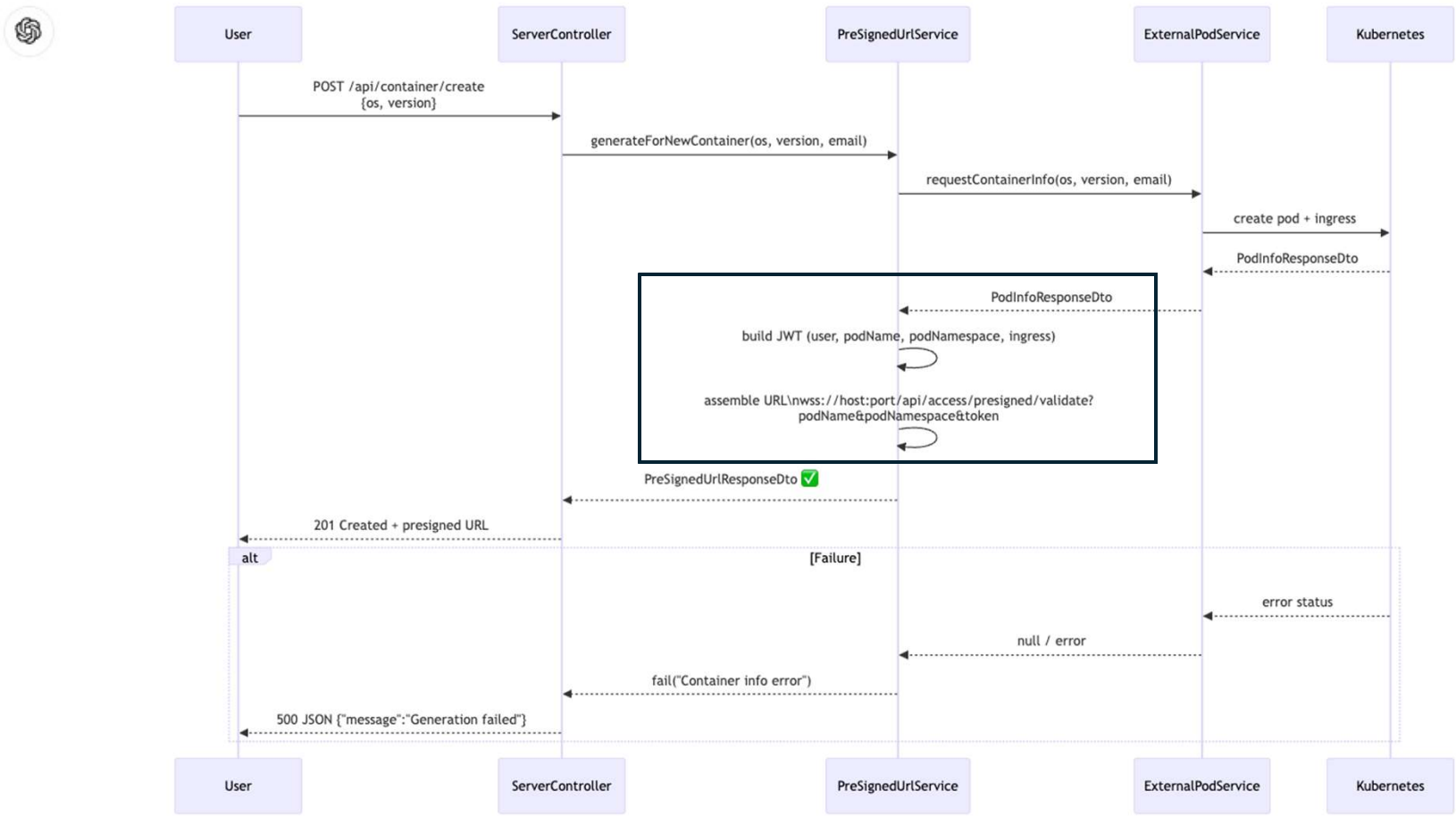
핵심 기능

주요 알고리즘

일정 및 계획

에필로그

Presigned URL 생성 알고리즘





프롤로그

프로젝트 개요

시스템 구조

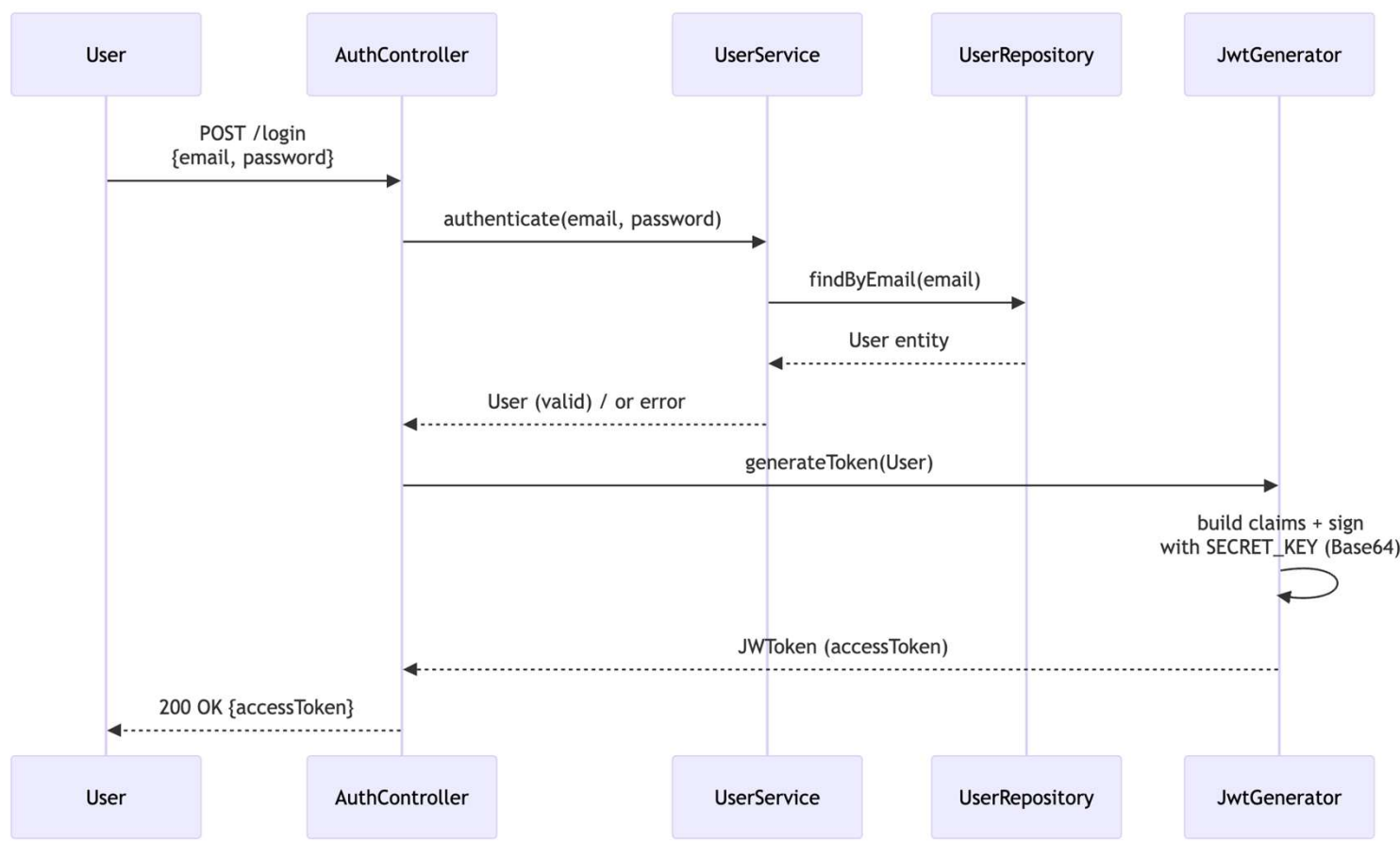
핵심 기능

주요 알고리즘

일정 및 계획

에필로그

JWT 생성 알고리즘



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그



1학기 일정

월

세부 계획

1월

아이디어 회의, 기획 / 스프링, 리액트, 도커 등 프로젝트를 위한 사전 지식 학습

2월

웹 사이트 프로토타입 제작을 통한 전체적인 웹 사이트의 기능들 기획

3월

4월

5월

6월

기능 설계 및 구현

기능 설계 및 구현 / 테스트

확장 가능하고, 재사용가능한 구조의 코드 작성이 목표

프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그



2학기 계획

월	세부 계획
7월	기획한 모든 서비스 구현
8월	서비스 배포 및 서버 관리
9월	취약점 분석 및 솔루션 탐색
10월	보안 솔루션 적용한 서비스 배포 / 캡스톤디자인2 마무리
11월	캡스톤디자인2 최종 발표
12월	-

외부 공격으로부터 안전한 보안 서비스 제공 목표

🔍 📄 🌐 🔊



프로로그

프로젝트 개요

시스템 구조

핵심 기능

주요 알고리즘

일정 및 계획

에필로그

⚠️ **Thank you** ⚠️

발표가 종료되었습니다. 감사합니다.

ANY QUESTION?

태훈 업고 투어(TCAR) – 한하영, 서민재, 김주령

