

SLASH 24

클러스터 운영부터 서비스 라이프 사이클 관리까지, 데브옵스 업무 효율화

양석준
DevOps Engineer

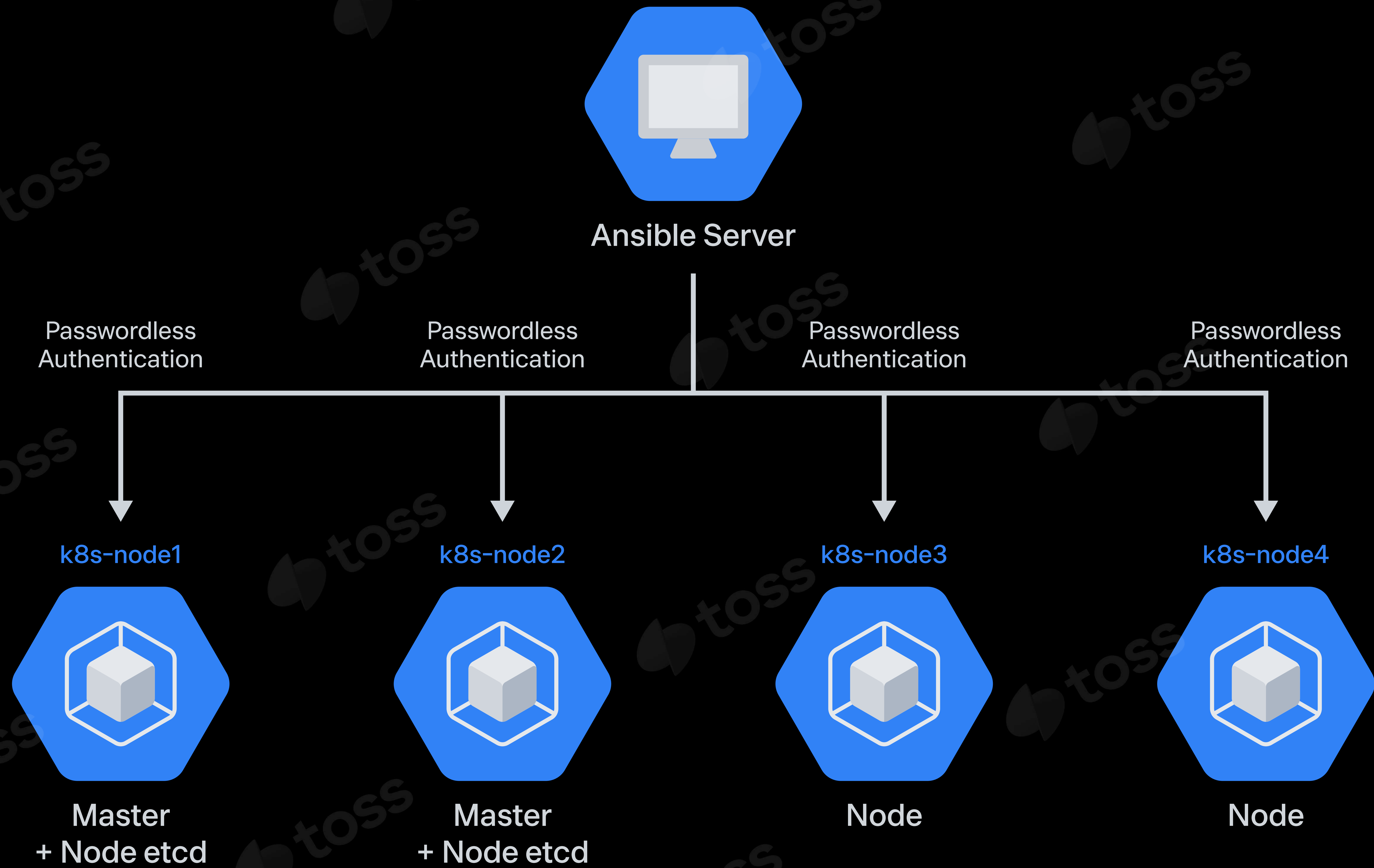
본 발표자료의 저작권은 연사에 있으며, 저작권자의 사전 서면 동의 없이 자료의 일부 또는 전부를 이용하거나 배포할 수 없습니다.

또한 해당 자료를 복제하여 SLASH 행사 홈페이지를 제외한 온라인상에 게재하는 행위는 연사가 동의한 저작권 및 배포전송권에 위배됩니다.

토스가 다루는 모든 개인정보는 고객에게 동의를 받은 후에 처리되고 있으며, 접근 권한이 분리되어 있습니다.
개발자는 모든 데이터가 아닌 담당 영역에 한하여 접근·이용할 수 있습니다.



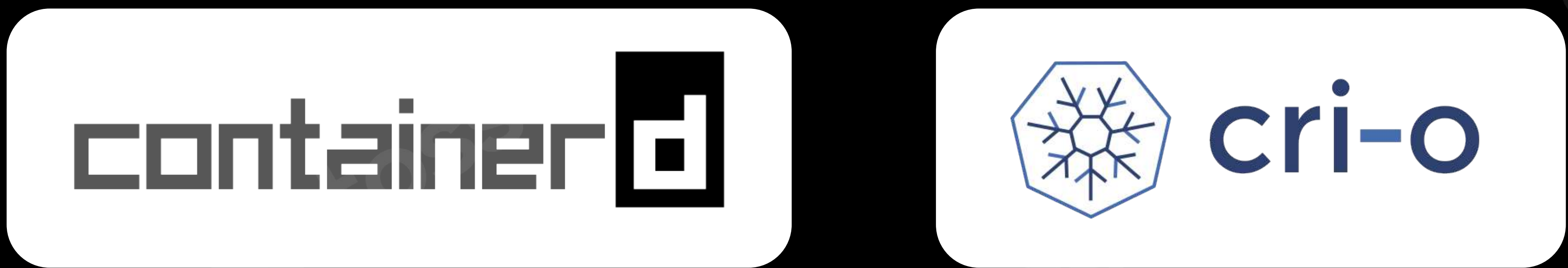
클러스터 단위에서의 개선 사례



Infrastructure Layer



Container Layer



Kubespary로 클러스터 노드를 관리하다 보니
이런 문제들이 있었습니다

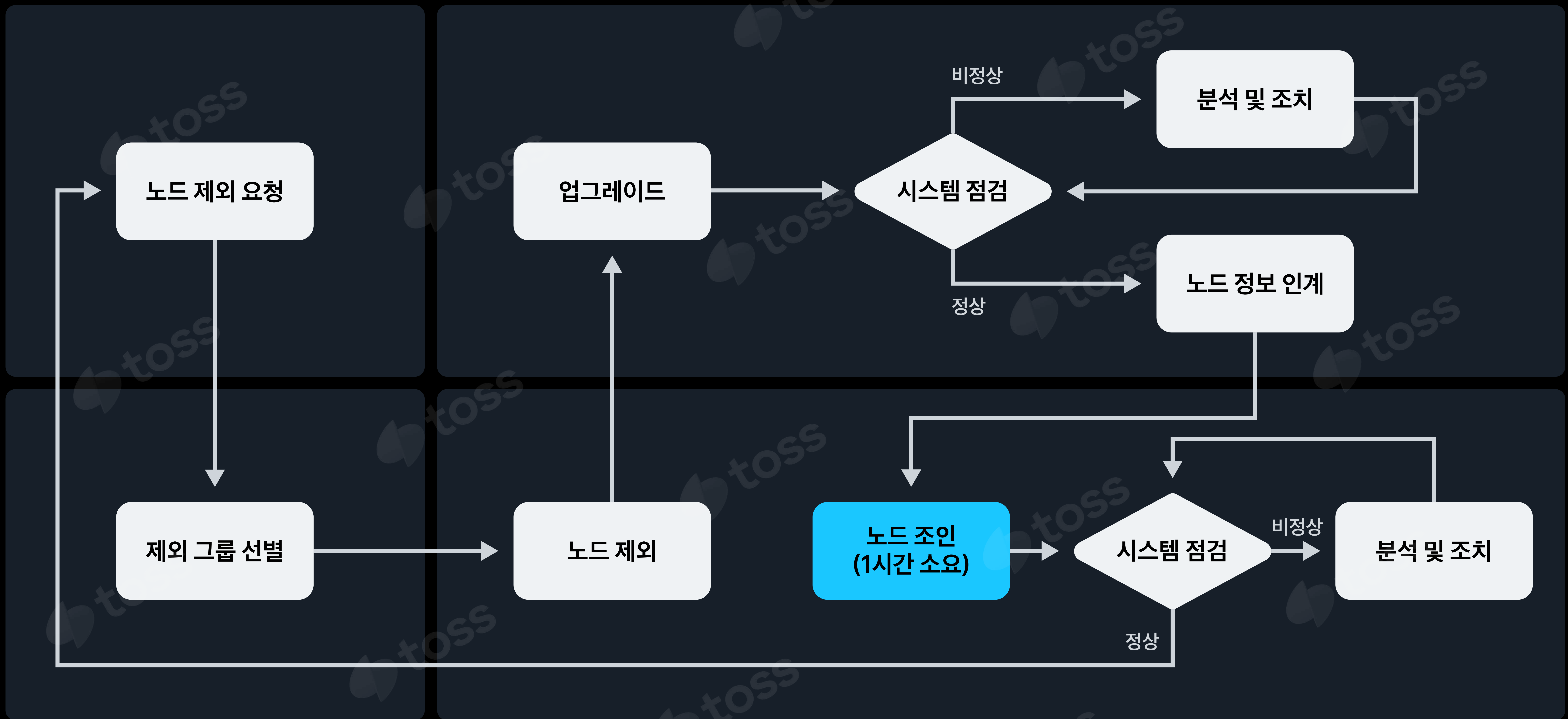

```
roles:
  - { role: kubepay-defaults }
  - { role: bastion-ssh-config, tags: ["localhost", "bastion"] }
roles:
  - { role: kubepay-defaults }
  - { role: bootstrap-os, tags: bootstrap-os }
roles:
  - { role: kubepay-defaults }
  - role: etcd
roles:
  - { role: kubepay-defaults, when: "not skip_downloads and download_run_once and not download_localhost" }
  - { role: kubernetes/preinstall, tags: preinstall, when: "not skip_downloads and download_run_once and not download_localhost" }
  - { role: download, tags: download, when: "not skip_downloads and download_run_once and not download_localhost" }
roles:
  - { role: kubepay-defaults }
  - { role: kubernetes/preinstall, tags: preinstall }
  - { role: container-engine, tags: "container-engine", when: deploy_container_engine }
  - { role: download, tags: download, when: "not skip_downloads" }
  - role: etcd
roles:
  - { role: kubepay-defaults }
  - { role: kubernetes/node, tags: node }
roles:
  - { role: kubepay-defaults }
roles:
  - { role: kubepay-defaults }
  - { role: kubernetes/kubeadm, tags: kubeadm }
  - { role: kubernetes/node-label, tags: node-label }
  - { role: network_plugin, tags: network }
roles:
  - { role: kubepay-defaults }
  - { role: kubernetes/preinstall, when: "dns_mode != 'none' and resolvconf_mode == 'host_resolvconf'", tags: resolvconf, dns_late: true }
```

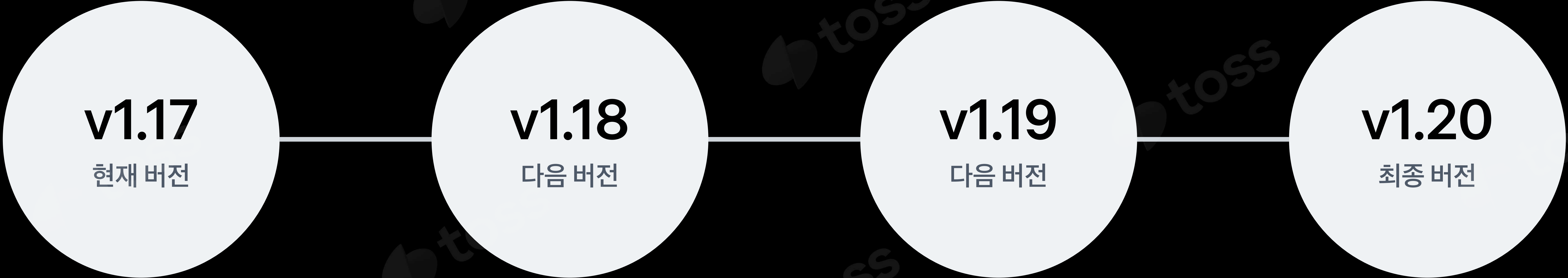
사전 작업

본 작업

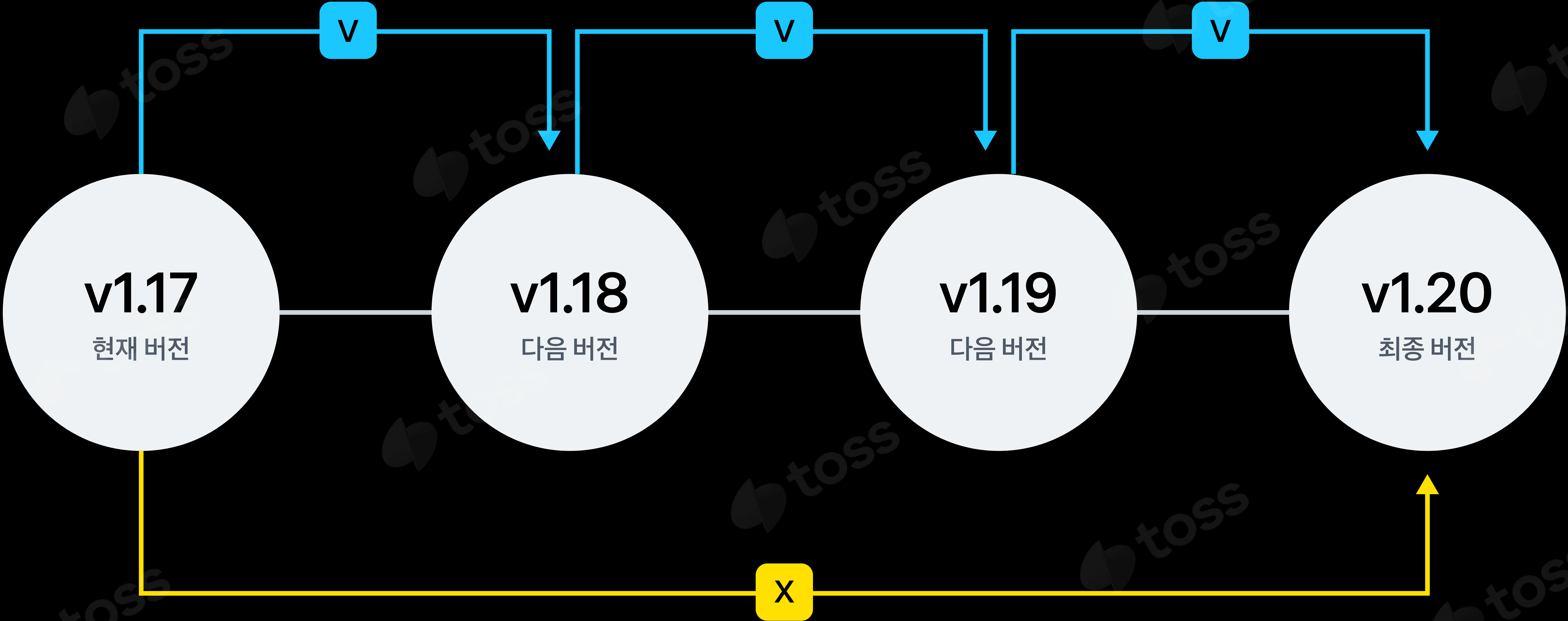
Infra

DevOps









Using Kubespray

30m

per node group

노드 조인

30m

per node group

운영체제/커널 업그레이드

2hr

per one version

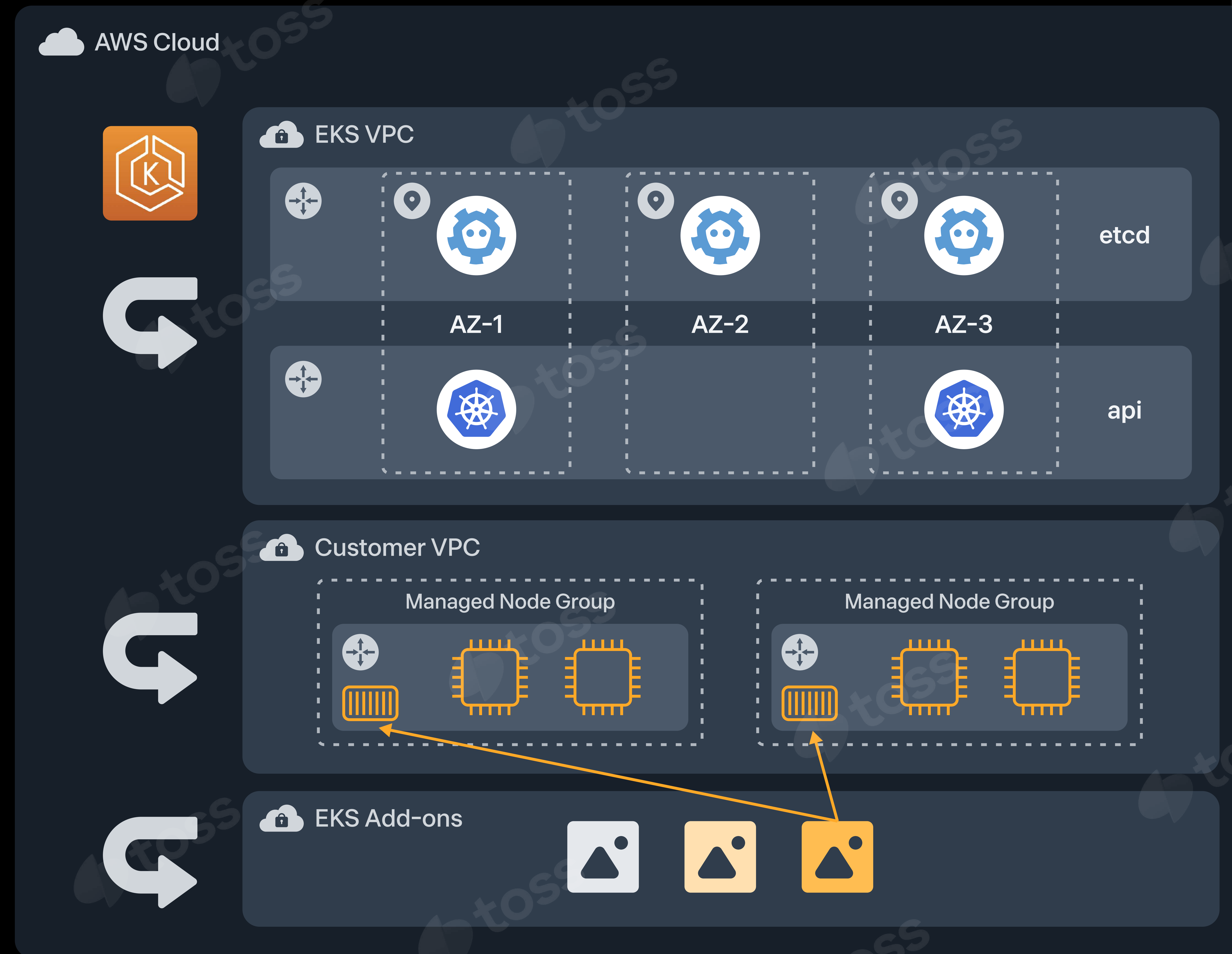
Kubernetes 업그레이드

**짧게는 30분, 길게는 6시간에서 12시간 정도 소요되는 작업들을
어떻게 하면 줄일 수 있을까요?**

EKS Managed
Control Plane

EKS Managed
Data Plane

EKS Add-ons

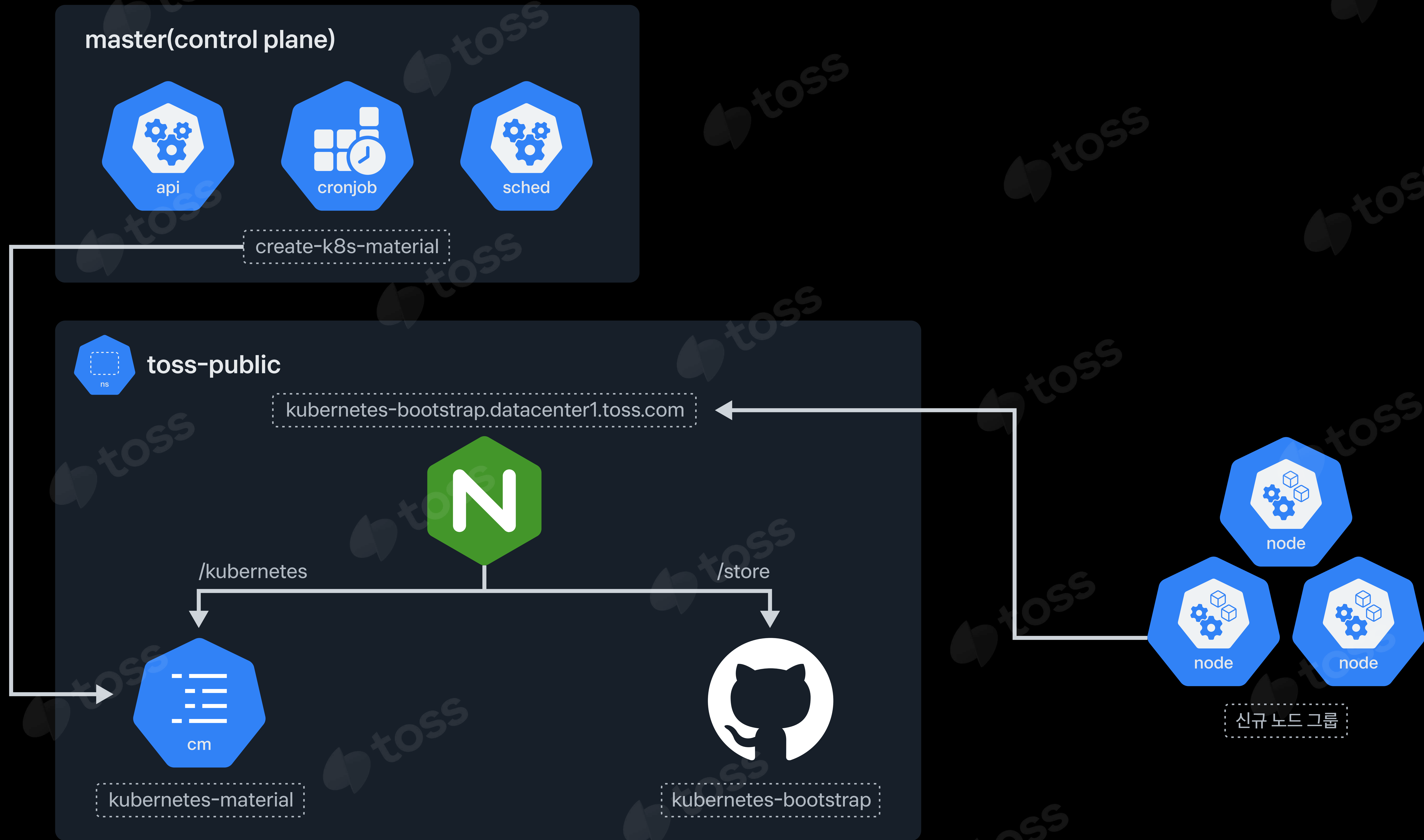


고려사항

1. 데브옵스 엔지니어의 공수 없이, 노드 조인 자동화
2. 디버깅이 간편하며, 복잡하지 않은 아키텍처
3. GitOps가 가능한 구성

고려사항

1. 데브옵스 엔지니어의 공수 없이, 노드 조인 자동화
2. 디버깅이 간편하며, 복잡하지 않은 아키텍처
3. GitOps가 가능한 구성



kube-{ClusterA}-{DataCenter}-worker01
= {DataCenter}.{ClusterA}.toss.com

```
- v1.21.6
- etc
  - NetworkManager/conf.d
  - calico/certs
  - cni/net.d
  - containerd
  - kubernetes
  - nginx
  - systemd/system

- v1.25.6
- etc
  - NetworkManager/conf.d
  - cni/net.d
  - containerd
  - kubernetes
  - nginx
  - systemd/system
```

파일 시스템 구조

```
common:
  cluster: 1.25.6
  runcVersion: 1.x.x
  criVersion: 1.x.x
  cniPluginVersion: 1.x.x
  cudaVersion: "12.2.1_535.86.10_linux"
  clusterDNS: "8.8.8.8"

master:
  k8s-master-dc2:
    address: x.x.x.x

worker:
  kube-serviceA-dc1-gpu-worker01:
    gpu: true
  previous-version-node:
    version:
      kubernetes: 1.21.6
```

클러스터 설정



노드A 10:10 AM

[A클러스터] 클러스터에 서비스 노드로 조인되었습니다. kubelet: v1.25.6

노드 조인 성공



노드B 10:10 AM

[A클러스터] 클러스터에 조인에 실패하였습니다.

[preflight] Running pre-flight checks

error execution phase preflight: [preflight] Some fatal errors occurred:

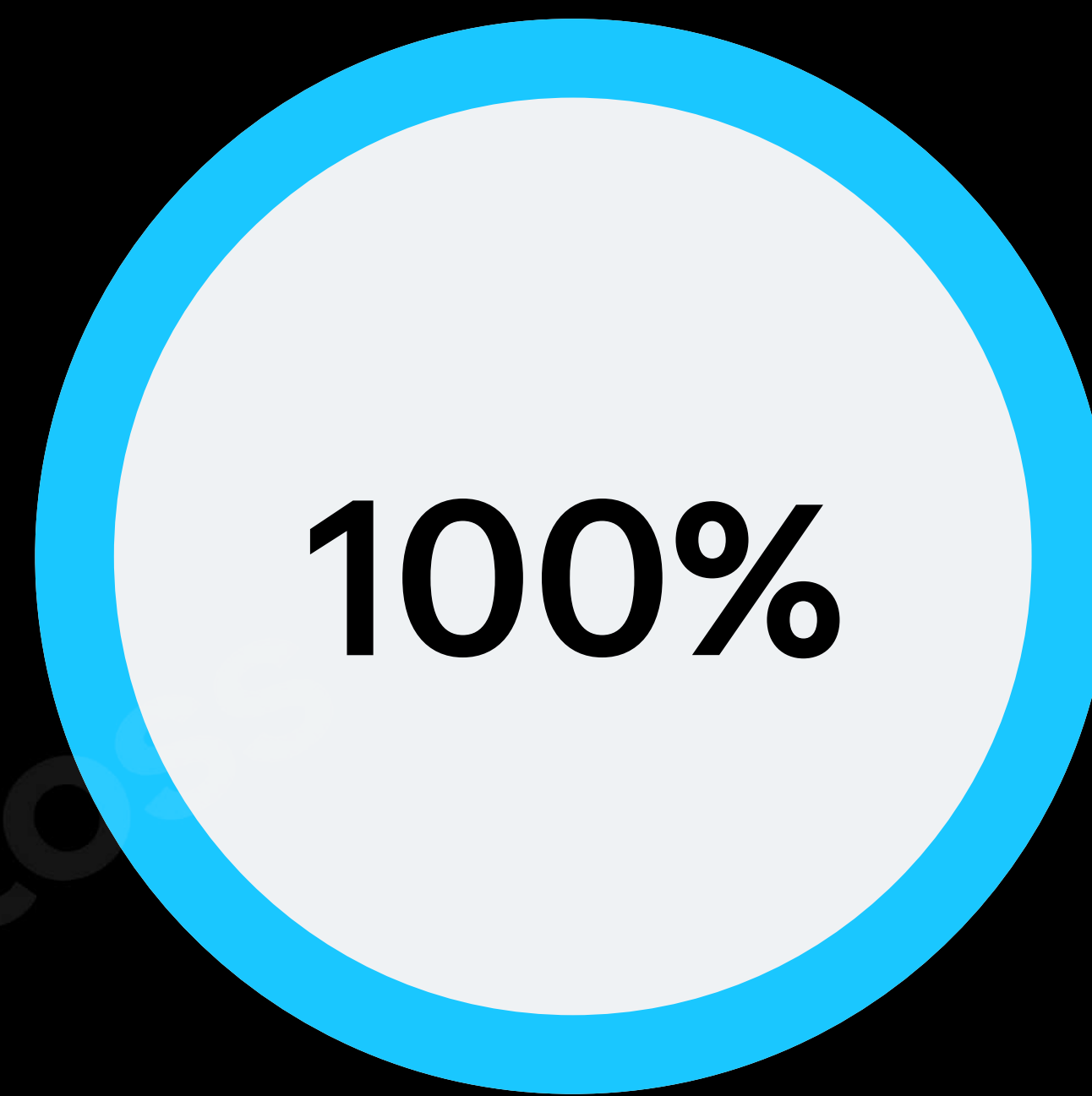
[ERROR CRI]: container runtime is not running: output: time="2024-04-02T00:14:27+09:00" level=fatal msg="unable to determine runtime API version: rpc error: code = Unavailable desc = connection error: desc = \\"transport: Error while dialing dial unix /var/run/containerd/containerd.sock: connect: no such file or directory\\"""
, error: exit status 1

[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`

To see the stack trace of this error execute with `--v=5` or higher

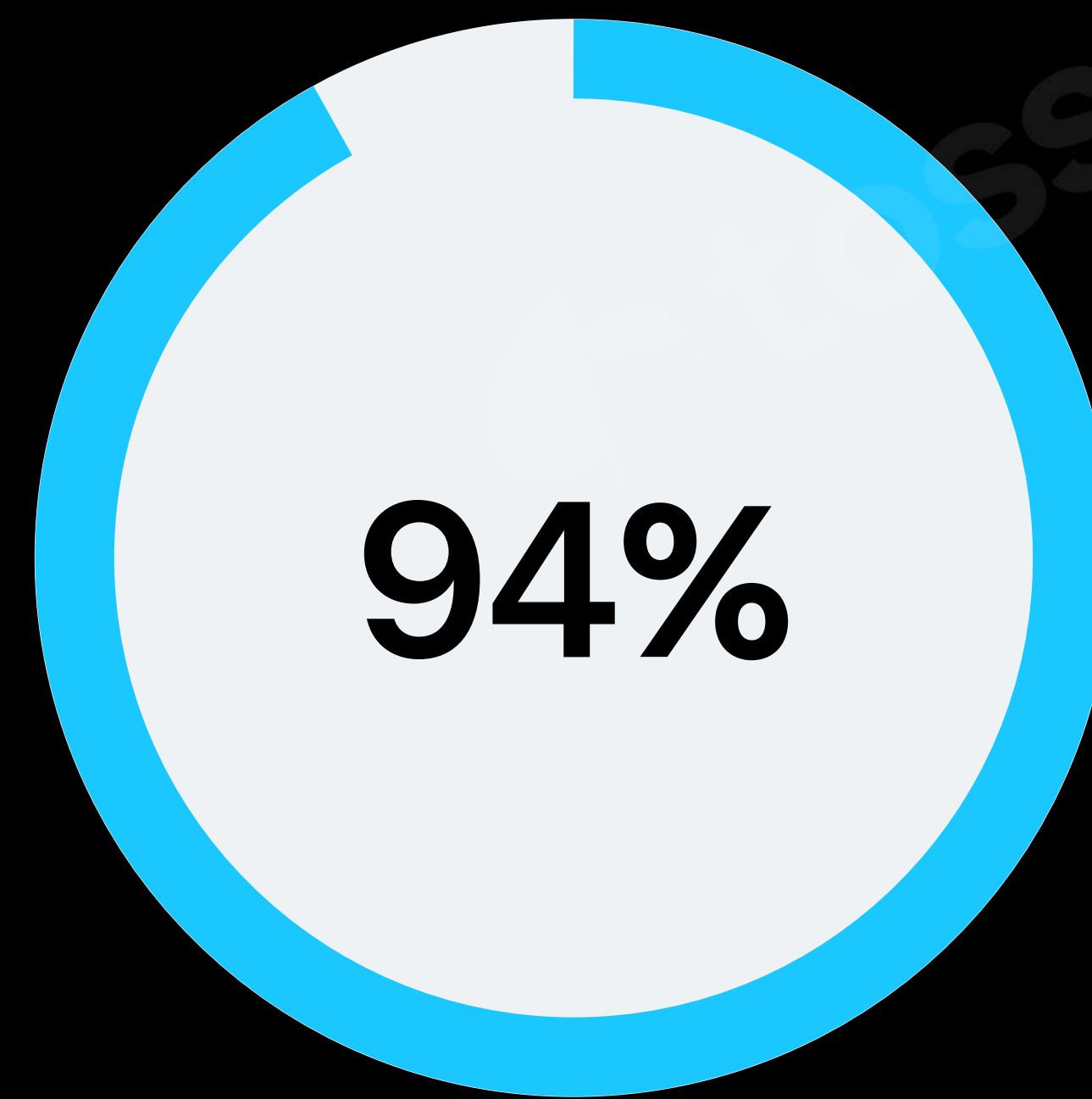
노드 조인 실패

노드 조인



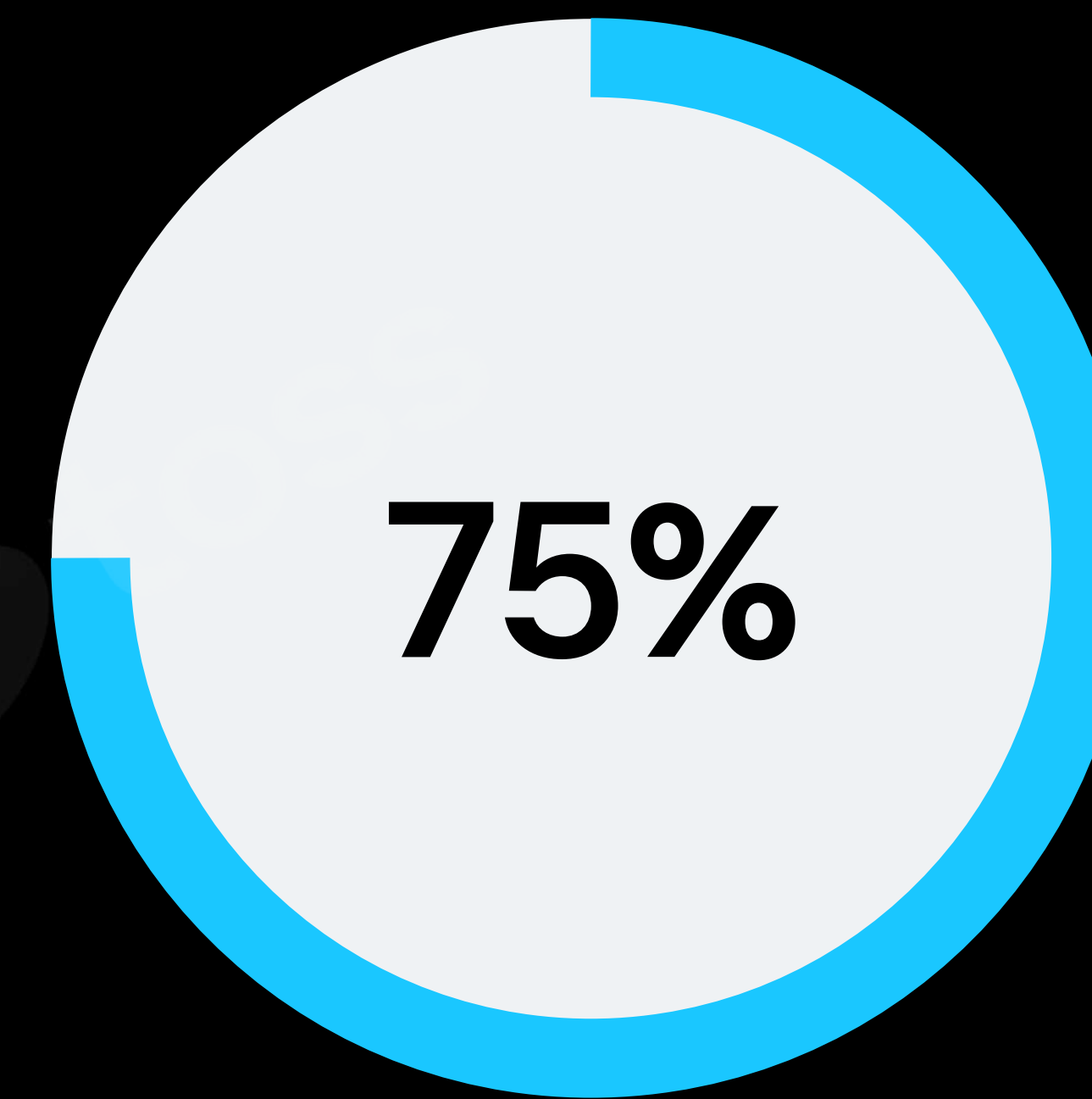
30m → 0m
per node group

운영체제/커널 업그레이드



30m → 2m
per node group

Kubernetes 업그레이드



2hr → 30m
per one version

서비스 단위에서의 개선 사례

신규 서비스가 추가되면

개발/운영 클러스터에 많은 파이프라인 생성 필요

통신 인터페이스에 따라 다른 배포 방식

서비스에서 사용하는 언어에 따라 다른 설정

vivasystem

파이프라인 바로가기

홈

A

B

C

D

F

관리

G

H

I

J

전체

> 서비스A success

> 서비스B success

> 서비스C success

> 서비스D success

> 서비스E success

> 서비스F success

> 서비스G success

> 서비스H cancelled

> 서비스I success

> 서비스J success

> 서비스K success

> 서비스L error

> 서비스N success

> 서비스M success

> 서비스O success

> 서비스P success

> adx-external success

파이프라인 만들기

* 서비스 이름:

* 레포지토리 명:

git@git.example.com:example/

.git

* 브랜치:

develop

프론트엔드 서비스인가요?:

빌드

빌드 환경변수:

빌드 시 사용되는 환경변수입니다.

+ Add field

Gradle 프로젝트인가요?:

ARM64 지원:

Gradle 태스크:

clean build

Gradle 옵션:

-Duser.timezone=Asia/Seoul

Gradle 실행 디렉토리:

.

Java 버전:

미선택시 1.8로 반영됩니다.

Dockerfile 위치:

Dockerfile

docker 빌드 인자:

--docker-args 로 들어가는 키 / 값

+ Add field

* 서비스 이름:

* 레포지토리 명:

git@git.example.com:example/

.git

* 브랜치:

develop

프론트엔드 서비스인가요?:

☐

빌드

빌드 환경변수:

빌드 시 사용되는 환경변수입니다.

+ Add field

Gradle 프로젝트인가요?:

☒

ARM64 지원 ⓘ:

☒

Gradle 테스트:

clean build

Gradle 옵션:

-Duser.timezone=Asia/Seoul

Gradle 실행 디렉토리:

.

Java 버전:

미선택시 1.8로 반영됩니다.

▼

UI



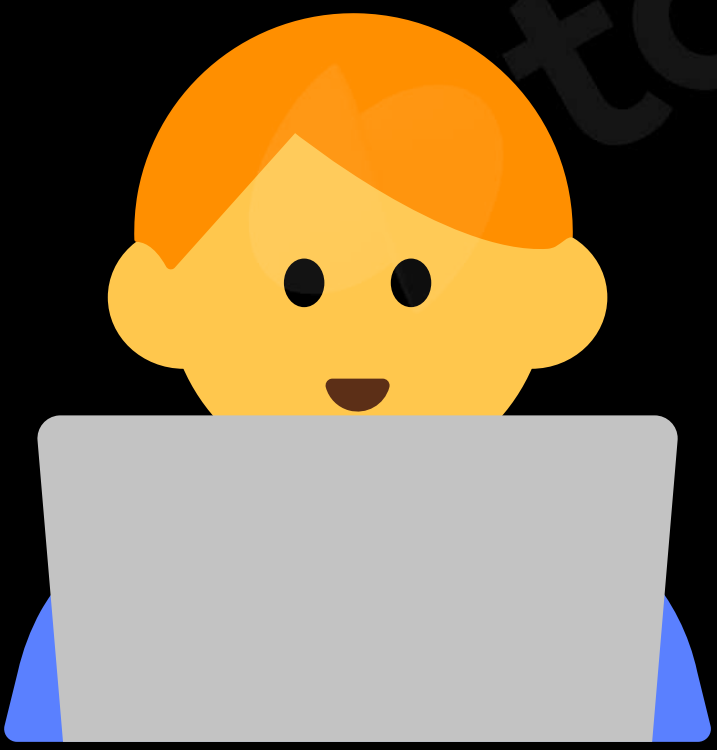
```
name: serviceA
repository:
  name: serviceA
  branch: develop
pipeline:
  type: service
  platform: linux/arm64
build:
  environments:
    JAVA_HOME: /opt/jdk-21
  batch:
    enabled: false
  docker:
    dockerfile: Dockerfile
resources:
  alpha:
    instance: 1
    cpu: '0.05'
    memory: 1024MB
```

Pipeline as a code



서비스 개발자

배치 태그 갱신 요청



데브옵스 엔지니어

태그 갱신

A 서비스

1버전

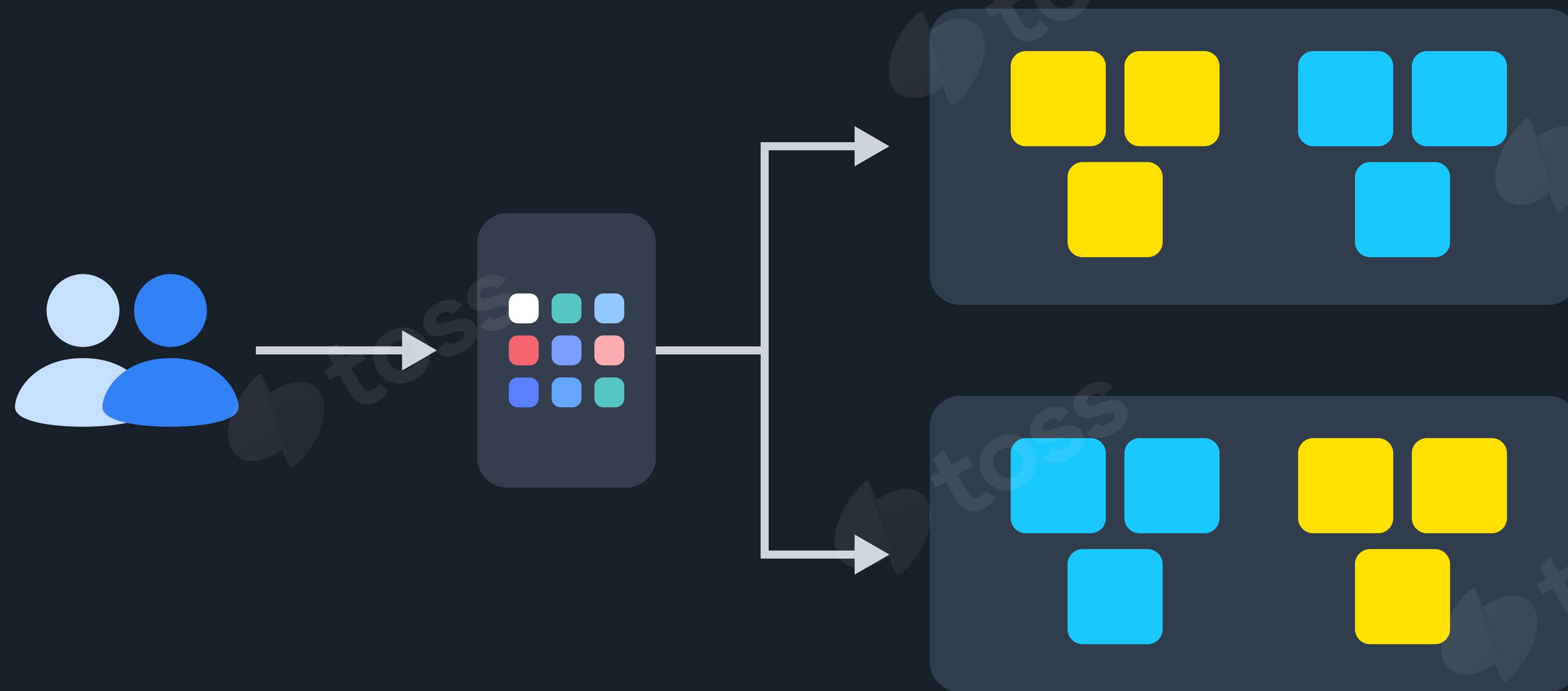
2버전

3버전

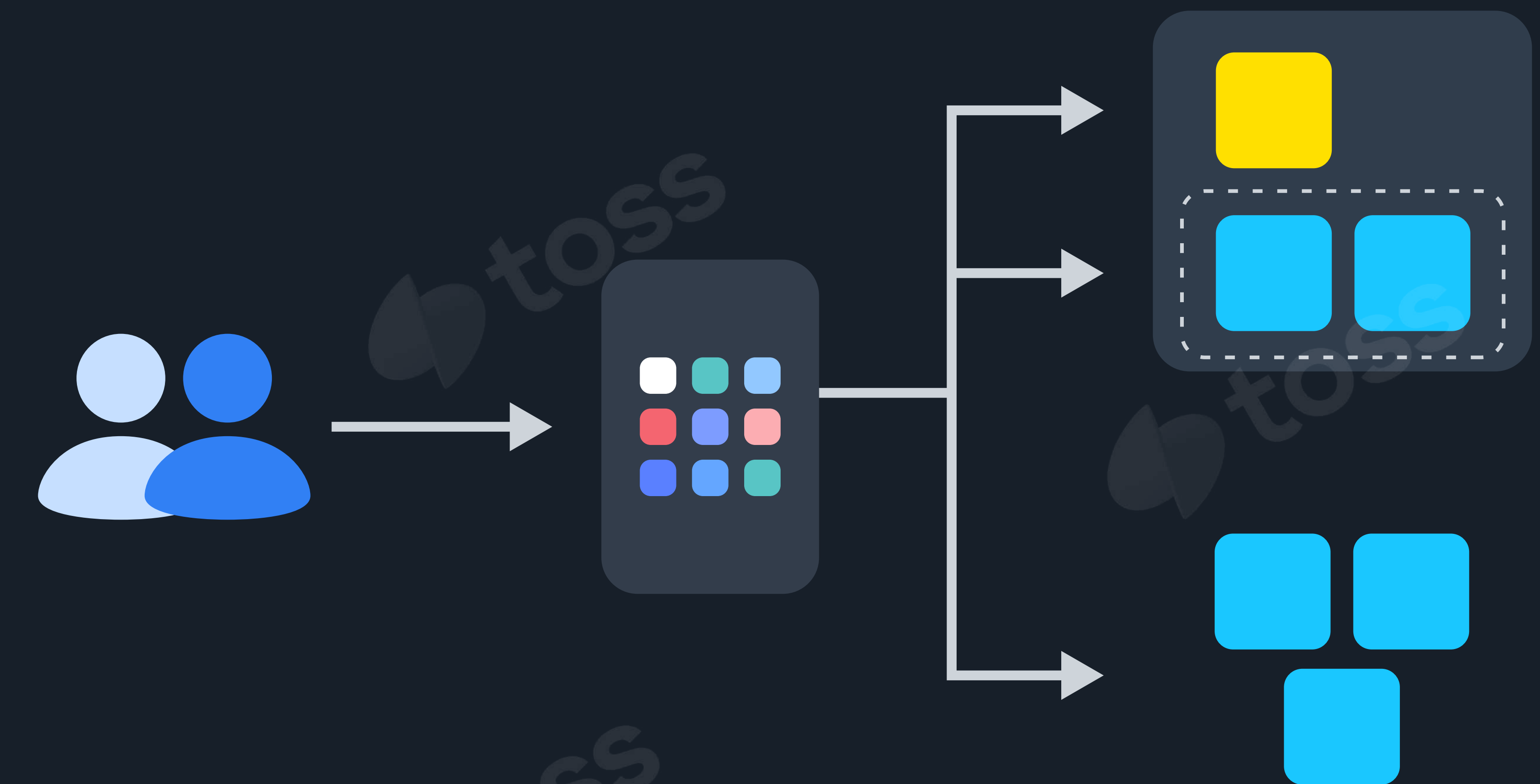
서비스A 이미지 태그 목록

<div>2182-develop-b05419e...</div> <div>latest 갱신명령어 복사</div>	<div>2181-develop-10520f4...</div> <div>latest 갱신명령어 복사</div>	<div>2180-develop-025698...</div> <div>latest 갱신명령어 복사</div>	<div>2179-develop-fa74df0...</div> <div>latest 갱신명령어 복사</div>	<div>2178-develop-e637990...</div> <div>latest 갱신명령어 복사</div>	<div>2177-develop-d7792e8...</div> <div>latest 갱신명령어 복사</div>
<div>2176-develop-f17fd19...</div> <div>latest 갱신명령어 복사</div>	<div>2175-develop-936176...</div> <div>latest 갱신명령어 복사</div>	<div>2174-develop-418c597...</div> <div>latest 갱신명령어 복사</div>	<div>2173-develop-241e744...</div> <div>latest 갱신명령어 복사</div>	<div>2172-develop-7d4a275...</div> <div>latest 갱신명령어 복사</div>	<div>2171-develop-fc5b02e...</div> <div>latest 갱신명령어 복사</div>
<div>2170-develop-5e58d9c...</div> <div>latest 갱신명령어 복사</div>	<div>2169-develop-5979d0c...</div> <div>latest 갱신명령어 복사</div>	<div>2168-develop-9891fa7...</div> <div>latest 갱신명령어 복사</div>	<div>2167-develop-19b0e32...</div> <div>latest 갱신명령어 복사</div>	<div>2166-develop-48a169a...</div> <div>latest 갱신명령어 복사</div>	<div>2165-develop-c09b4cd...</div> <div>latest 갱신명령어 복사</div>
<div>private-csjang-plf-8877...</div> <div>latest 갱신명령어 복사</div>	<div>private-csjang-plf-8877...</div> <div>latest 갱신명령어 복사</div>	<div>private-csjang-plf-8877...</div> <div>latest 갱신명령어 복사</div>	<div>private-las-plf-8930-1-...</div> <div>latest 갱신명령어 복사</div>	<div>private-las-plf-8930-2-...</div> <div>latest 갱신명령어 복사</div>	

Blue-Green Deployment



Canary Deployment





vivasystem 10:10 AM

카나리와 블루그린간의 랙이 발생하고 있습니다.

서비스A 랙 12일 @김토스A 배포책임자: @김토스B, @김토스C, @김토스D

블루그린 페이지이동 블루그린 트리거

서비스B 랙 12일 @김토스B 배포책임자: @김토스E, @김토스F, @김토스G

블루그린 페이지이동 블루그린 트리거

서비스C 랙 12일 @김토스C 배포책임자: @김토스H, @김토스I

블루그린 페이지이동 블루그린 트리거

서비스D 랙 12일 @김토스D 배포책임자: @김토스J, @김토스K, @김토스L

블루그린 페이지이동 블루그린 트리거

서비스E 랙 12일 @김토스E 배포책임자: @김토스M, @김토스N, @김토스O

블루그린 페이지이동 블루그린 트리거

서비스F 랙 12일 @김토스F 배포책임자: @김토스P

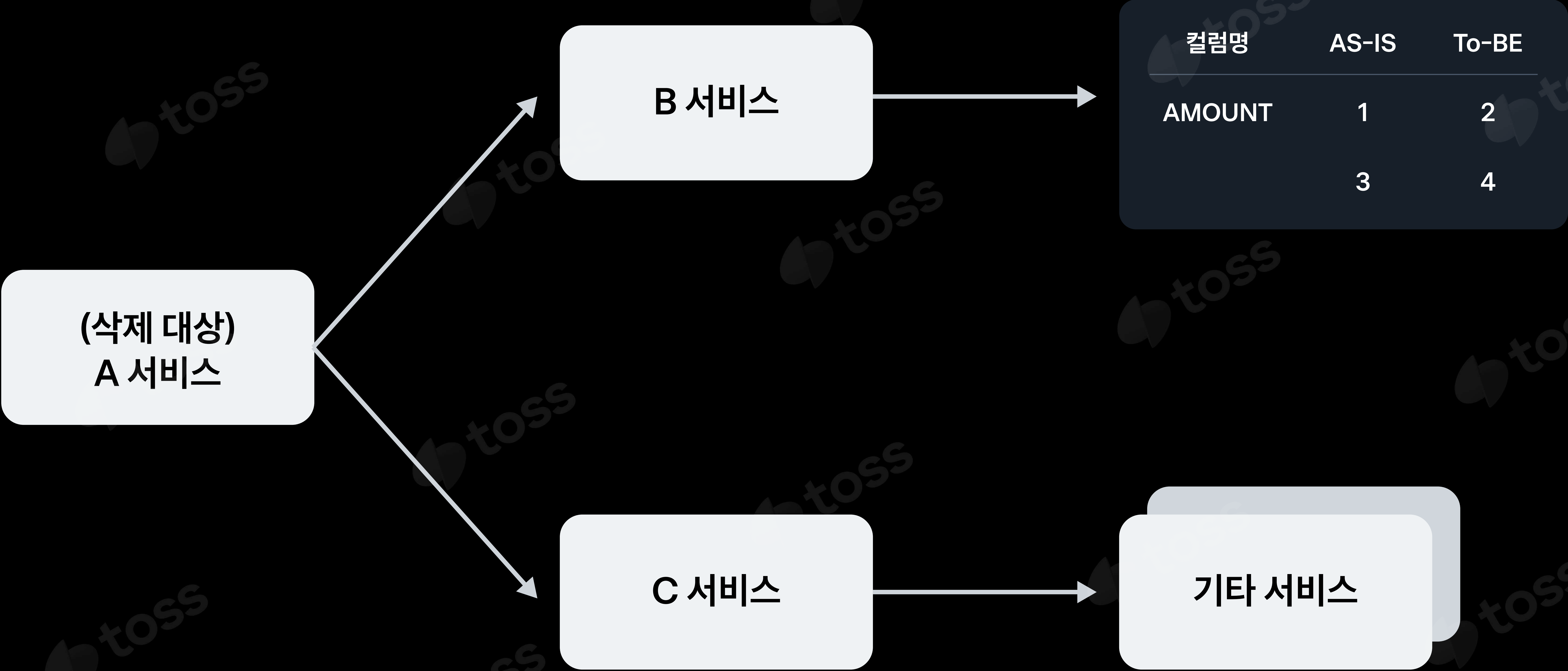
블루그린 페이지이동 블루그린 트리거

서비스G 랙 12일 @김토스G 배포책임자: @김토스Q, @김토스R, @김토스S

블루그린 페이지이동 블루그린 트리거

서비스H 랙 12일 @김토스H 배포책임자: @김토스T, @김토스U

블루그린 페이지이동 블루그린 트리거



파이프라인 삭제

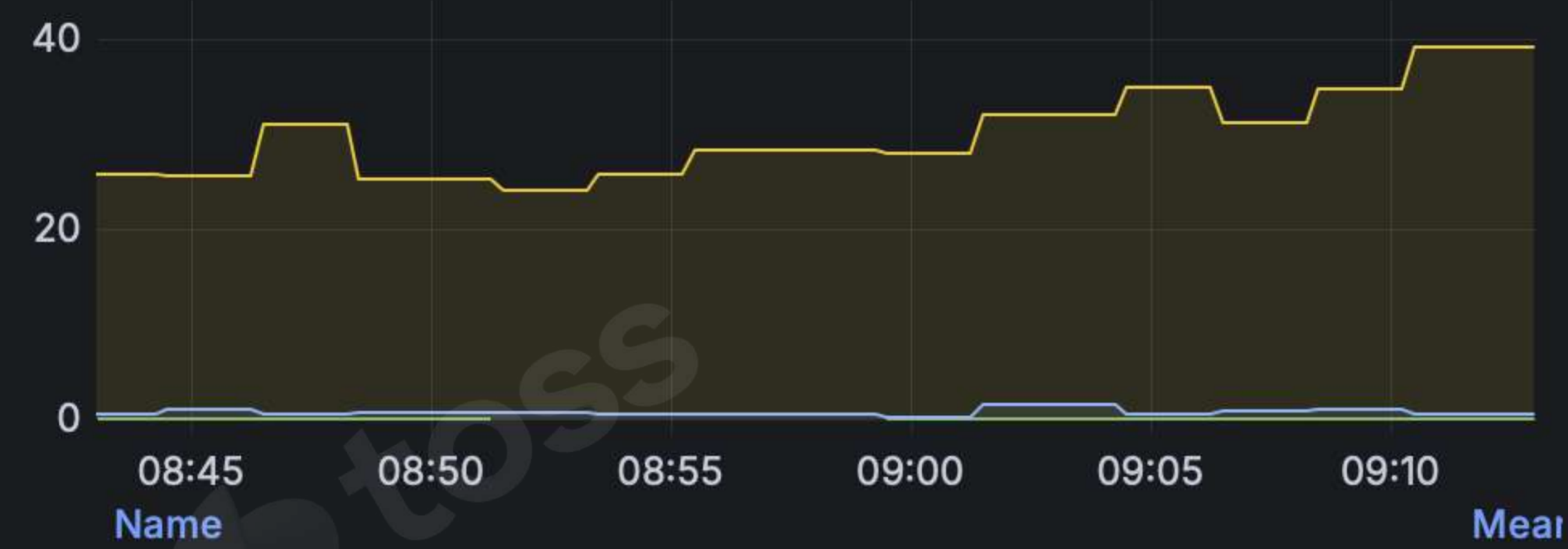
연결된 배치가 있다면 먼저 삭제해야 합니다.

연결된 배치

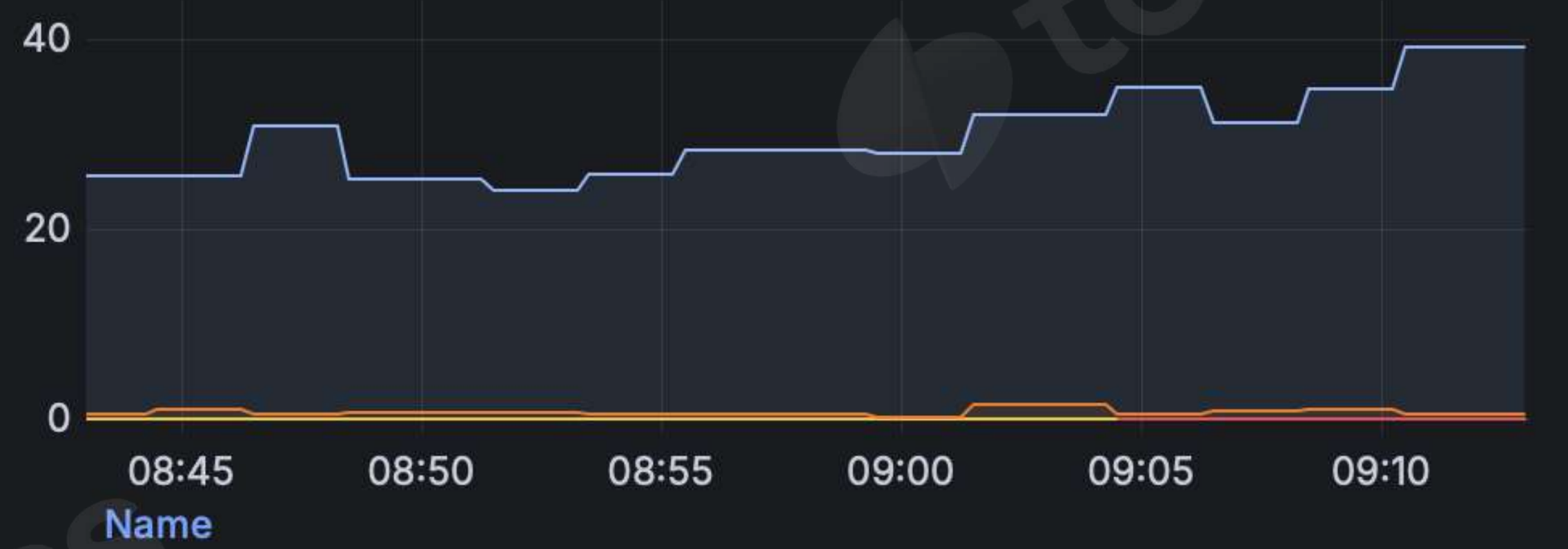
크론잡 리소스 이름	
	배치 A
	배치 B
	배치 C
	배치 D
	배치 E
	배치 F
	배치 G
	배치 I
	배치 J

파이프라인 스펙 삭제

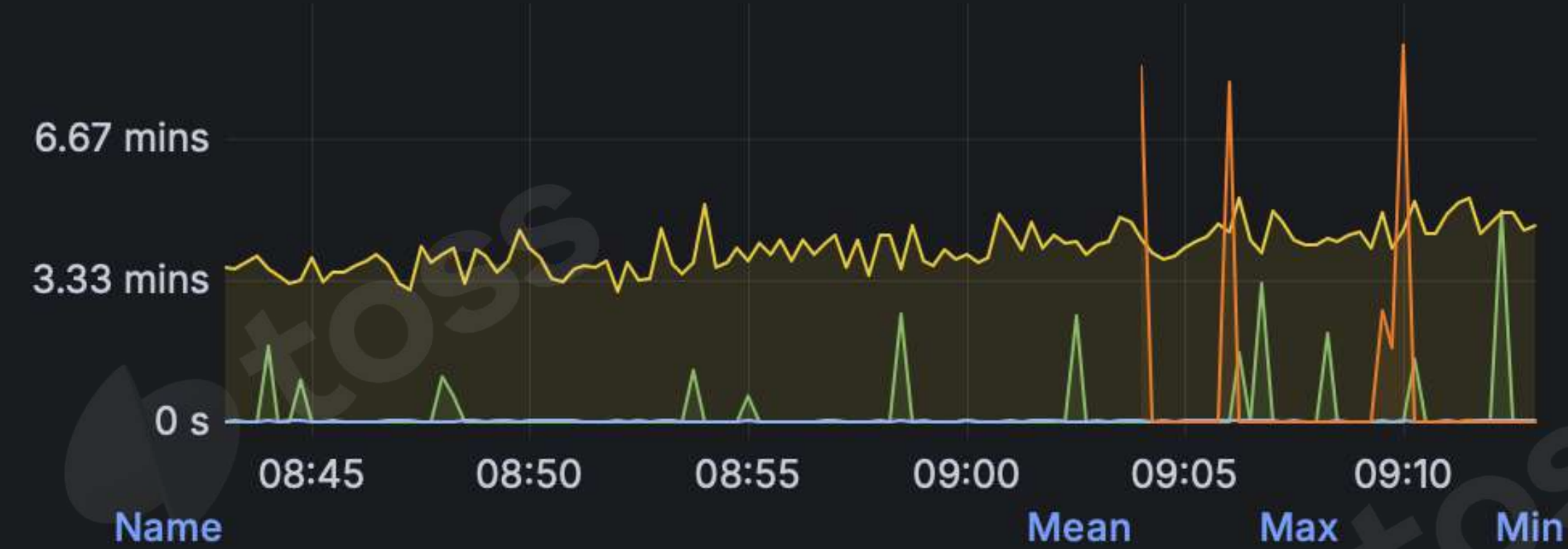
Request Count by API



Request Count by Source Application



Request Duration Avg by Source Application



Request Duration by Source Application



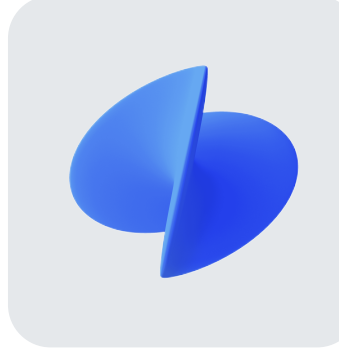


vivasystem 10:10 AM

미사용 서비스 확인 요청 알림

KAFKA 요청 기준으로 일주일간 트래픽이 없었던 서비스 목록입니다.
아직 사용 중인 서비스인지 확인이 필요합니다.

- [@김토스A, @김토스B, @김토스C, @김토스D] 서비스A ...
- [@김토스E, @김토스F, @김토스G, @김토스H] 서비스B ...
- [@김토스I, @김토스J, @김토스K, @김토스L] 서비스C ...
- [@김토스M, @김토스N, @김토스O] 서비스D ...
- [@김토스P, @김토스Q] 서비스E ...
- [@김토스R, @김토스S, @김토스T] 서비스F ...
- [@김토스U, @김토스V, @김토스W, @김토스X] 서비스G ...
- [@김토스Y, @김토스Z] 서비스H ...

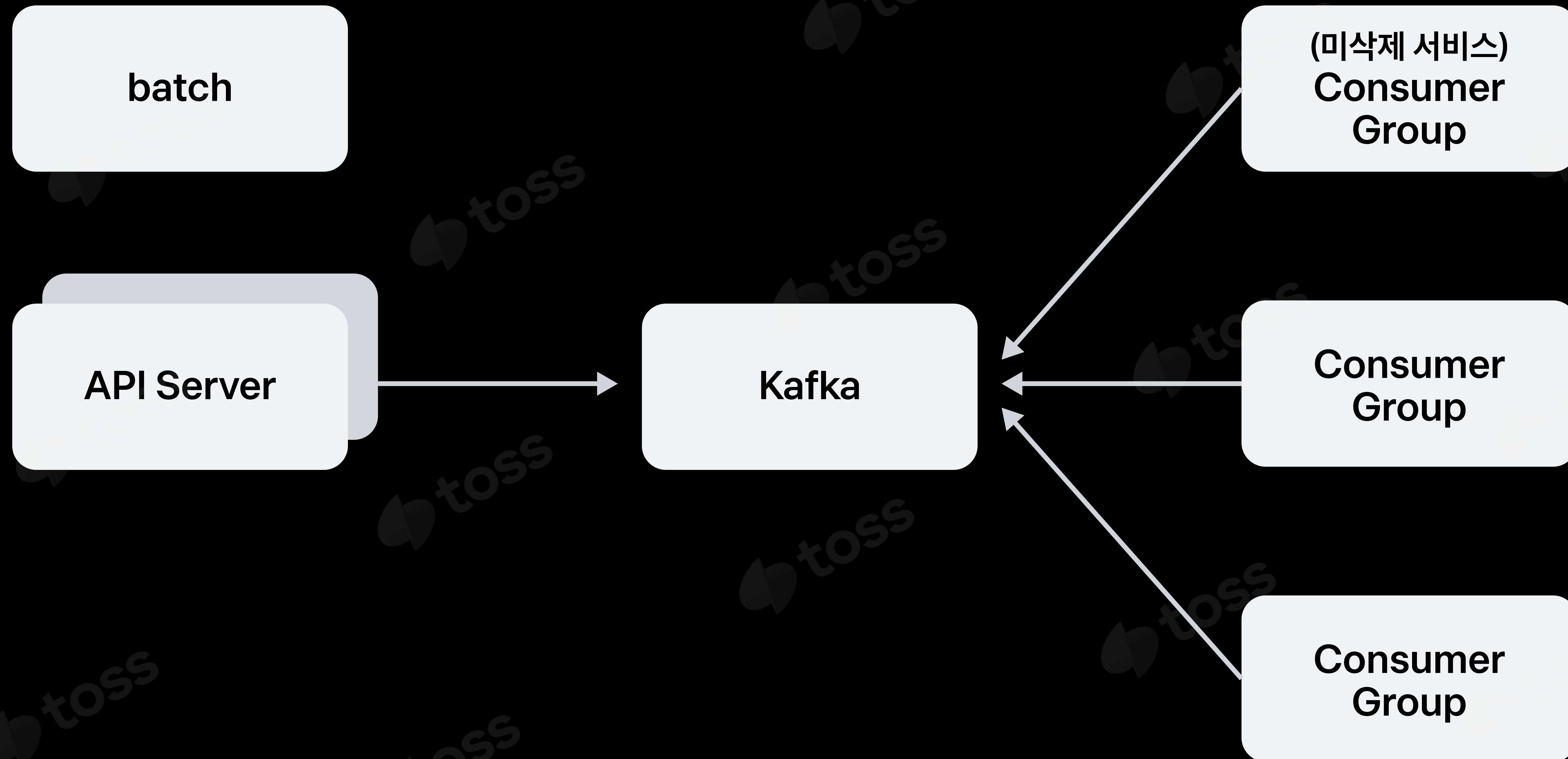


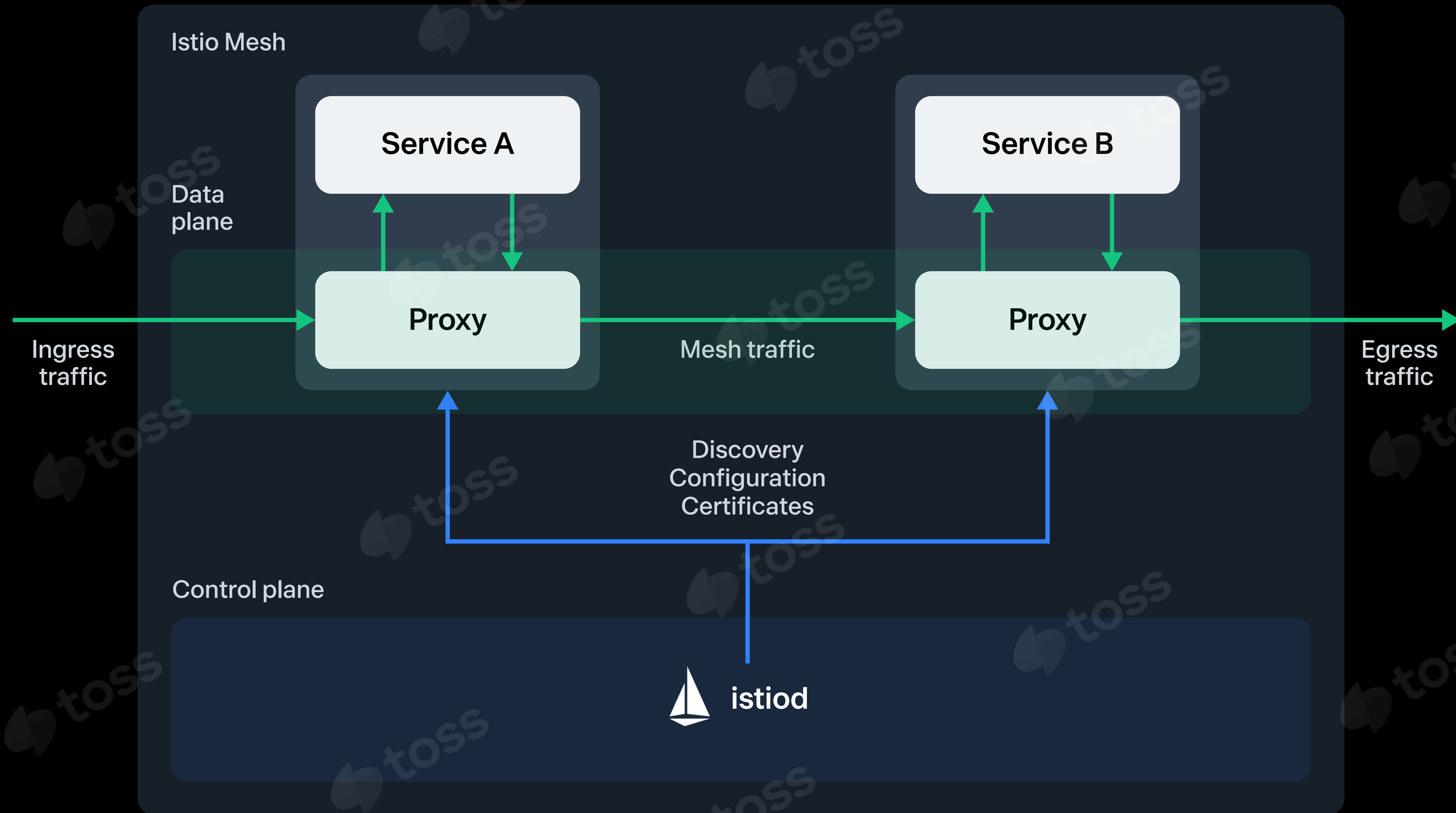
vivasystem 10:10 AM

미사용 HTTP 서비스 확인 요청 알림

HTTP 요청 기준으로 일주일간 트래픽이 없었던 서비스 목록입니다.
아직 사용 중인 서비스인지 확인이 필요합니다.

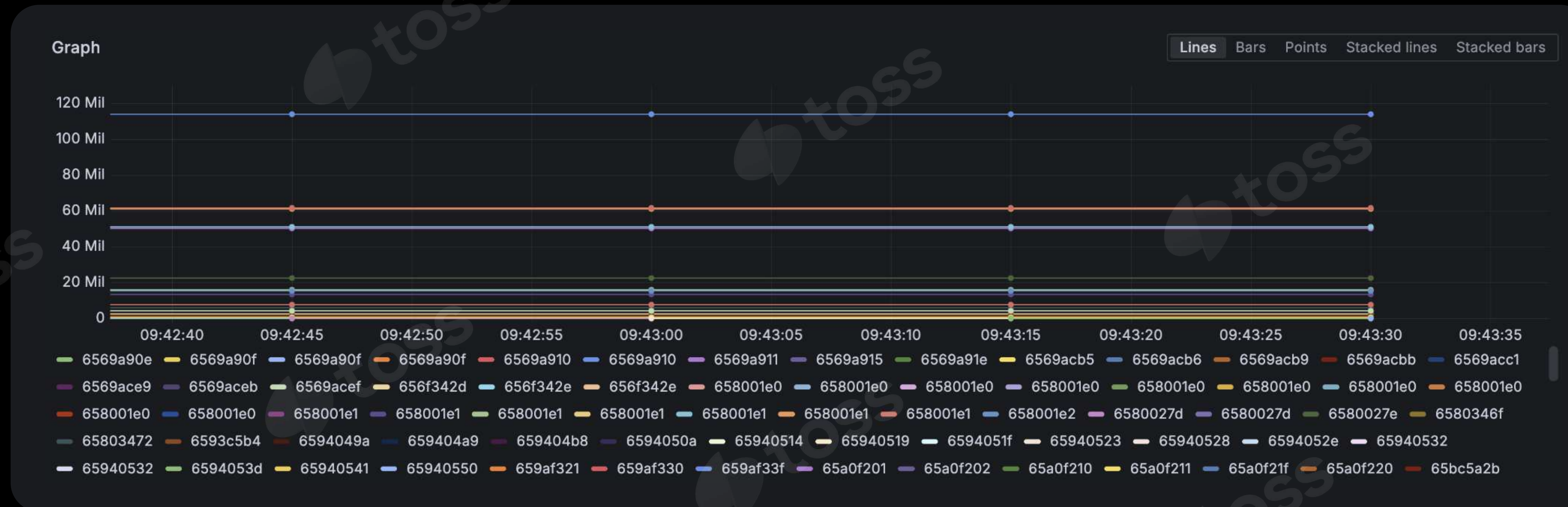
- [@김토스A, @김토스B, @김토스C, @김토스D] 서비스I ...
- [@김토스E, @김토스F, @김토스G, @김토스H] 서비스J ...
- [@김토스I, @김토스J, @김토스K, @김토스L] 서비스K ...
- [@김토스M, @김토스N, @김토스O] 서비스L ...
- [@김토스P, @김토스Q] 서비스M ...
- [@김토스R, @김토스S, @김토스T] 서비스N ...
- [] 서비스O ...
- [] 서비스P ...





HTTP 트래픽

istio_request_total 메트릭을 aggregation 해서 7일치 추출
메트릭이 나오지 않는다면 실제 통신 이력 없음

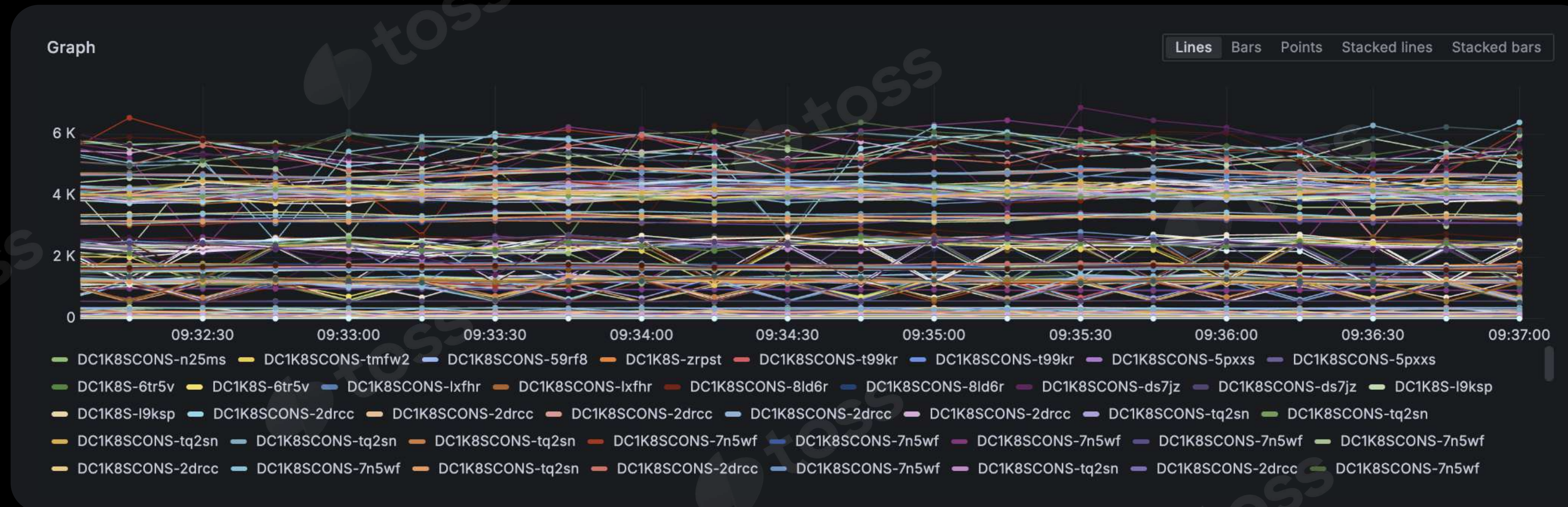


컨슈머 트래픽

spring actuator 에서 kafka 메트릭 추출

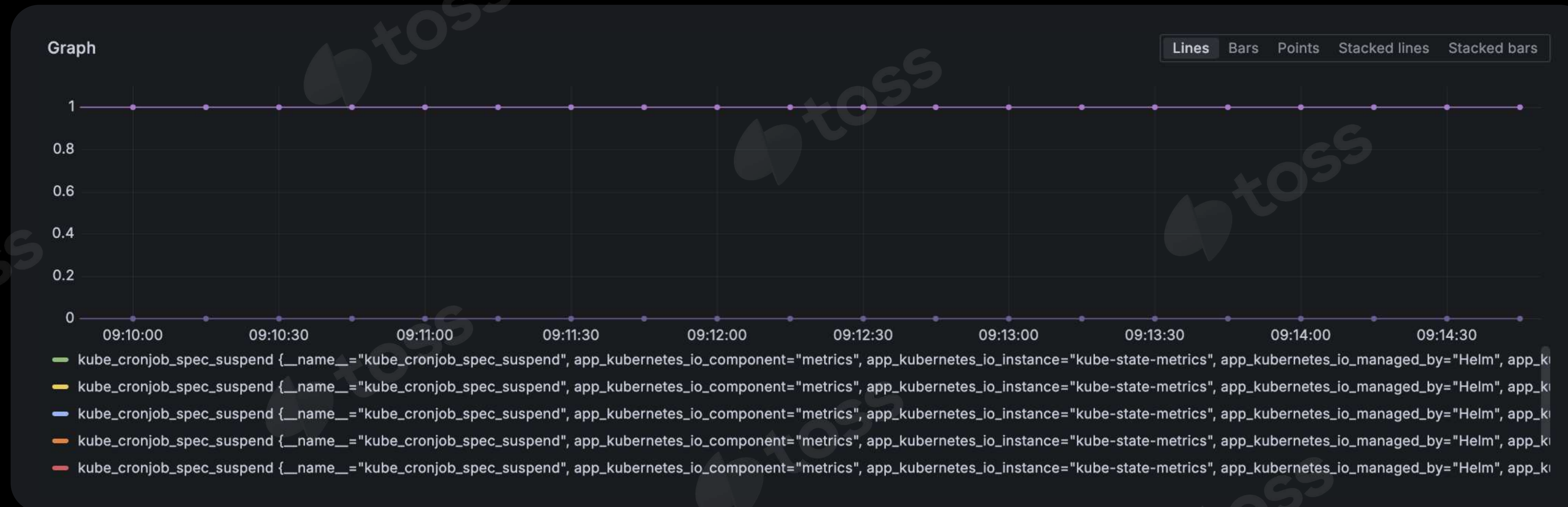
kafka_consumer_fetch_manager_records_consumed_rate

메트릭으로 컨슈밍 이력 확인



BATCH 미사용 여부

kube_cronjob_spec_suspend 메트릭 사용해서 확인



서비스
80개 제거

CPU
1280코어
감소

메모리
1280Gi
감소

