

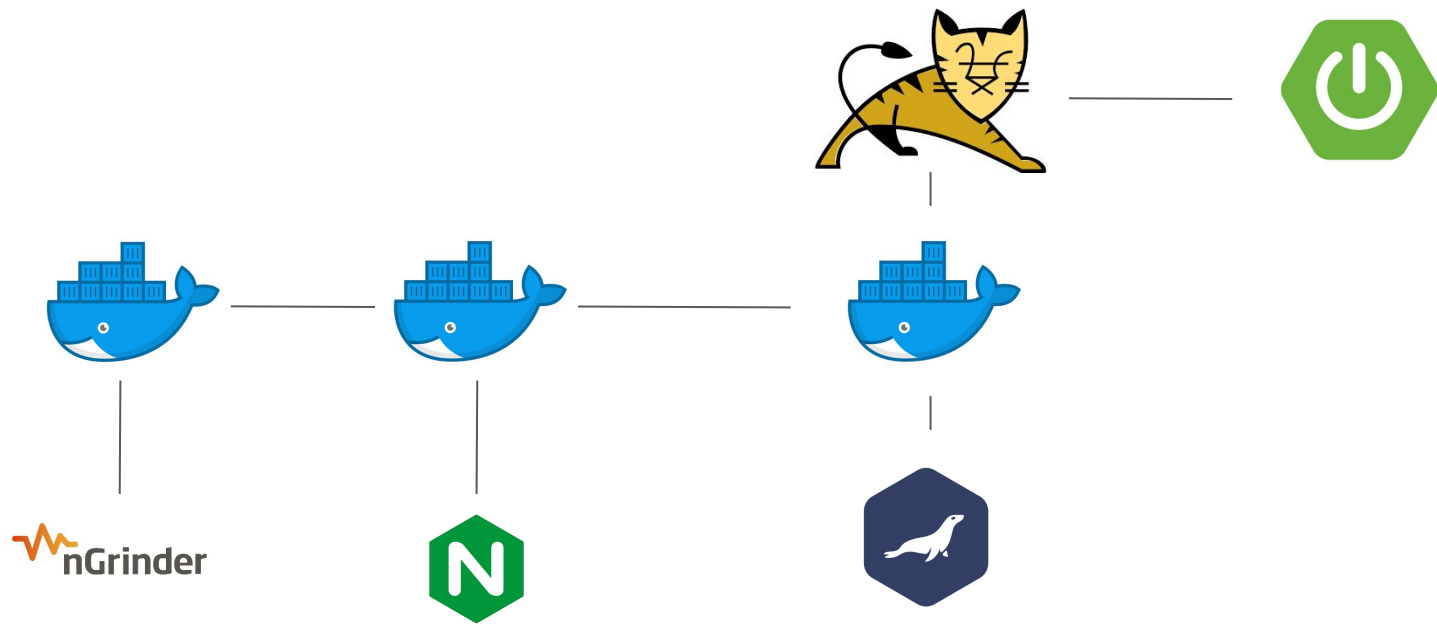
Devops Project

이동혁 - App data
김민우 - Nginx-controller
김민성 - Proxy
김단아 - Nginx-Agent

요구사항

1. 아래 도식과 같이 DB 와 연결된 **SPRING-BOOT APP-SERVER** 준비
2. **NGINX PROXY** 서버는 모든 사용자 요청을 받아 **APP-SERVER** 로 전단 중계하는 대리자 **PROXY** 역할을 하도록 구성
3. 기본 **NGINX** 설정으로 최대 **TPS** 를 도출한다.
4. 최대 **TPS** 의 기준은 (a) 에러가 약 100인 상태에서 (b) 평균응답속도가 초기 대비 1.2 배 상승하지 않는 범위 안에서 (b) 5분간 유지
5. 테스트 환경 구성은 여러대의 컴퓨터를 네트워크로 연결하여 구성하며 각각 구성은 아래와 같음
6. ref(A) 가이드를 참조하여 proxy 단의 캐싱을 적용하여 성능을 개선하고
7. 개선 전후를 비교하는 결과 리포트를 생성
8. 결과 리포트에는 (A) 전/후 성능지표 (B) 캐싱적용 방법 (C) 종합 분석 및 해석 내용을 포함

구성



WBS

목표	결과물	주요업무										
대목표	중목표	소목표	현황			02/28				02/29		
Level-1	Level-2	Level-3	시작	완료	진행중	9:00	11:00	14:00	16:00	9:00	11:00	14:00
프로젝트 계획				o								
	요구사항 분석			o								
	리소스 할당 및 일정 수립			o								
	위험 평가 및 대응 계획 수립			o								
환경 설정					o							
	DB 연결 설정			o								
	SPRING BOOT APP SERVER 설정			o								
	NGINX PROXY 서버 설정				o							
성능 테스트 환경 구축												
	컴퓨터 네트워크 연결											
	각 서버별 구성											
		DB 서버 구성		o	o							
성능 테스트		SPRING BOOT APP SERVER 구성		o	o							
		NGINX PROXY 서버 구성		o	o							
	TPS 측정											
성능 개선 결과 리포트 작성		TPS 측정 환경 설정		o	o							
		TPS 측정 실행		o	o							
		결과 분석 및 문서화		o	o							
	캐싱 적용 전 성능 측정											
		캐싱 미적용 환경 설정			o							
		성능 측정 실행			o							
		결과 분석 및 문서화			o							
	캐싱 적용 후 성능 측정											
		캐싱 적용 환경 설정			o							
		성능 측정 실행			o							
		결과 분석 및 문서화			o							
	성능 측정 전/후 비교 분석				o							
성능 개선 결과 리포트 작성	캐싱 적용 방법 설명				o							
	종합 분석 및 해석 내용 작성				o							

테스트 시나리오

사전 조건

- 테스트 환경에 접근 가능한 유효한 계정이 있어야 함.
- 로그인 페이지에 접근 가능해야 함.

시나리오

- 사용자는 로그인 페이지에 접근한다.
- 사용자는 유효한 사용자 이름과 비밀번호를 입력한다.
- 사용자는 “로그인” 버튼을 클릭하여 로그인을 시도한다.
- 시스템은 입력 받은 사용자 이름과 비밀번호를 바탕으로 **DB**와 대조하여 검증한다.
- 만약 자격 증명이 유효하면, 사용자는 시스템에 성공적으로 로그인된다.
- 만약 자격 증명이 유효하지 않으면, 사용자는 로그인에 실패한다.

테스트 환경구성

- 가상 사용자 그룹 생성: nGrinder를 사용하여 여러 개의 가상 사용자 그룹을 생성한다.
- 로그인 요청 전송: 각 가상 사용자 그룹은 동시에 로그인 요청을 전송한다. 각 사용자는 같은 자격 증명을 사용하여 로그인을 시도한다. 예를 들어, 1000명의 가상 사용자가 1분 동안 로그인을 반복적으로 시도할 수 있도록 설정한다.
- 부하 증가: 부하를 조절하여 서버에 어떤 수준의 로그인 부하가 처리 가능한지 확인한다. 예를 들어, 가상 사용자의 수를 조정하여 서버에 가해지는 부하를 증가시킨다.
- 부하 테스트 결과 분석: 로그인 요청의 응답 시간, 성공률 및 에러율을 분석하여, 시스템이 어느 정도의 로그인 부하를 견딜 수 있는지를 결정한다.

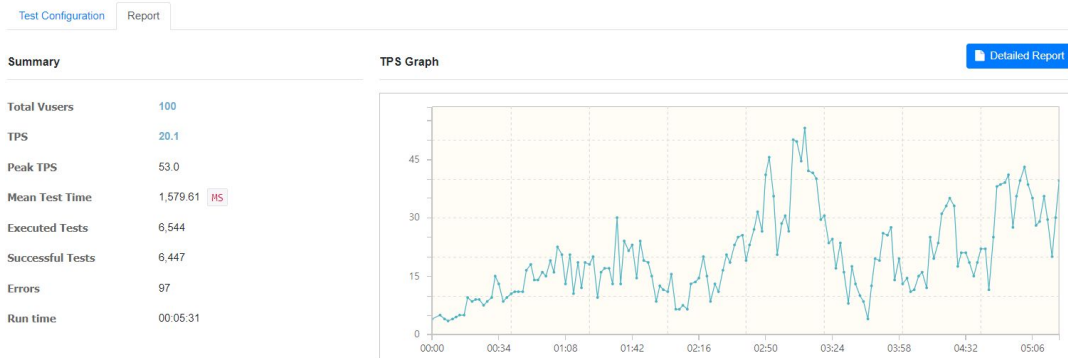
초기 테스트 실행

```
http {
    client_header_buffer_size 16k;
    large_client_header_buffers 4 32k;
    proxy_buffer_size 16k;
    proxy_buffers 4 32k;
    upstream serv {
        server 192.168.0.30:8888;
        # 추가: 연결 시간 초과 시도 설정
        keepalive 32; # 연결 풀 크기 설정
        keepalive_timeout 60s; # 연결 유지 시간 설정
    }
    server {
        listen 80;
        location / {
            # 추가: 프록시 설정
            proxy_read_timeout 60s; # 읽기 타임아웃 설정
            proxy_connect_timeout 10s;
            proxy_send_timeout 10s;
            proxy_http_version 1.1; # HTTP 버전 설정
            proxy_set_header Upgrade $http_upgrade; # 업그레이드 설정
            proxy_set_header Connection 'upgrade'; # 커넥션 설정
            proxy_set_header Host $host;
            proxy_cache_bypass $http_upgrade; # 캐시 설정 우회
            proxy_pass http://serv/;
        }
    }
}
```

프록시 설정

- 초기 설정에는 다양한 설정이 적용된 설정파일을 적용하였습니다.

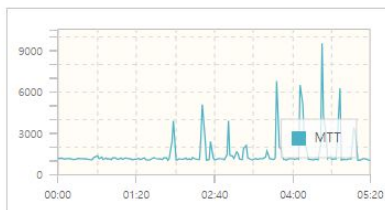
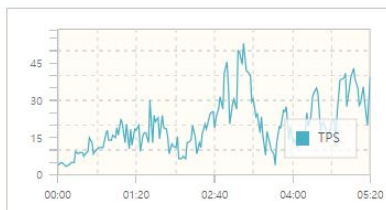
초기 테스트 실행



부하 테스트 결과

- Error : 97
- Err Rate : 1.5%
- MTT : 1,579

<input type="checkbox"/>	Status	Test Name	Script Name	Owner	Start Time	Threshold	TPS	MTT	Err Rate	Users	Actions
<input type="checkbox"/>	●	TODAY we-1 cache, Vuser 100	test_to_192.168.0.11.groovy	admin	2024-02-29 12:09	00:05:30	20.1	1,579.6	1.5%	100	 



테스트 1 실행

```
upstream serv {  
    server 192.168.0.30:8888;  
}  
server {  
    listen 80;  
    location /  
    {  
        proxy_pass http://serv/;  
    }  
}
```


캐싱 설정X

테스트 1 실행



부하 테스트 결과

- Error : 100
- Err Rate : 1.7%
- MTT : 1,722

<input type="checkbox"/>	Status	Test Name	Script Name	Owner	Start Time	Threshold	TPS	MTT	Err Rate	Vusers	Actions
<input type="checkbox"/>		we-1 cache, Vuser 100 캐싱완전제거	test_to_192.168.0.11.groovy	admin	2024-02-29 12:30	00:05:30	18.3	1,722.2	1.7%	100	 



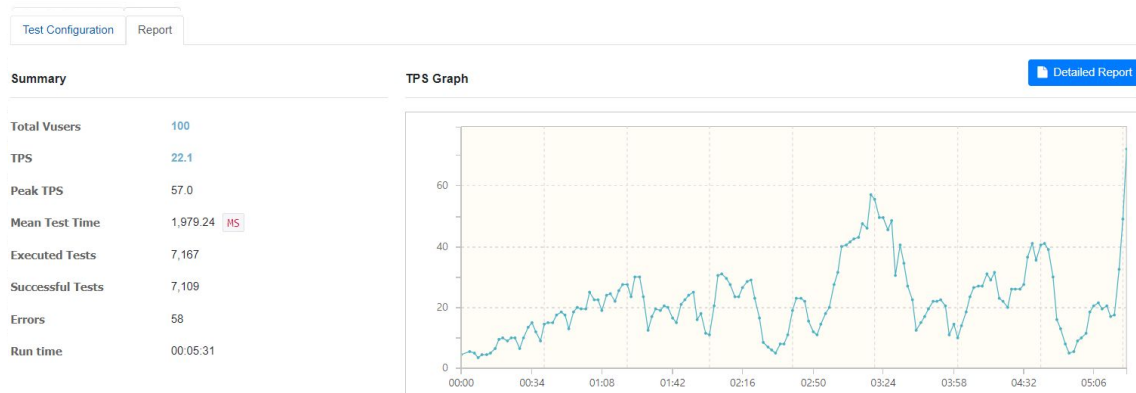
테스트 2 실행

```
upstream serv {  
    server 192.168.0.30:8888;  
}  
server {  
    listen 80;  
    location /  
    {  
        proxy_read_timeout 10s;  
        proxy_connect_timeout 10s;  
        proxy_pass http://serv/;  
    }  
}
```

프록시 설정

- timeout 값에 대한 문제인지 확인하기 위해 timeout 값을 설정하였습니다

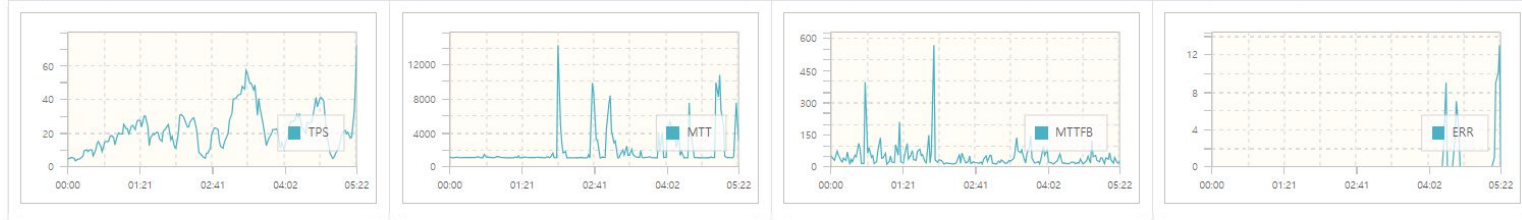
테스트 2 실행



부하 테스트 결과

- Error : 58
- Err Rate : 0.8%
- MTT : 1,979

<input type="checkbox"/>	Status	Test Name	Script Name	Owner	Start Time	Threshold	TPS	MTT	Err Rate	Vusers	Actions
<input type="checkbox"/>		we-1 cache, Vuser 100 캐싱제거	test_to_192.168.0.11.groovy	admin	2024-02-29 12:21	00:05:30	22.1	1,979.2	0.8%	100	



테스트 3 실행

```
upstream serv {  
    server 192.168.0.30:8888;  
}  
proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_zone:10m inactive=60m;  
server {  
    listen 80;  
    location /  
    {  
        proxy_cache my_zone;  
        proxy_cache_key $host$uri$is_args$args;  
        proxy_read_timeout 10s;  
        proxy_connect_timeout 10s;  
        proxy_pass http://serv;  
    }  
}
```

프록시 설정

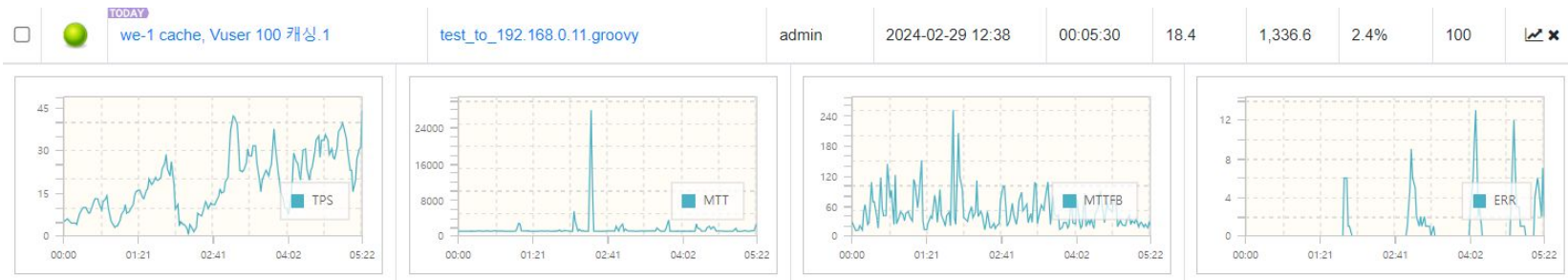
- 캐시를 저장하기 위한 경로 설정 뒤 캐시를 활성화하였습니다.

테스트 3 실행



부하 테스트 결과

- Error : 147
- Err Rate : 2.4%
- MTT : 1,336



테스트 4 실행

```
proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_zone:10m inactive=60m;
server {
    listen 80;
    location /
    {
        proxy_cache_methods GET HEAD POST;
        proxy_cache my_zone;
        proxy_cache_key $host$uri$is_args$args;
        proxy_cache_valid 200 401 301 304 5s;
        add_header X-Cache-Status $upstream_cache_status;
        add_header Cache-Control "public";
        proxy_pass http://192.168.0.30:8888/;
    }
    location /login
    {
        proxy_cache_methods GET HEAD POST;
        proxy_cache my_zone;
        proxy_cache_key $host$uri$is_args$args;
        proxy_cache_valid 200 401 301 304 5s;
        add_header X-Cache-Status $upstream_cache_status;
        add_header Cache-Control "public";
        proxy_pass http://192.168.0.30:8888/login;
    }
}
```

프록시 설정

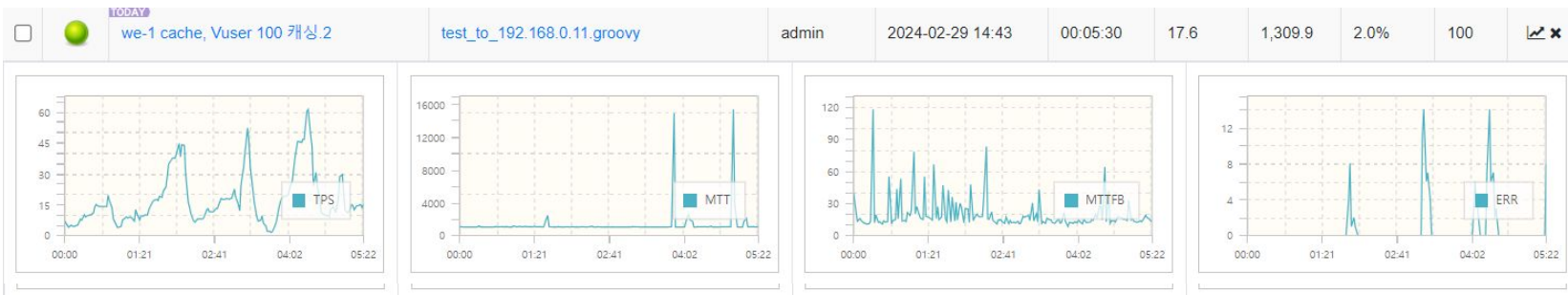
- 테스트4에서 api를 post 방식으로 적용하고 있기 때문에 proxy_method를 설정하여 POST 요청을 처리하도록 구성하였습니다.

테스트 4 실행



부하 테스트 결과

- Error : 118
- Err Rate : 2.0%
- MTT : 1,309



결과 분석

no cache				cache			
TPS	Peak TPS	MTT	Errors	TPS	Peak TPS	MTT	Errors
22.1	57	979	58	20.1	53	579	97
18.3	66	722	100	18.4	42	336	147
21.0	56	531	120	17.6	61	309	118
20.46	59.6	744	92.6	18.7	52	408	120.6

TPS : 9% 하락, Peak TPS : 14% 하락, MTT : 46% 하락, Errors : 30% 증가

시행 횟수가 적어 유의미한 결과가 아닐 수 있음

결과 분석

- TPS : 9% 하락, Peak TPS : 14% 하락, MTT : 46% 하락, Errors : 30% 증가
- 시행 횟수가 적어 유의미한 결과가 아닐 수 있음
- curl 로 요청을 한 결과 캐시에서 응답을 찾을 수 없어 **MISS**가 발생함
- 앱서버에 정적파일이 존재하지 않기에 캐시가 생성되지 않았을 것이라 추측됨
- timeout 설정을 한 결과 에러가 감소하는 모습을 확인

개선 사항

- Proxy의 timeout 에러 지속적으로 발생

```
2024-02-29 12:56:47 2024/02/29 03:56:47 [error] 282#282: *4625 upstream timed out (110: Connection timed out) while connecting to upstream, client: 172.17.0.1, server: , request: "POST /api/authenticate HTTP/1.1", upstream: "http://192.168.0.30:8888/api/authenticate", host: "192.168.0.11:9006"
```

```
2024-02-29 12:56:48 2024/02/29 03:56:48 [error] 280#280: *4662 upstream timed out (110: Connection timed out) while connecting to upstream, client: 172.17.0.1, server: , request: "POST /api/authenticate HTTP/1.1", upstream: "http://192.168.0.30:8888/api/authenticate", host: "192.168.0.11:9006"
```

- 캐싱 파일 생성확인
- TPS 상승 , ERROR 감소