Node.js A6 Team

Test PDCA _ Plan

전제조건

검증을 통한 시스템 응답속도 및 안정성 개선

✅ 테스트 대상 시스템 범위

- Test Server > LB > Web Application Server > Data Base

✅ 데이터 양

- 사용자 정보 200,000 건 / 상품 데이터 : 10,000 건 / 사용 기간 : 1년
- 사용자 패턴 : 로그인 > 전체페이지 이동
  > 입찰가능 상품목록 조회 [2분 미만] > 이벤트 페이지 접속
  > 입찰 시도 [평균 인당 30분 미만 / 분당 1회 요청]

✅ 지속적인 성능 유지 기간 : 데이터량이 급증하는 상품별 입찰 마감 30분간 성능 유지

✅ 부하방법

- nGrinder Controller : Local
- nGrinder Agent : EC2 t.Series
- V-User : 1 > 10 > 20 > 100 > 200 > 400 > 1000 > 2000
- 부하 구간
  - 1구간 : Attack > Web Server (Nginx) - GET : static File
  - 2구간 : Attack > Web Application (Nest server) - GET api request

✅ 목표값

- Latency : 500ms 이하 / 부하 발생시 1000ms 이하
- Throughput
  DAU = 100,000
  피크 AU = 60,000
  평균 AU = 20 (Peck x 3 60)
  TPS = 약 210 ~ 300 tps [(100,000 x 20 / 86,400) x 3 x 3]

✅ 테스트 환경

1. Load Balancer : HAProxy
2. Instance : Naver Cloud Instance
3. DB : AWS RDS Aurora postgreSQL

✅ 테스트 시나리오

1. 로그인 > Token 발급 > Main Page
2. Event Page/:rafflesId > Token 검증 > Page render

Test PDCA _ Plan

검증을 통한 시스템 응답속도 및 안정성 개선

전제조건

✅ 테스트 대상 시스템 범위

- Test Server > LB > Web Application Server > Data Base

✅ 데이터 양

- 사용자 정보 200,000 건 / 상품 데이터 : 10,000 건 / 사용 기간 : 1년
- 사용자 패턴 : 로그인 > 전체페이지 이동
  > 입찰가능 상품목록 조회 [2분 미만] > 이벤트 페이지 접속
  > 입찰 시도 [평균 인당 30분 미만 / 분당 1회 요청]

✅ 지속적인 성능 유지 기간 : 데이터량이 급증하는 상품별 입찰 마감 30분간 성능 유지

✅ 부하방법

- nGrinder Controller : Local
- nGrinder Agent : EC2 t.Series
- V-User : 1 > 10 > 20 > 100 > 200 > 400 > 1000 > 2000
- 부하 구간
  - 1구간 : Attack > Web Server (Nginx) - GET : static File
  - 2구간 : Attack > Web Application (Nest server) - GET api request

✅ 목표값

- Latency : 500ms 이하 / 부하 발생시 1000ms 이하
- Throughput
  DAU = 100,000
  피크 AU = 60,000
  평균 AU = 20 (Peck x 3 60)
  TPS = 약 210 ~ 300 tps [(100,000 x 20 / 86,400) x 3 x 3]

✅ 테스트 환경

1. Load Balancer : HAProxy
2. Instance : Naver Cloud Instance
3. DB : AWS RDS Aurora postgreSQL

✅ 테스트 시나리오

1. 로그인 > Token 발급 > Main Page
2. Event Page/:rafflesId > Token 검증 > Page render

Test PDCA _ Do