SOA#1

Do not take it for granted

W czym rzecz?

- Pisanie aplikacji to jedna sprawa
- Utrzymanie aplikacji to sprawa zupełnie inna
- Utrzymanie wymaga szerszego spojrzenia

Tworzenie aplikacji

- Lokalne środowisko dev nie odpowiada temu co mamy produkcyjnie
- config/development.rb
- Konfiguracja poza aplikacją - ENV
- Produkcyjnie przeważnie mamy dużo więcej zależności - API, środowisko, DB, background jobs, itd.

Production vs. Development

- Więcej niż jeden serwer
- Procesowanie w tle
- Obciążenie aplikacji
- Wiele punktów wejścia do aplikacji - użytkownicy, API, event consumer
- Komunikacja z żywymi systemami zewnętrznymi
- Często całkiem złożony własny ekosystem

Development vs. Production

- Błędy pojawiają się produkcyjnie

- Próbujemy je reprodukować i naprawić lokalnie

Przykład #1

Aplikacja przez 1h w ciągu doby działa nieprawidłowo

Why care?

Why care?

Przez pozostałe 23h działa prawidłowo ;-)

Developerzy kochają rozwiązywać problemy!

Próba reprodukcji problemu lokalnie

Próba reprodukcji problemu lokalnie

FAIL

Próba reprodukcji problemu produkcyjnie

Próba reprodukcji problemu produkcyjnie

FAIL

Próba odtworzenia kontekstu produkcyjnego lokalnie i reprodukcji problemu

Próba odtworzenia kontekstu produkcyjnego lokalnie i reprodukcji problemu

FAIL

Próba odtworzenia produkcyjnie z zachowaniem kontekstu

Próba odtworzenia produkcyjnie z zachowaniem kontekstu

SUCCESS!

Honey I’m home!

Honey I’m home!

No, tak jakby…

```
SELECT * FROM items
WHERE date < CURRENT_DATE
ORDER BY date DESC limit 1;
```

Wiemy co nie działa, wiemy kiedy nie działa, nie wiemy dlaczego nie działa…

```
SELECT * FROM items
 WHERE date < CURRENT_DATE
ORDER BY date DESC limit 1;
```

Tylko jedna wartość może być różna pomiędzy środowiskami

```
SELECT CURRENT_DATE;
```

```
SELECT CURRENT_DATE;
```

Data w PostgresSQL jest inna niż faktyczna!

Dlaczego?

- Serwer został uruchomiony z błędnym ustawieniem strefy czasowej
- PostgreSQL został uruchomiony z błędnym ustawieniem które zaczytał z serwera
- Strefa czasowa na serwerze została poprawiona
- PostgreSQL dalej pracował z błędnym ustawieniem strefy czasowej

Data dzienna zmieniała się w PostgreSQL o jedną godzinę wcześniej niż rzeczywiście…

„Jutro to dziś, tyle że jutro.”

Przykład #2

Zniknęły dane zapisane w Redis

Próba reprodukcji problemu lokalnie

Próba reprodukcji problemu lokalnie

FAIL

Próba reprodukcji problemu produkcyjnie

Próba reprodukcji problemu produkcyjnie

FAIL

Analiza konfiguracji produkcyjnej instancji Redis

Analiza konfiguracji produkcyjnej instancji Redis

SUCCESS!

Dlaczego?

- Dodana jakiś czas wcześniej biblioteka korzystała z tej samej instancji Redis jako magazynu cache (długoterminowego)

- Zużycie pamięci w Redis stopniowo rosło, aż osiągnięty został zadany w configu limit

- Po osiągnięciu limitu, Redis robił miejsce na nowe dane wg ustawionej eviction policy, w tym przypadku usuwając najmniej używane klucze.

Dane „znikały” z Redis bez oczywistego powodu i w przypadkowych momentach…

Wnioski

- Aplikacja to nie tylko kod
- Context is king!
- „Kod nie modyfikuje się sam”
- Szerokie spojrzenie

Dziękuję

Kuba Łopusiński / jakub.lopusinski@ironin.pl