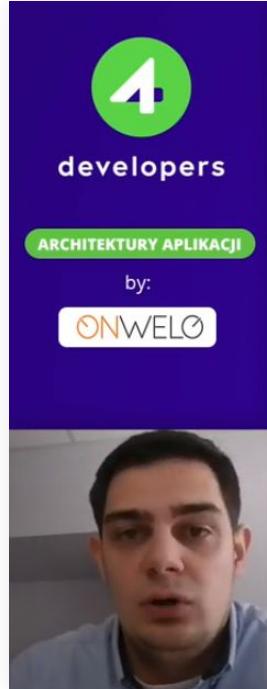


Multitenancy from scratch

Marek Chrapek

ONWELG



Hello Multitenancy World!

by Marek Chrapek and
Paweł Młynarczyk

ONWELØ

Multitenancy from scratch

Marek Chrapek

ONWELG

O mnie



Marek Chrapek

Architect / Technical Leader



/marek-chrapek



/mchrapek



Hello Astra - inteligentny system rekrutacyjny



Pozyskiwanie kandydatów



Algorytmy rekomendacji AI



Ochrona danych osobowych



Prowadzenie rekrutacji



Automatyczna analiza rynku



Analiza danych i raportowanie



Agenda

01

dłaczego
architektura
multitenant?

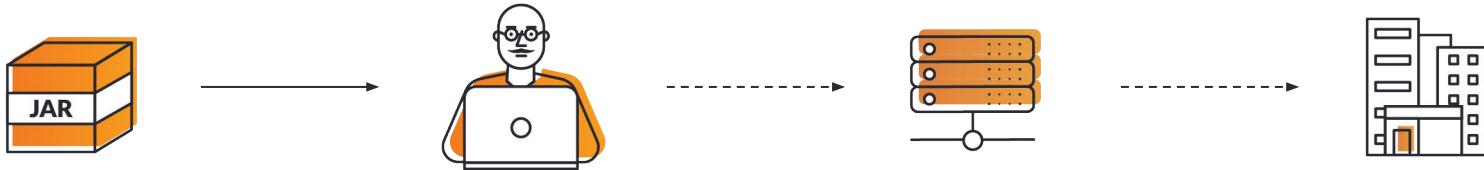
02

przygotowanie
środowiska

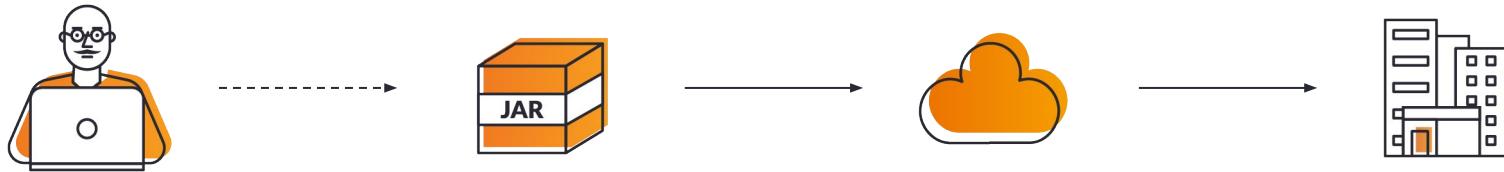
03

implementacja
architektury
multitenant

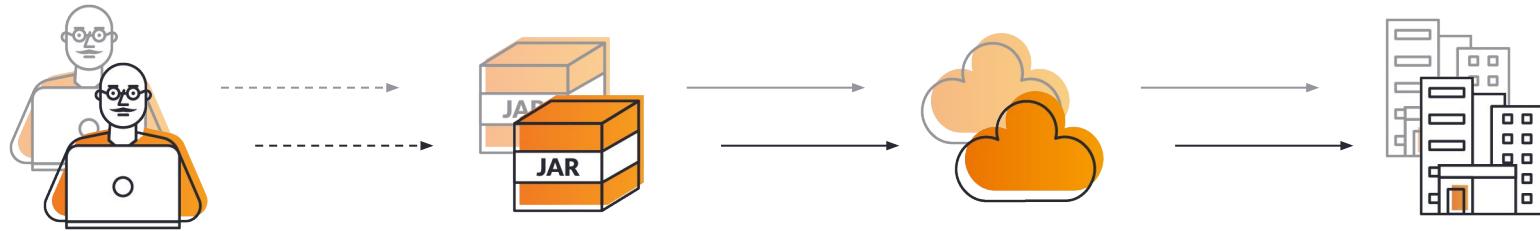
Oprogramowanie single tenant



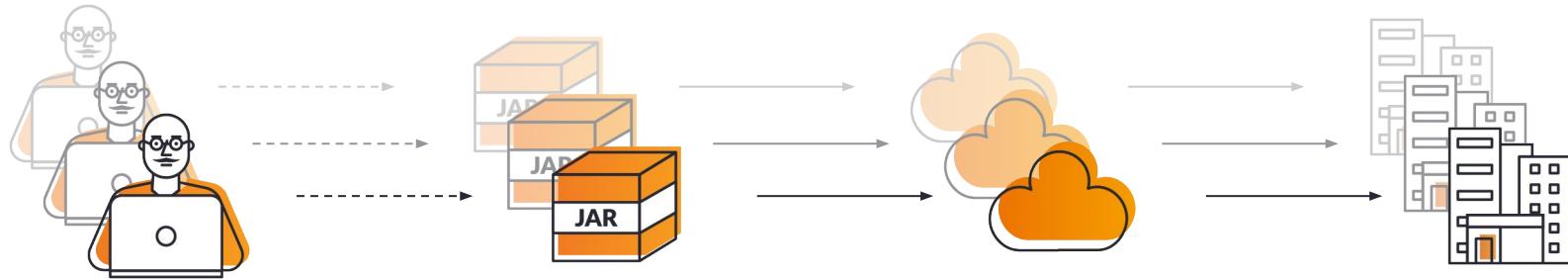
Oprogramowanie single tenant



Oprogramowanie single tenant



Oprogramowanie single tenant





Źródło: <https://giphy.com/gifs/theoffice-sVnKj2wDhUTsFKFWhx>



Źródło: <https://giphy.com/gifs/theoffice-sVnKj2wDhUTsFKFWhx>



Źródło: <https://giphy.com/gifs/theoffice-sVnKj2wDhUTsFKFWhx>

Problemy

- Aktualizacja wszystkich środowisk



Źródło: <https://giphy.com/gifs/theoffice-sVnKj2wDhUTsFKFWhx>

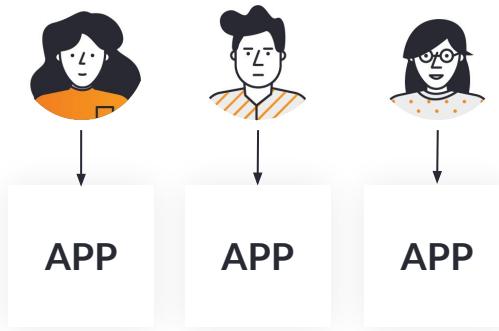
Problemy

- Aktualizacja wszystkich środowisk
- Olbrzymi koszt utrzymywania infrastruktury



Źródło: <https://giphy.com/gifs/theoffice-sVnKj2wDhUTsFKFWhx>

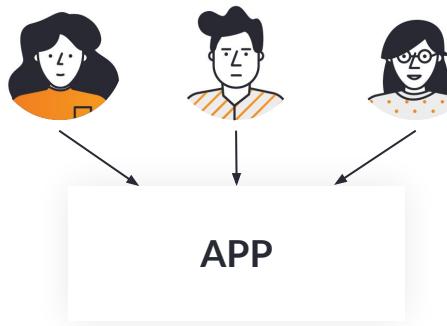
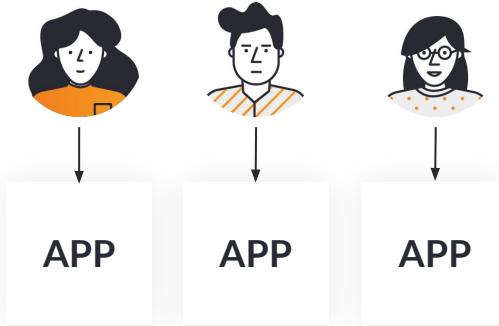
Single tenant vs multitenant



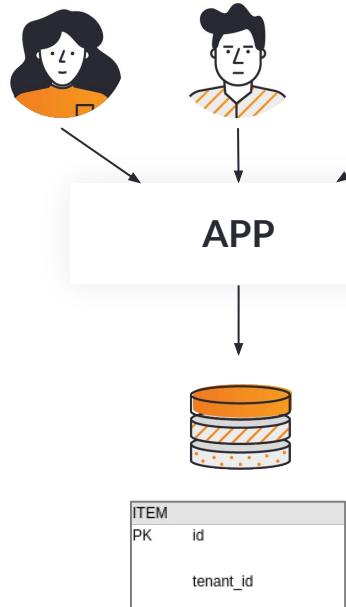
Single tenant vs multitenant



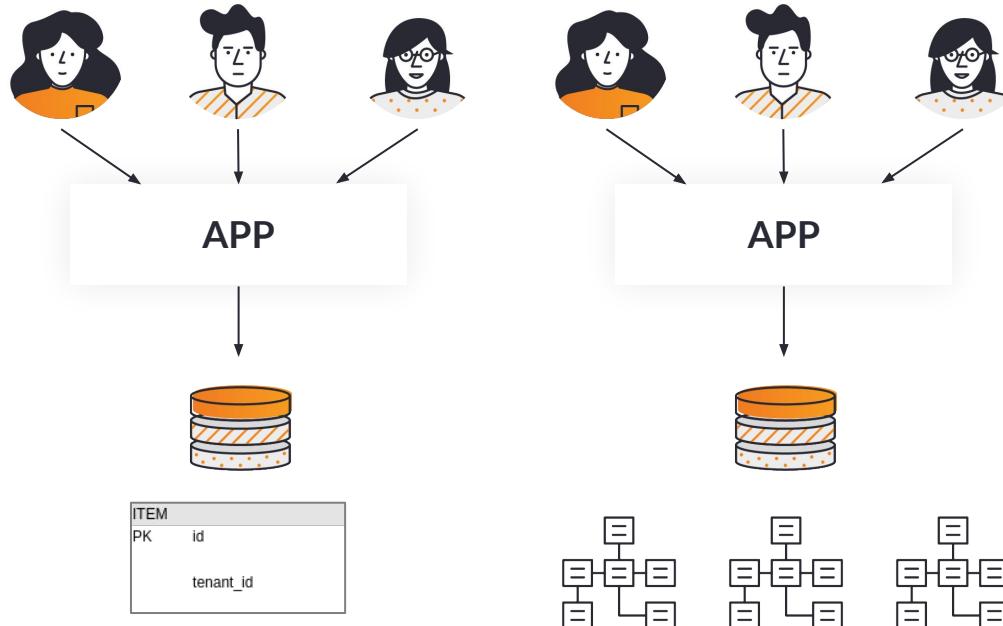
Single tenant vs multitenant



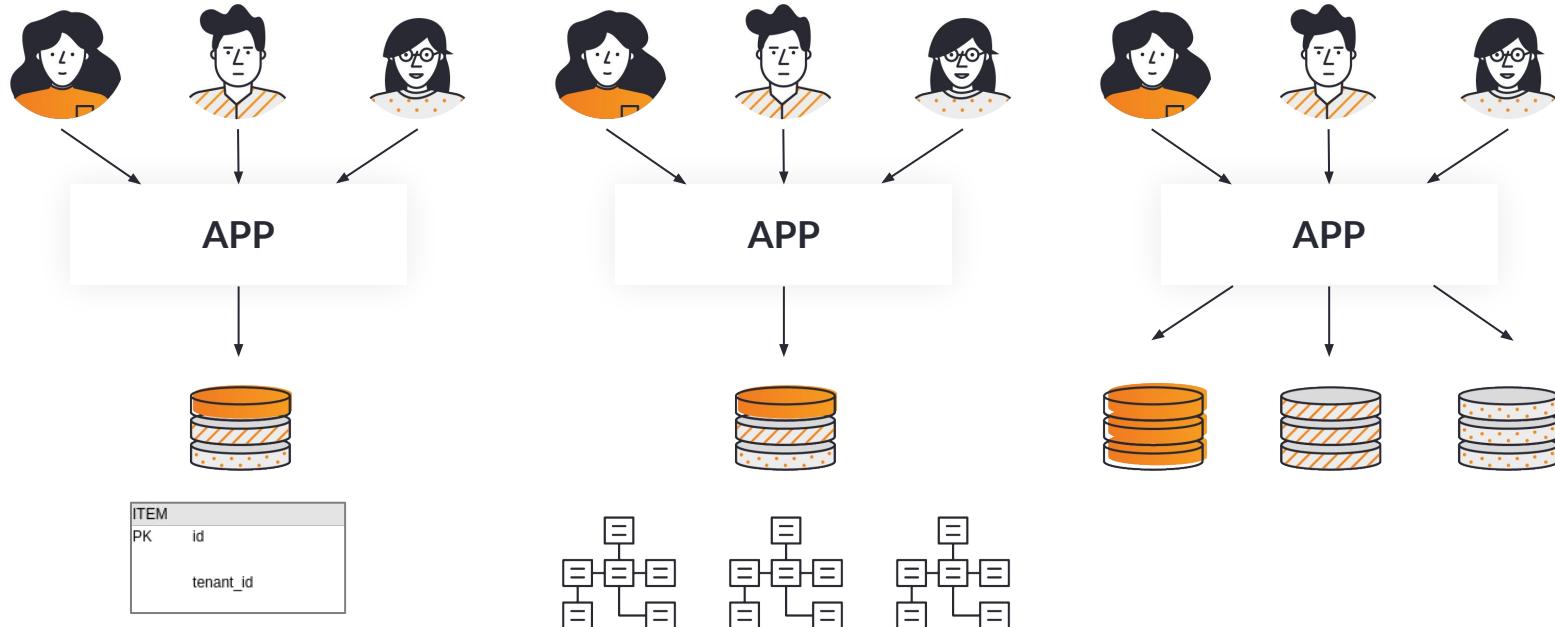
Multitenant – separacja danych



Multitenant – separacja danych



Multitenant – separacja danych



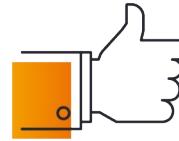
Multitenant - zalety

- Efektywność kosztowa



Multitenant - zalety

- Efektywność kosztowa
- Prostsze utrzymanie



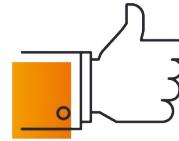
Multitenant - zalety

- Efektywność kosztowa
- Prostsze utrzymanie
- Dodawanie nowych klientów



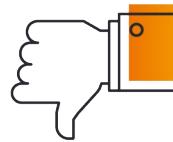
Multitenant - zalety

- Efektywność kosztowa
- Prostsze utrzymanie
- Dodawanie nowych klientów
- Łatwość skalowania



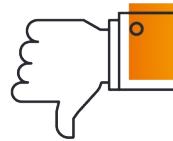
Multitenant - wady

- Głośni sąsiedzi



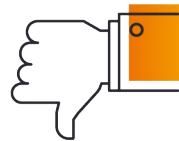
Multitenant - wady

- Głośni sąsiedzi
- Ryzyko łatwego zepsucia aplikacji wszystkim klientom



Multitenant - wady

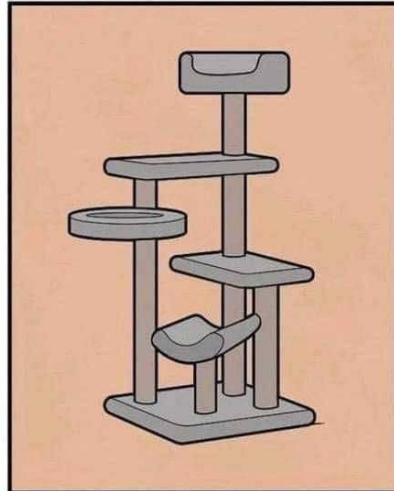
- Głośni sąsiedzi
- Ryzyko łatwego zepsucia aplikacji wszystkim klientom
- Rosnący poziom skomplikowania



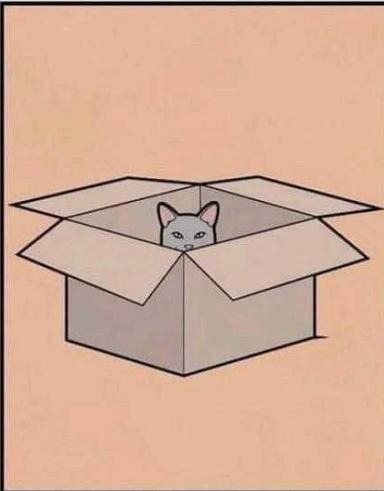
punkt
startowy



Product features



User needs



⌚ _yes_but

Źródło: <https://whiteboards.io/blog/agile-product-development-ultimate-guide/>

Drivery architektoniczne

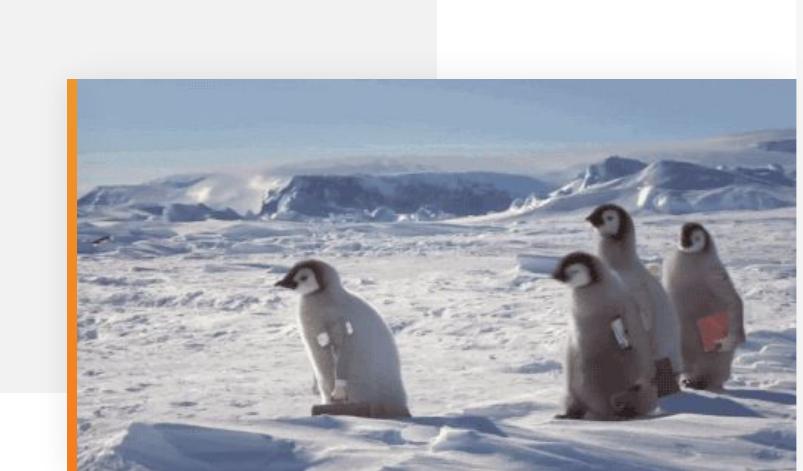


Źródło:

https://www.reddit.com/r/birdswithharms/comments/6b0dnu/in_the_animal_world/

Drivery architektoniczne

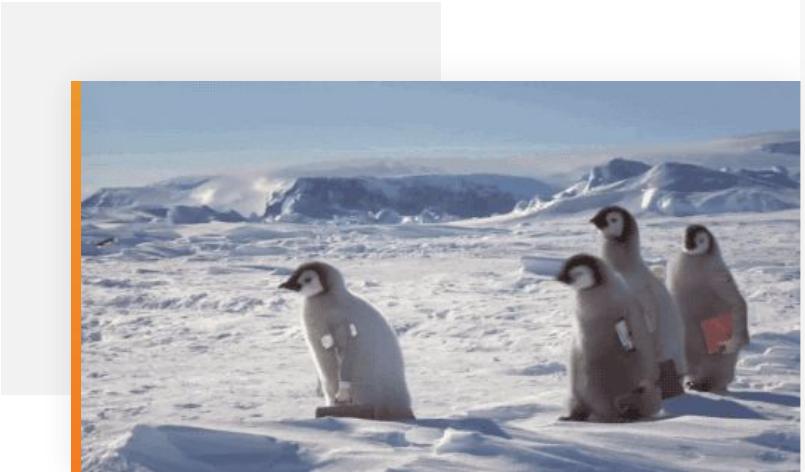
- Łatwość wdrożenia nowej wersji



Źródło:
https://www.reddit.com/r/birdswithharms/comments/6b0dnu/in_the_animal_world/

Drivery architektoniczne

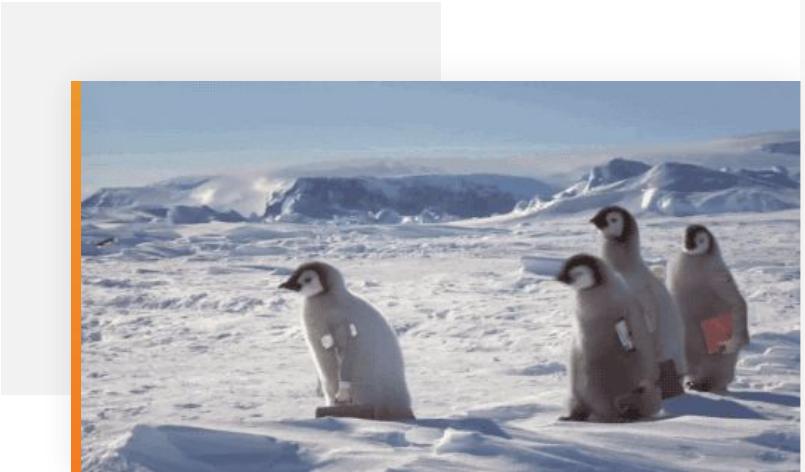
- Łatwość wdrożenia nowej wersji
- Migracja do chmury



Źródło:
https://www.reddit.com/r/birdswithharms/comments/6b0dnu/in_the_animal_world/

Drivery architektoniczne

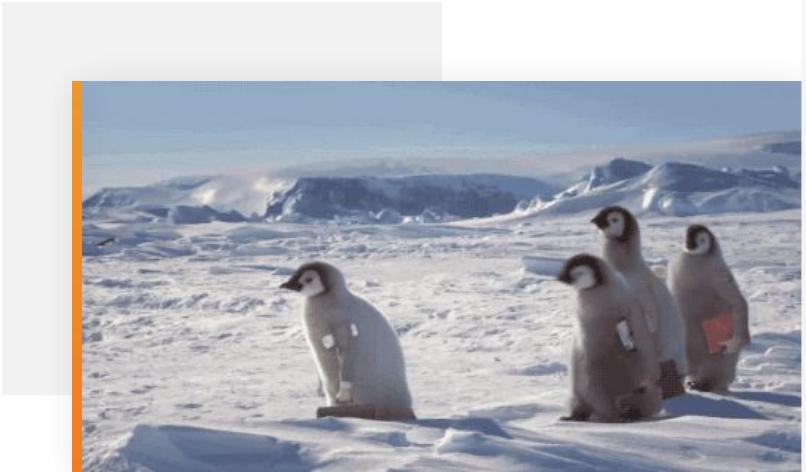
- Łatwość wdrożenia nowej wersji
- Migracja do chmury
- Zmiana partnerstwa



Źródło:
https://www.reddit.com/r/birdswithharms/comments/6b0dnu/in_the_animal_world/

Drivery architektoniczne

- Łatwość wdrożenia nowej wersji
- Migracja do chmury
- Zmiana partnerstwa
- Sprawna obsługa nowych klientów



Źródło:
https://www.reddit.com/r/birdswithharms/comments/6b0dnu/in_the_animal_world/

Drivery architektoniczne

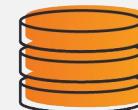
- Łatwość wdrożenia nowej wersji
- Migracja do chmury
- Zmiana partnerstwa
- Sprawna obsługa nowych klientów
- Zmniejszenie kosztów infrastruktury



Źródło:
https://www.reddit.com/r/birdswithharms/comments/6b0dnu/in_the_animal_world/

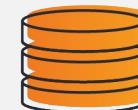
Pierwsza wersja aplikacji

- Tomcat (WAR)
- Spring
- Keycloak
- MongoDB



Pierwsza wersja aplikacji

- Tomcat (WAR)
 - Spring
 - Keycloak
 - MongoDB
-
- Produkt
 - Pierwszy klient



przygotowanie aplikacji

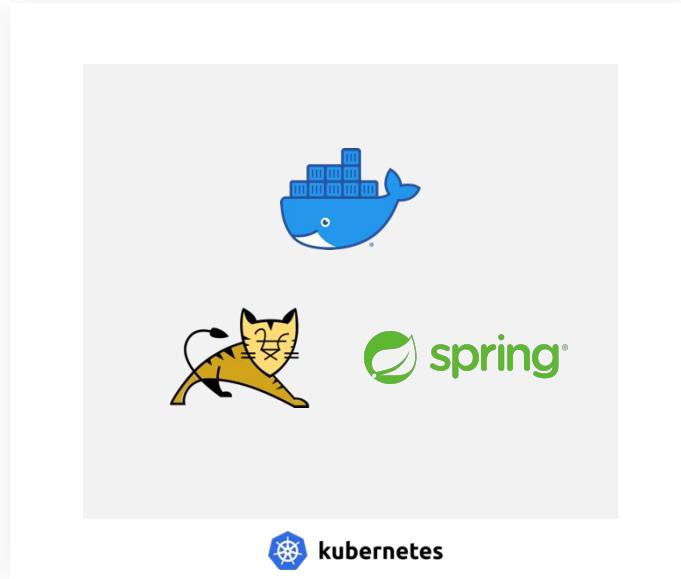
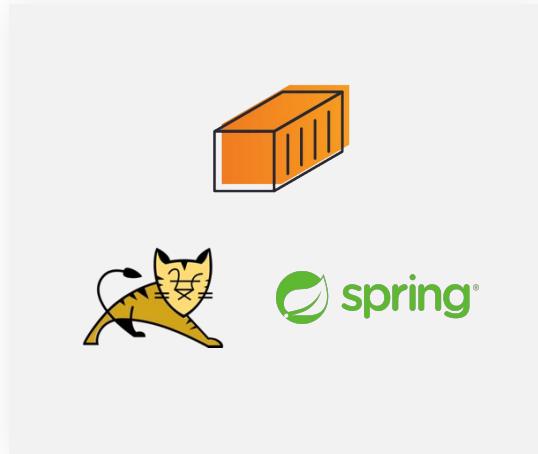


Migracja do chmury



Migracja do chmury

“Lift and shift”



Wdrożenie nowego klienta



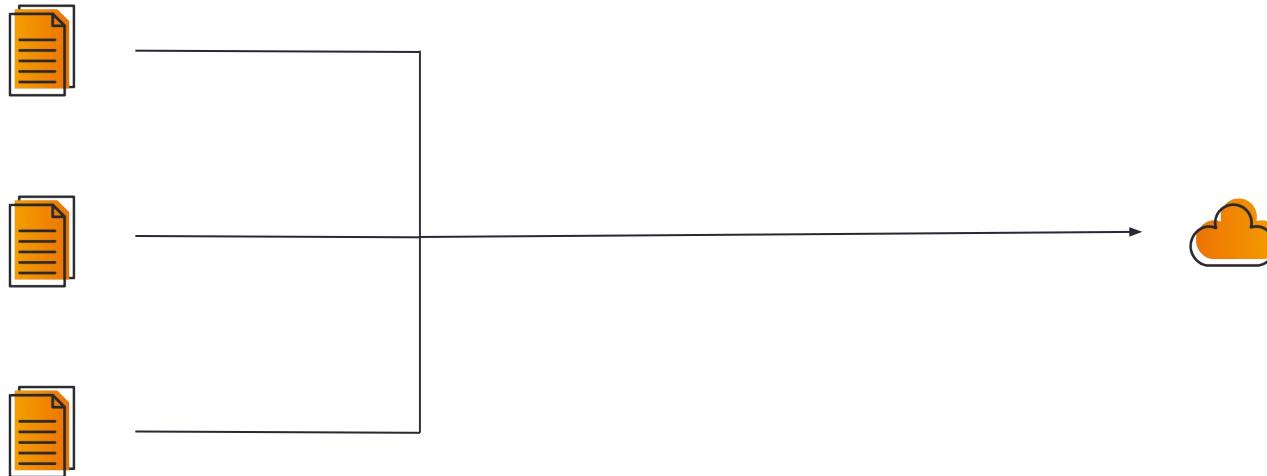
kubernetes

Wdrożenie nowego klienta

Namespace i Ingress przeznaczony dla każdego klienta

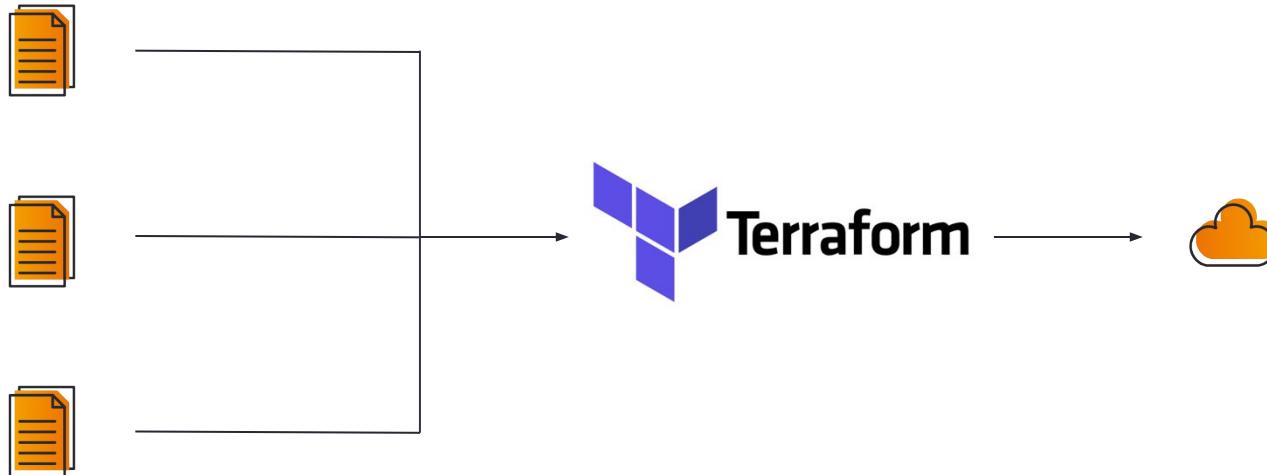


Wdrożenie nowego klienta

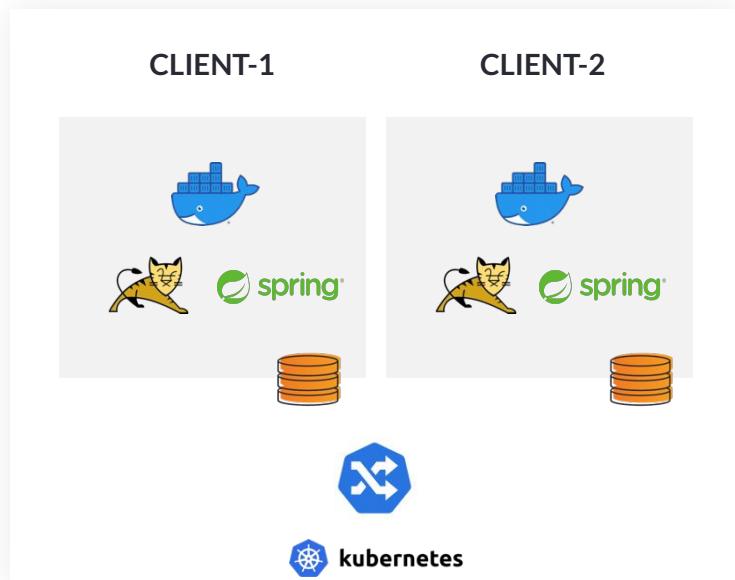


Wdrożenie nowego klienta

Automatyzacja

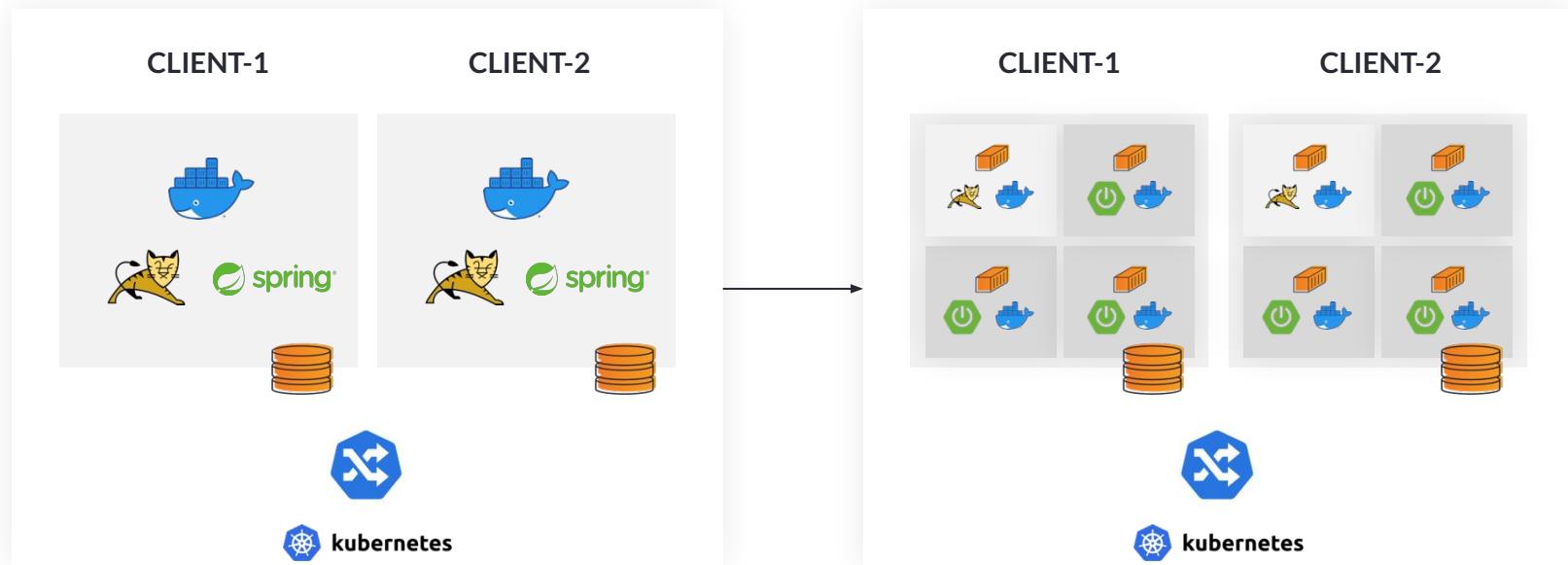


Monolityczna aplikacja



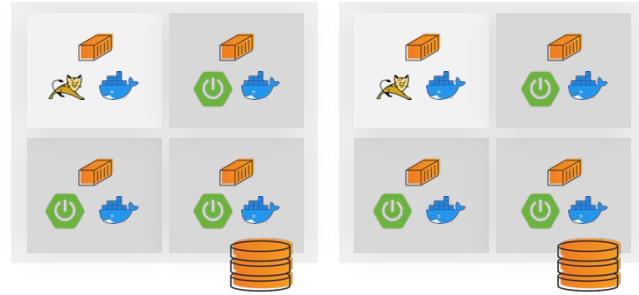
Monolityczna aplikacja

Migracja do (mikro)serwisów



Komunikacja pomiędzy serwisami

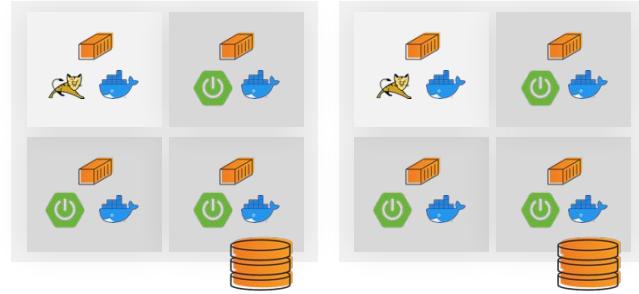
CLIENT-1 CLIENT-2



Komunikacja pomiędzy serwisami

Stopniowa migracja na komunikację z wykorzystaniem Kafki

CLIENT-1 CLIENT-2

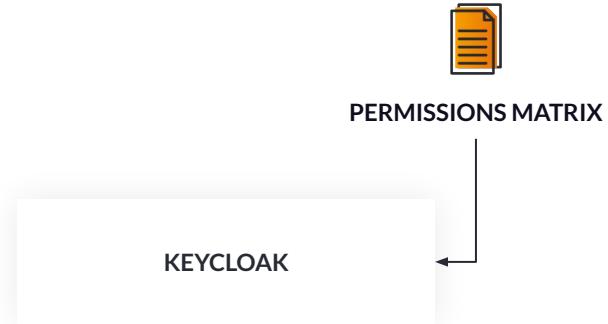


kubernetes

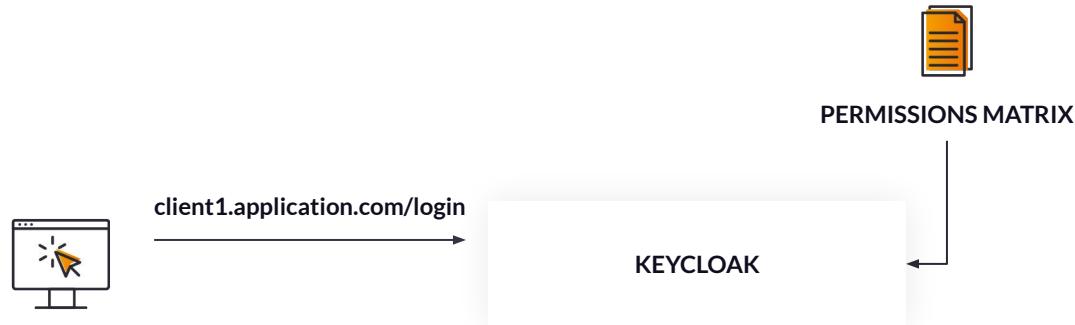
Udostępnianie funkcjonalności dla tenantów

KEYCLOAK

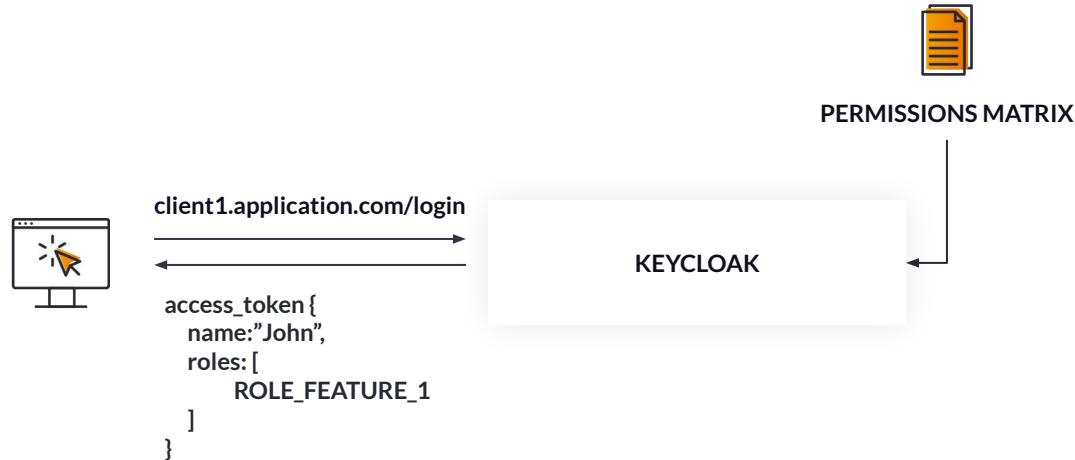
Udostępnianie funkcjonalności dla tenantów



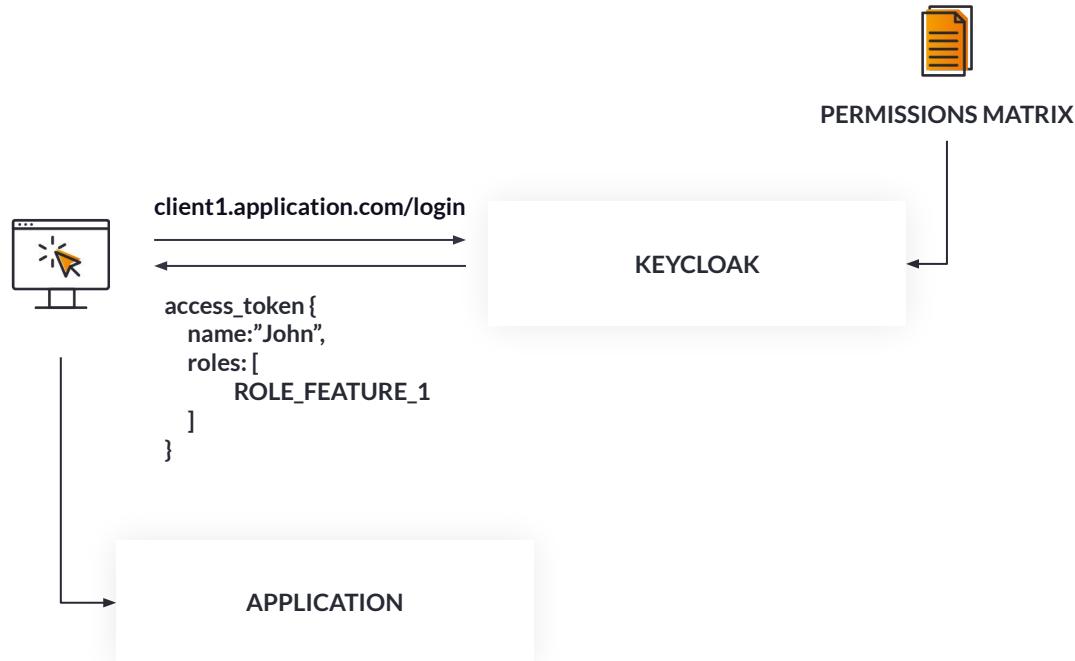
Udostępnianie funkcjonalności dla tenantów



Udostępnianie funkcjonalności dla tenantów

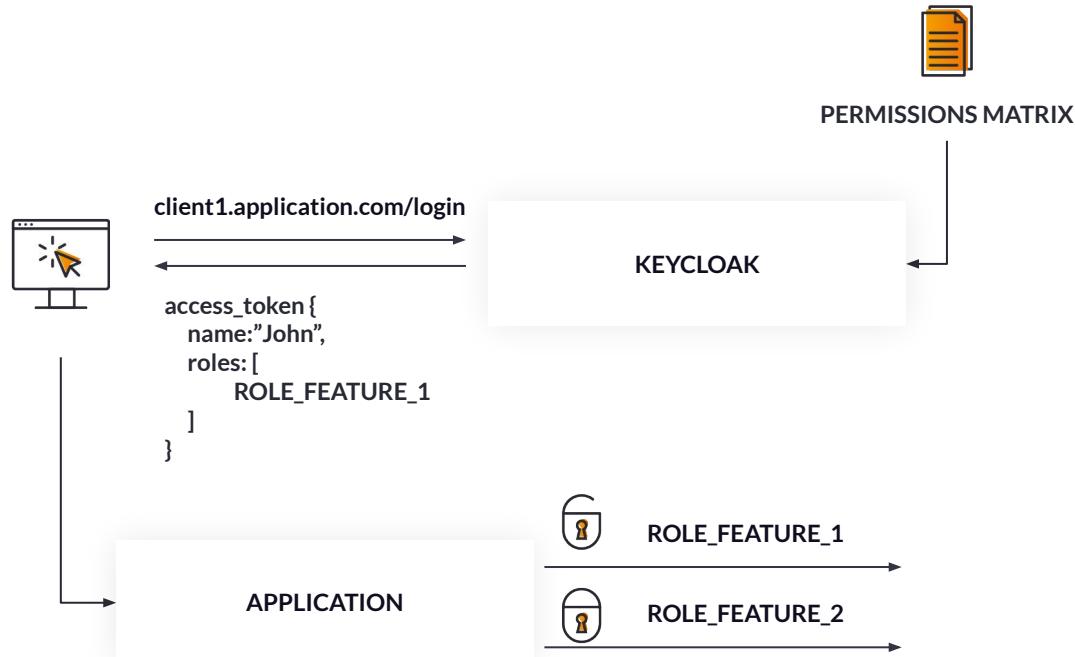


Udostępnianie funkcjonalności dla tenantów



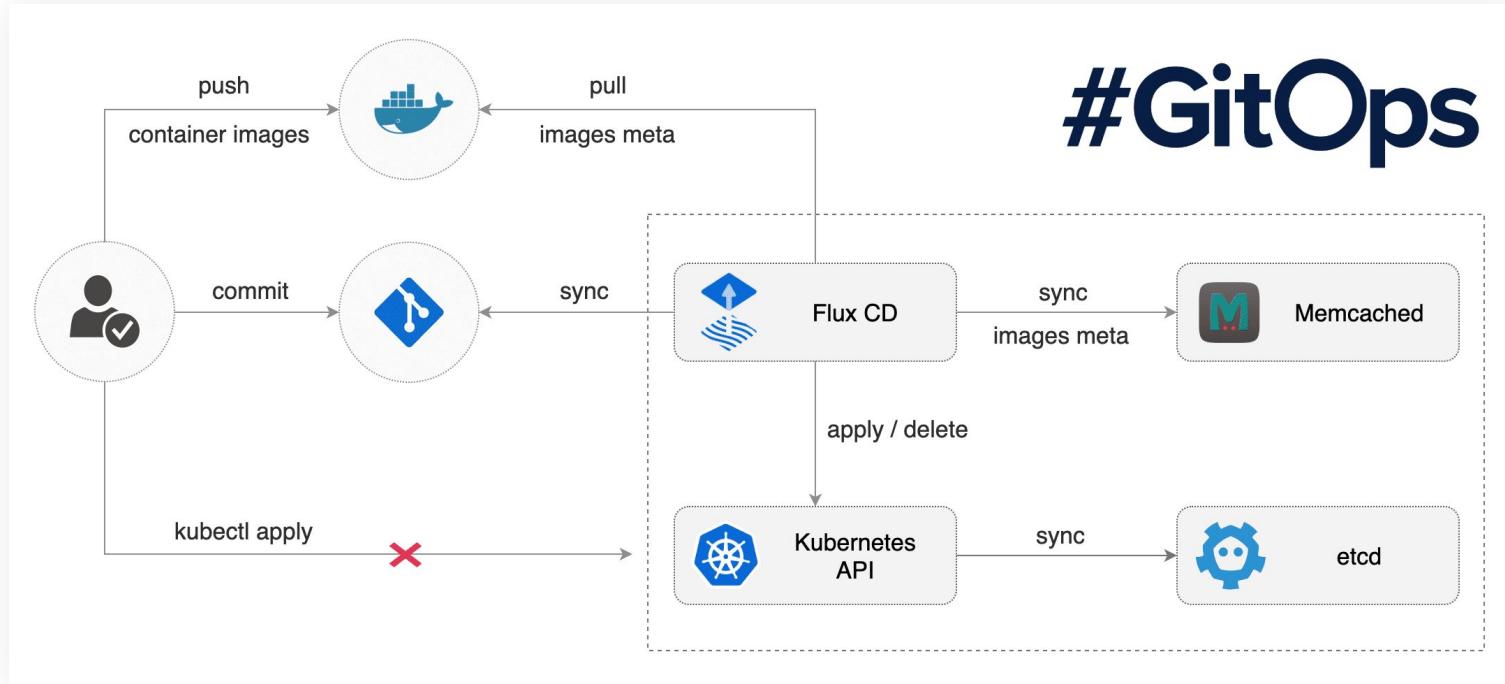
Udostępnianie funkcjonalności dla tenantów

- Feature toggle
- Osobny realm dla klienta



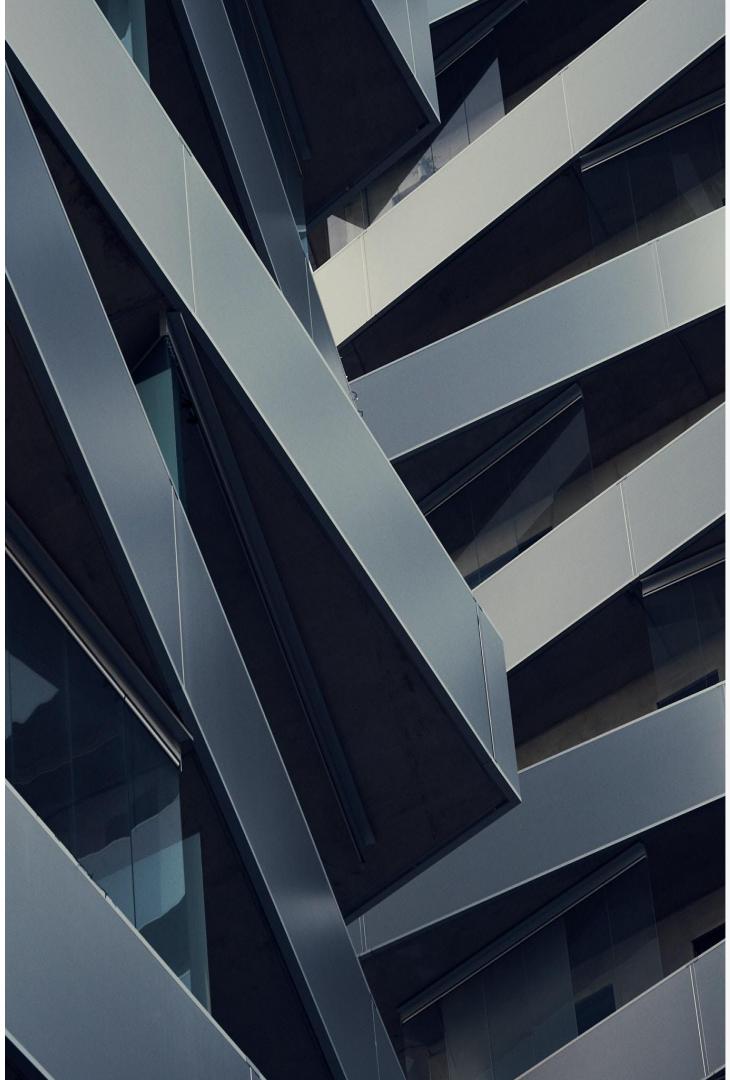
Stan serwera

#GitOps



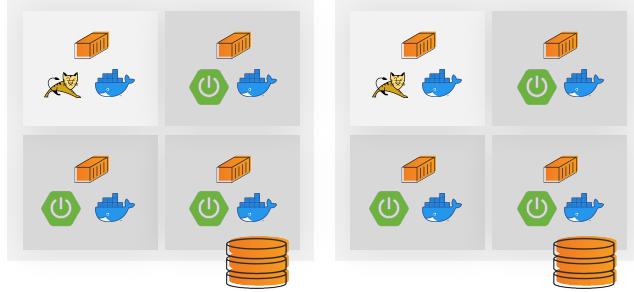
Źródło: <https://vnclagoon.com/gitops-why-your-company-should-embrace-it/>

implementacja architektury multitenant



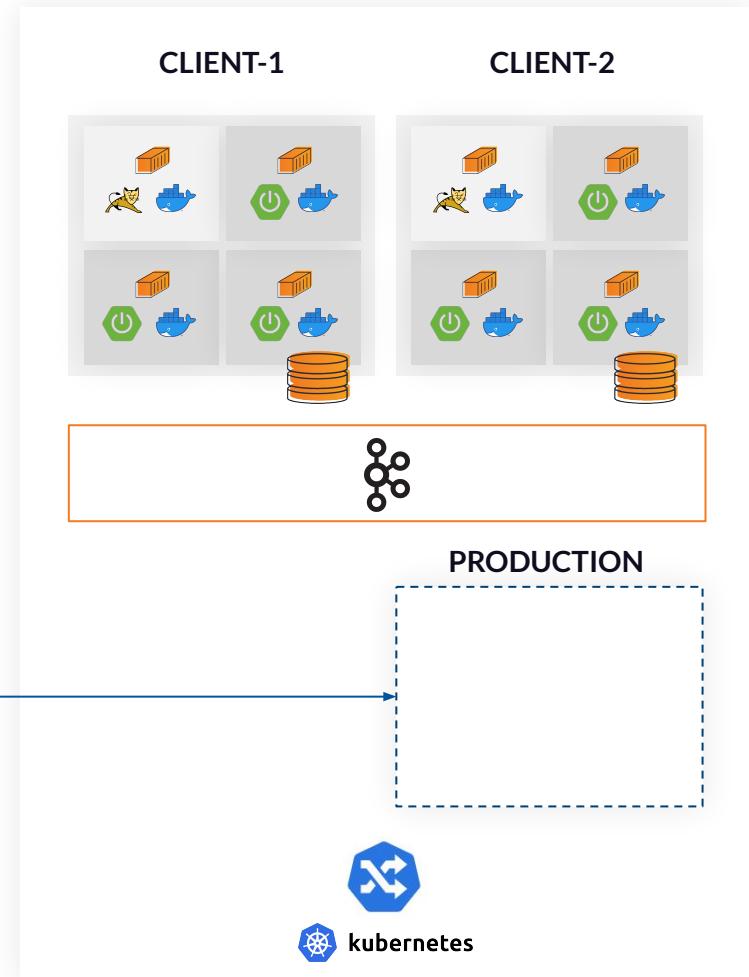
Wydzielenie pierwszego modułu

CLIENT-1 CLIENT-2



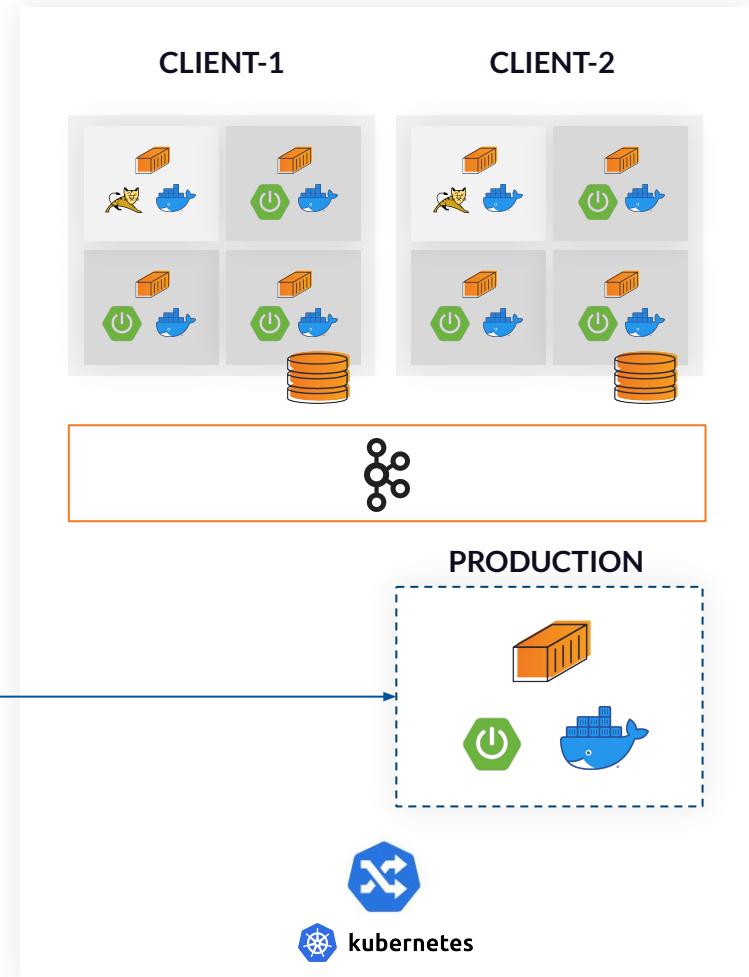
Wydzielenie pierwszego modułu

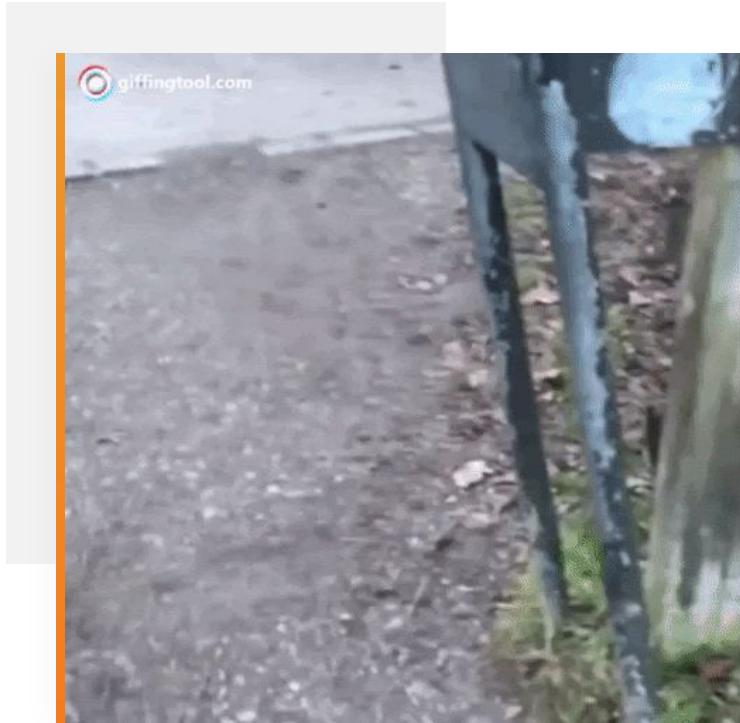
- Stworzenie nowego namespace'a



Wydzielenie pierwszego modułu

- Stworzenie nowego namespace'a

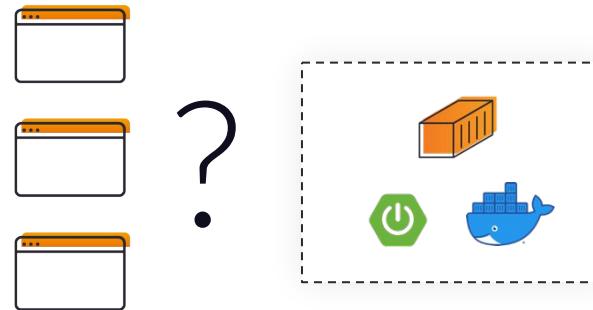




Źródło: <https://www.giffingtool.com>

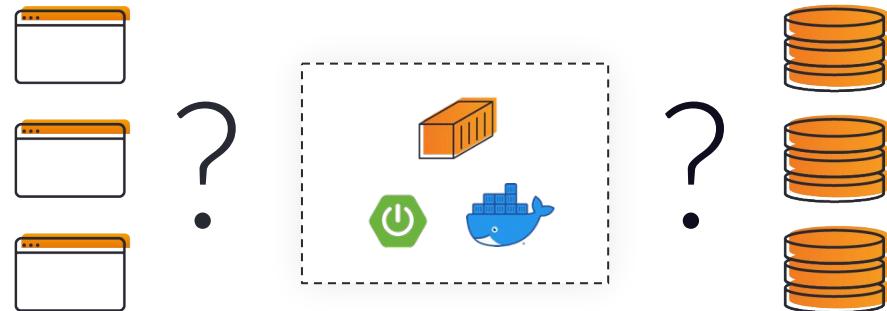
Wydzielenie pierwszego modułu

- Identyfikacja tenanta po stronie zapytania



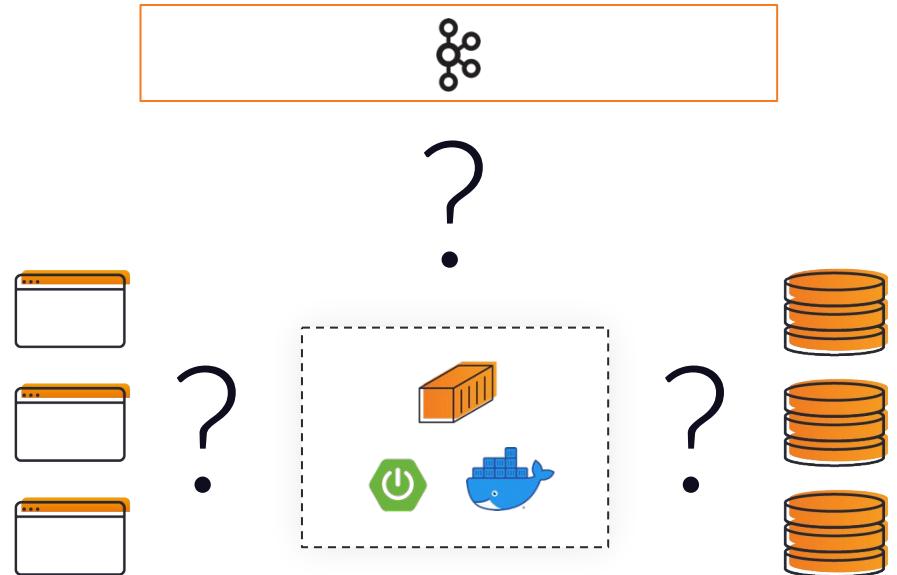
Wydzielenie pierwszego modułu

- Identyfikacja tenanta po stronie zapytania
- Identyfikacja tenanta po stronie bazy danych



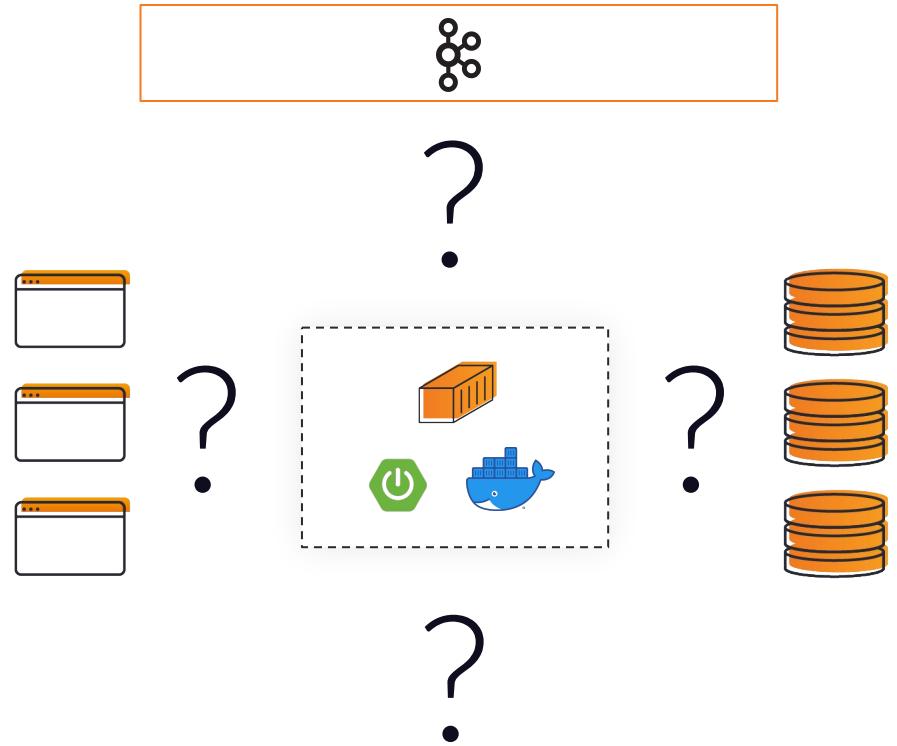
Wydzielenie pierwszego modułu

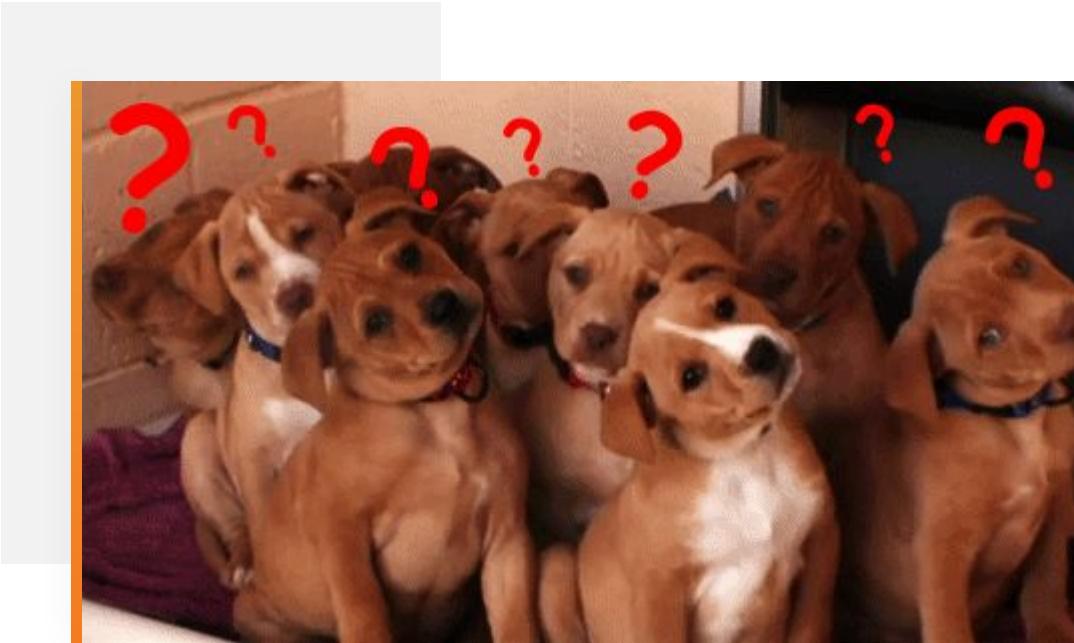
- Identyfikacja tenanta po stronie zapytania
- Identyfikacja tenanta po stronie bazy danych
- Subskrypcja odpowiednich topiców



Wydzielenie pierwszego modułu

- Identyfikacja tenanta po stronie zapytania
- Identyfikacja tenanta po stronie bazy danych
- Subskrypcja odpowiednich topiców
- Pojawienie się nowego tenanta



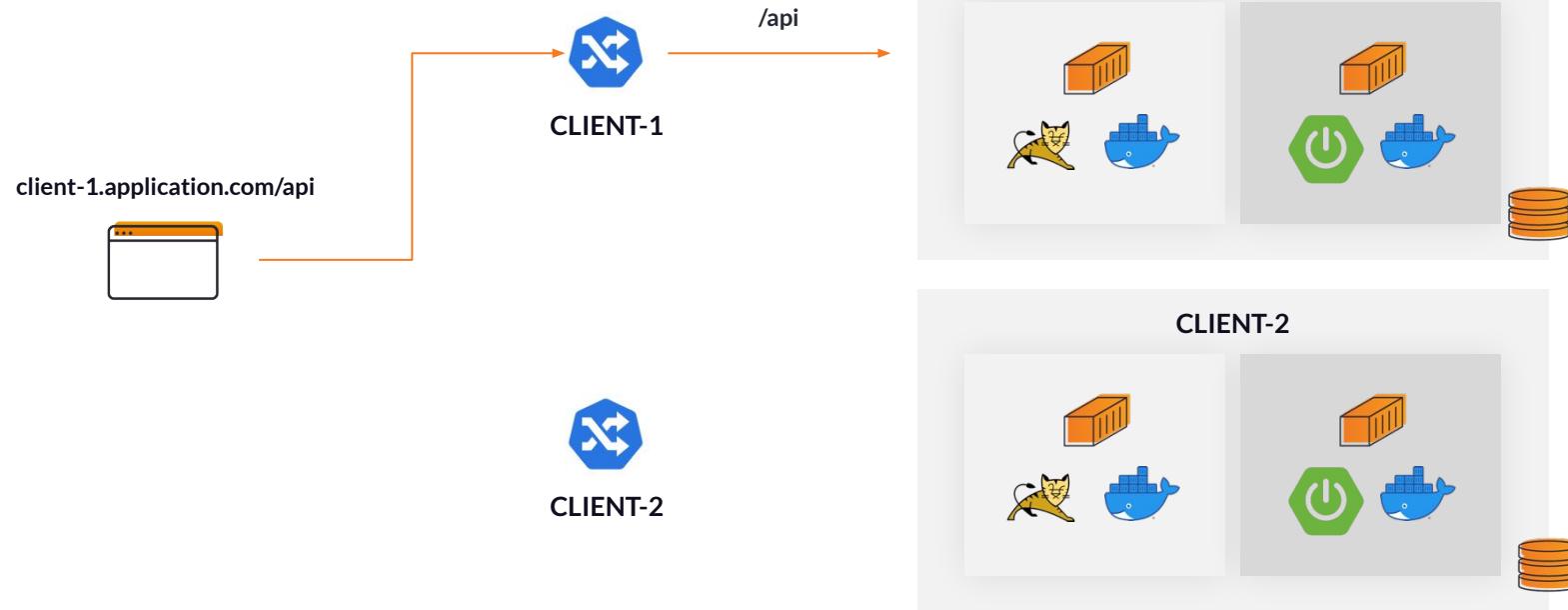


Źródło: <https://giphy.com/gifs/dog-what-confused-7K3p2z8Hh9QOI>

Identyfikacja tenanta po stronie zapytania



Identyfikacja tenanta po stronie zapytania



Identyfikacja tenanta po stronie zapytania



Identyfikacja tenanta po stronie zapytania

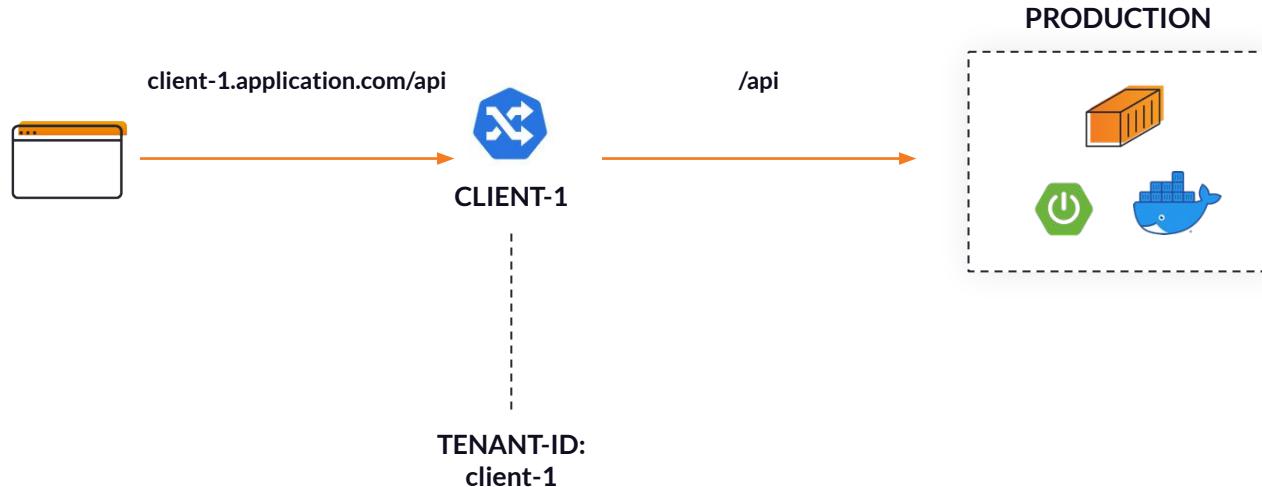
hacker voice

I'm in

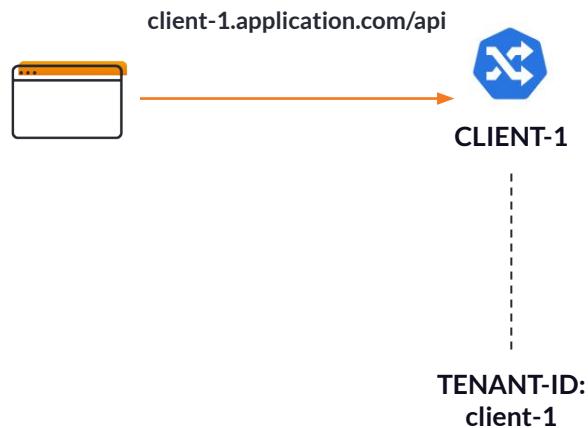


Źródło: <https://damn-funny.net/post/643014387634946048>

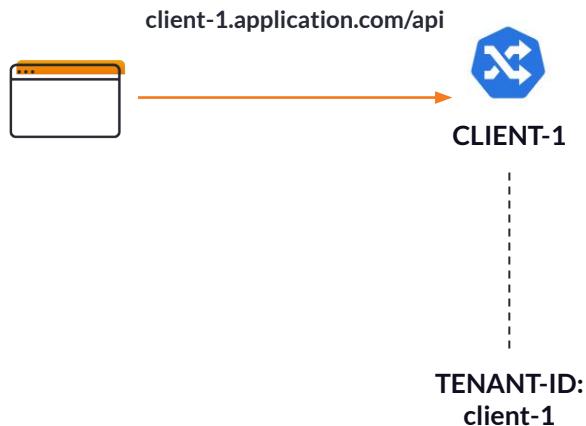
Identyfikacja tenanta po stronie zapytania



Identyfikacja tenanta po stronie zapytania



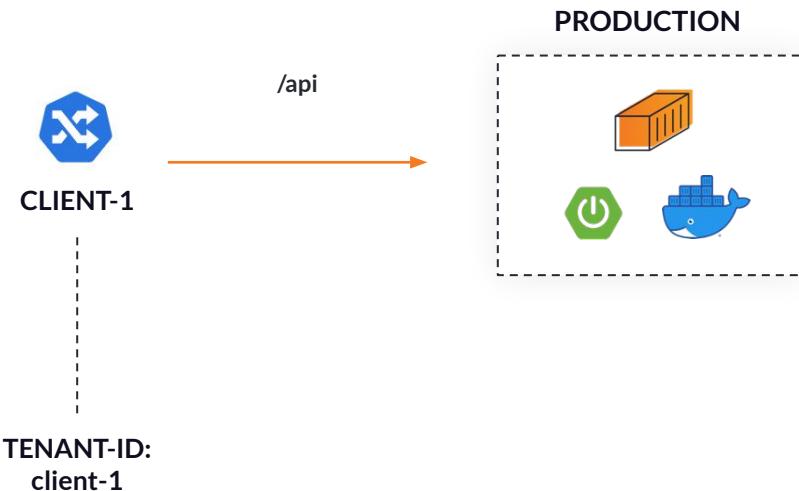
Identyfikacja tenanta po stronie zapytania



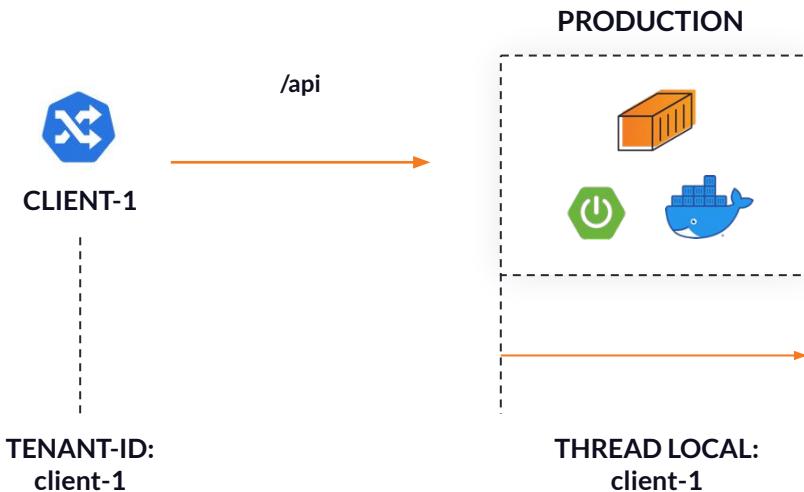
```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    # ...
    nginx.ingress.kubernetes.io/configuration-snippet: |
      proxy_set_header X-Tenant "client-1";
  name: client-1
  namespace: develop
spec:
  rules:
    - host: client-1.helloastr้า.com
      http:
```

`client-1-ingress.yml`

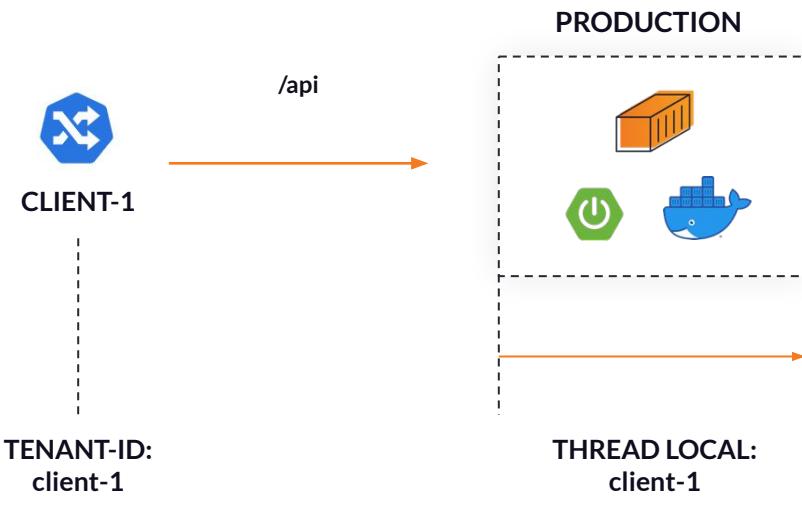
Identyfikacja tenanta w aplikacji



Identyfikacja tenanta w aplikacji

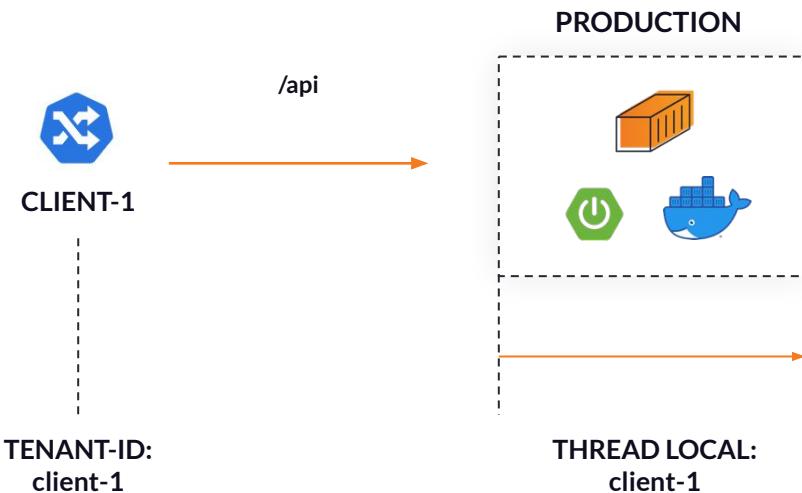


Identyfikacja tenanta w aplikacji



```
public KeycloakDeployment resolve(Request facade) {  
    String tenant = getTenantHeader(facade);  
    UserContextHolder.setCurrentTenant(tenant);  
    return cache.get(tenant, this::loadKeycloakConfig);  
}
```

Identyfikacja tenanta w aplikacji



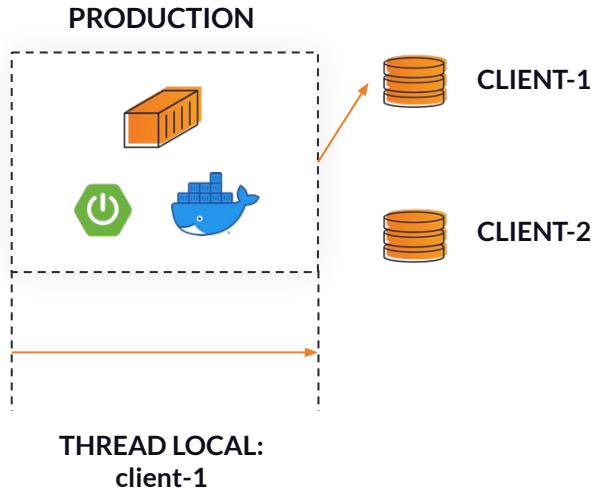
```
public class UserContextHolder {  
  
    private static ThreadLocal<UserContext> userContext; ...  
  
    public static void setCurrentUser(User user) {  
        userContext.get().setUser(user);  
    }  
  
    public static void setCurrentTenant(String tenant) {  
        userContext.get().setTenant(tenant);  
    }  
    // ...  
}
```

User Context Holder

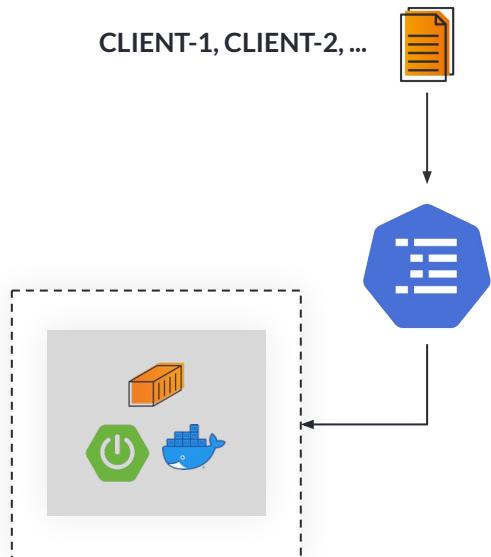
```
public KeycloakDeployment resolve(Request facade) {  
    String tenant = getTenantHeader(facade);  
    UserContextHolder.setCurrentTenant(tenant);  
    return cache.get(tenant, this::loadKeycloakConfig);  
}
```

Multitenant Security

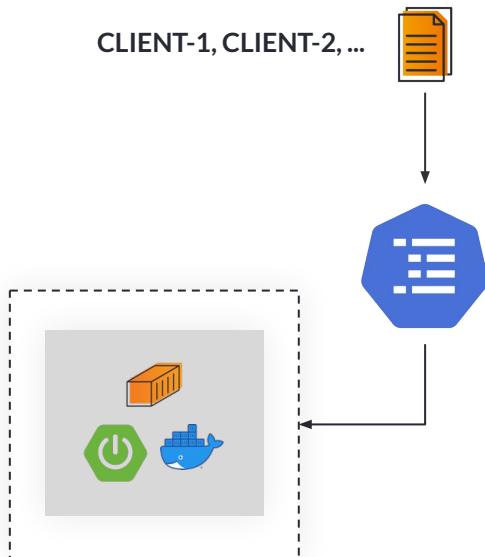
Identyfikacja tenanta po stronie bazy danych



Konfiguracja dla tenanta



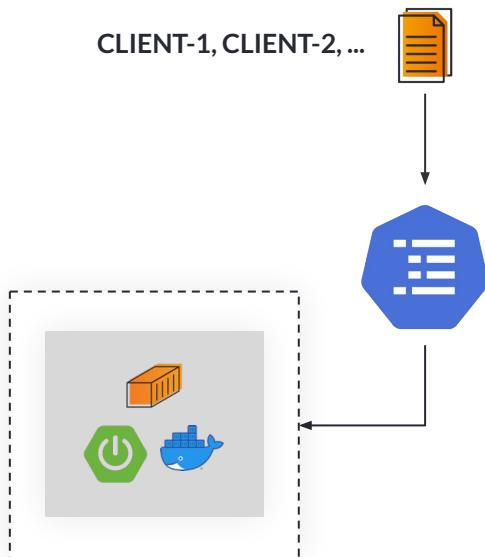
Konfiguracja dla tenanta



```
/data/appconfig # ls -la
total 24
drwxrwxrwx    6 root      root          4096 May 31 12:31 .
drwxr-xr-x    3 root      root          4096 May 31 12:31 ..
lrwxrwxrwx    1 root      root          29 May 31 12:31 application.properties
drwxrwxrwx    3 root      root          4096 May 31 12:31 log4j
drwxrwxrwx    4 root      root          4096 May 31 12:31 properties
```

Container Files

Konfiguracja dla tenanta



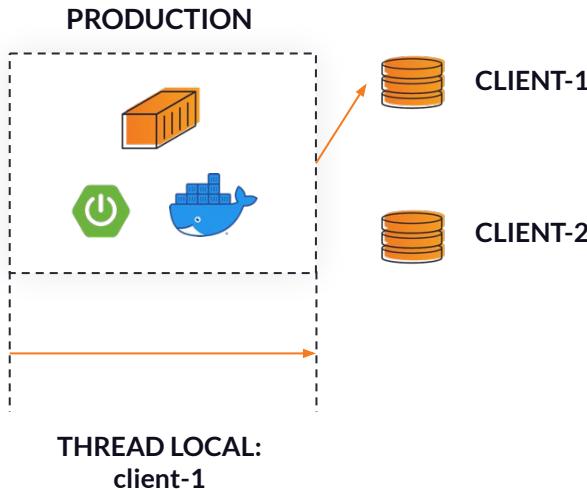
```
/data/appconfig # ls -la
total 24
drwxrwxrwx    6 root      root          4096 May 31 12:31 .
drwxr-xr-x    3 root      root          4096 May 31 12:31 ..
lrwxrwxrwx    1 root      root          29 May 31 12:31 application.properties
drwxrwxrwx    3 root      root          4096 May 31 12:31 log4j
drwxrwxrwx    4 root      root          4096 May 31 12:31 properties
```

Container Files

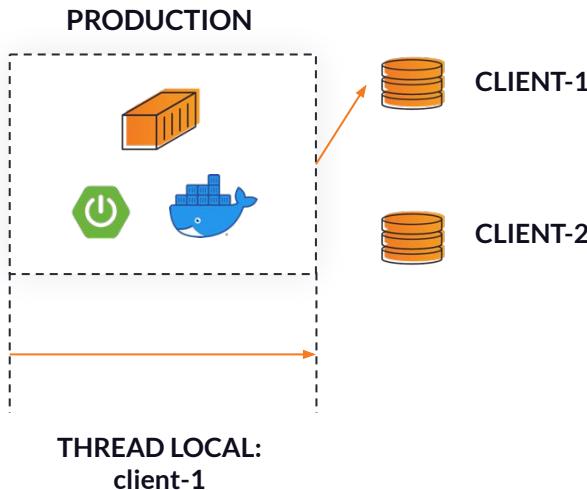
```
@Override
public Option<String> retrieve(@NotNull String propertyName, @NotNull Option<String> tenant) {
    return tenant
        .flatMap(t -> retrievePropertyFromTenant(t, propertyName))
        .orElse(() -> Option.of(environment.getProperty(propertyName)));
}
```

Multitenant Properties Loader

Identyfikacja tenanta po stronie bazy danych



Identyfikacja tenanta po stronie bazy danych



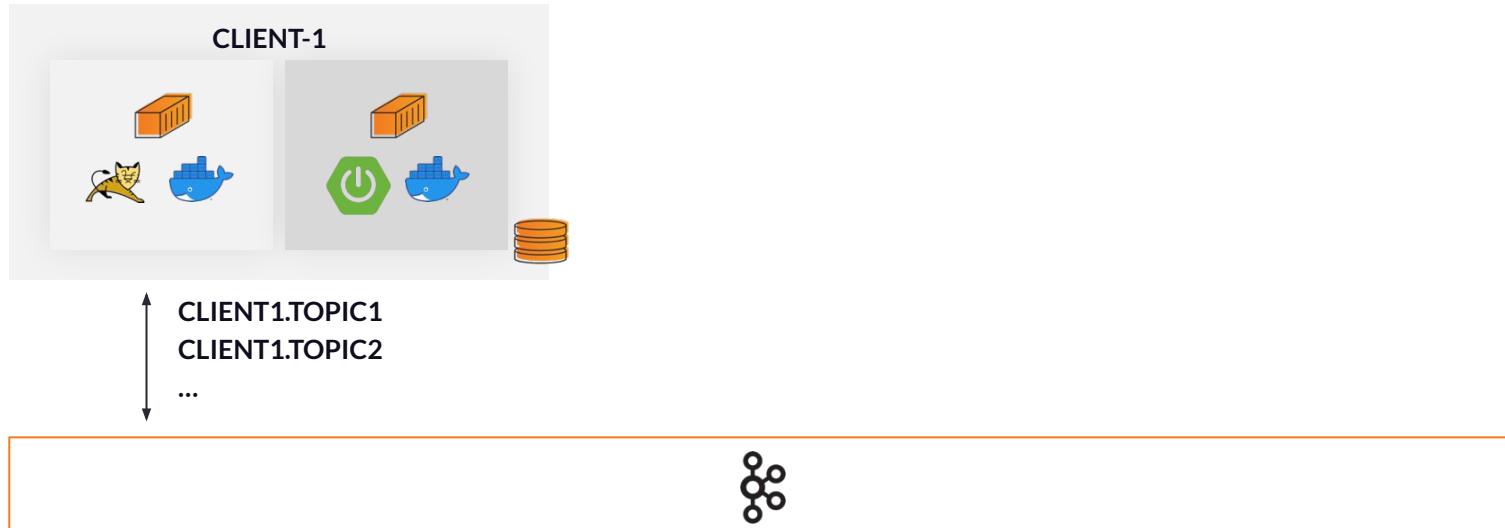
```
MongoCredential credential = buildMongoCredential(mongoConnectionConfigFacade);
List<ServerAddress> serverAddress = buildServerAddresses(mongoConnectionConfigFacade);

MongoClientSettings settings = MongoClientSettings.builder()
    .credential(credential)
    .applyToSslSettings(builder -> builder
        .enabled(sslEnabled)
        .invalidHostNameAllowed(true)
    )
    .applyToClusterSettings(builder -> builder
        .hosts(serverAddress)
    )
    .build();

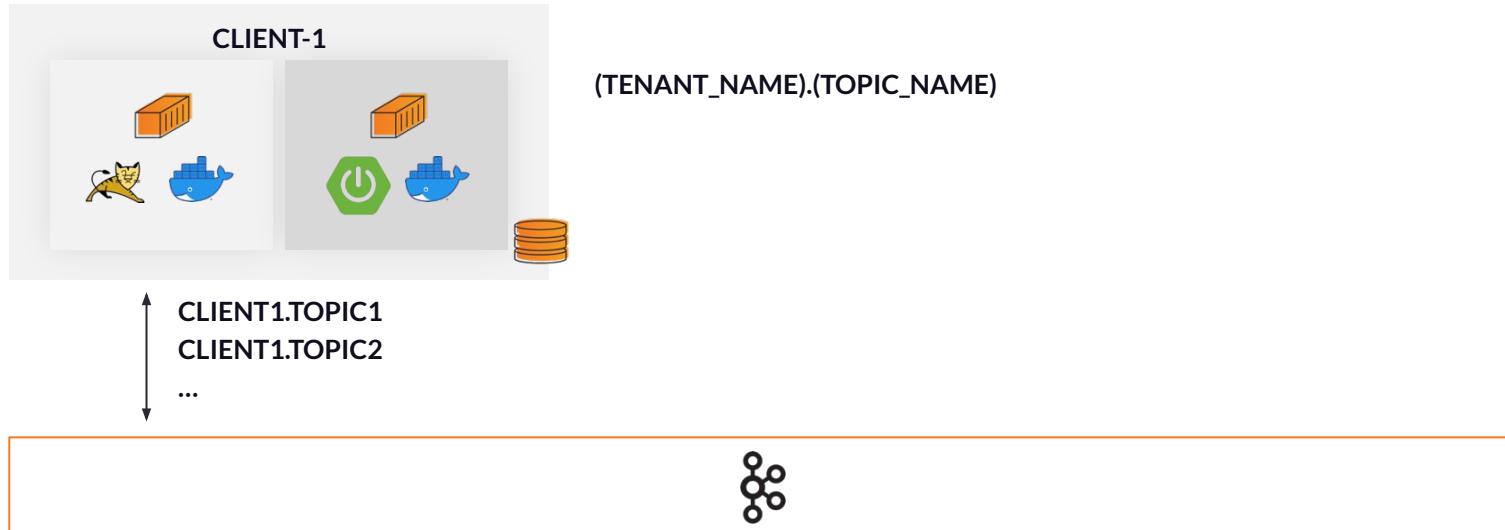
return MongoClients.create(settings);
```

Multitenant DB Connector

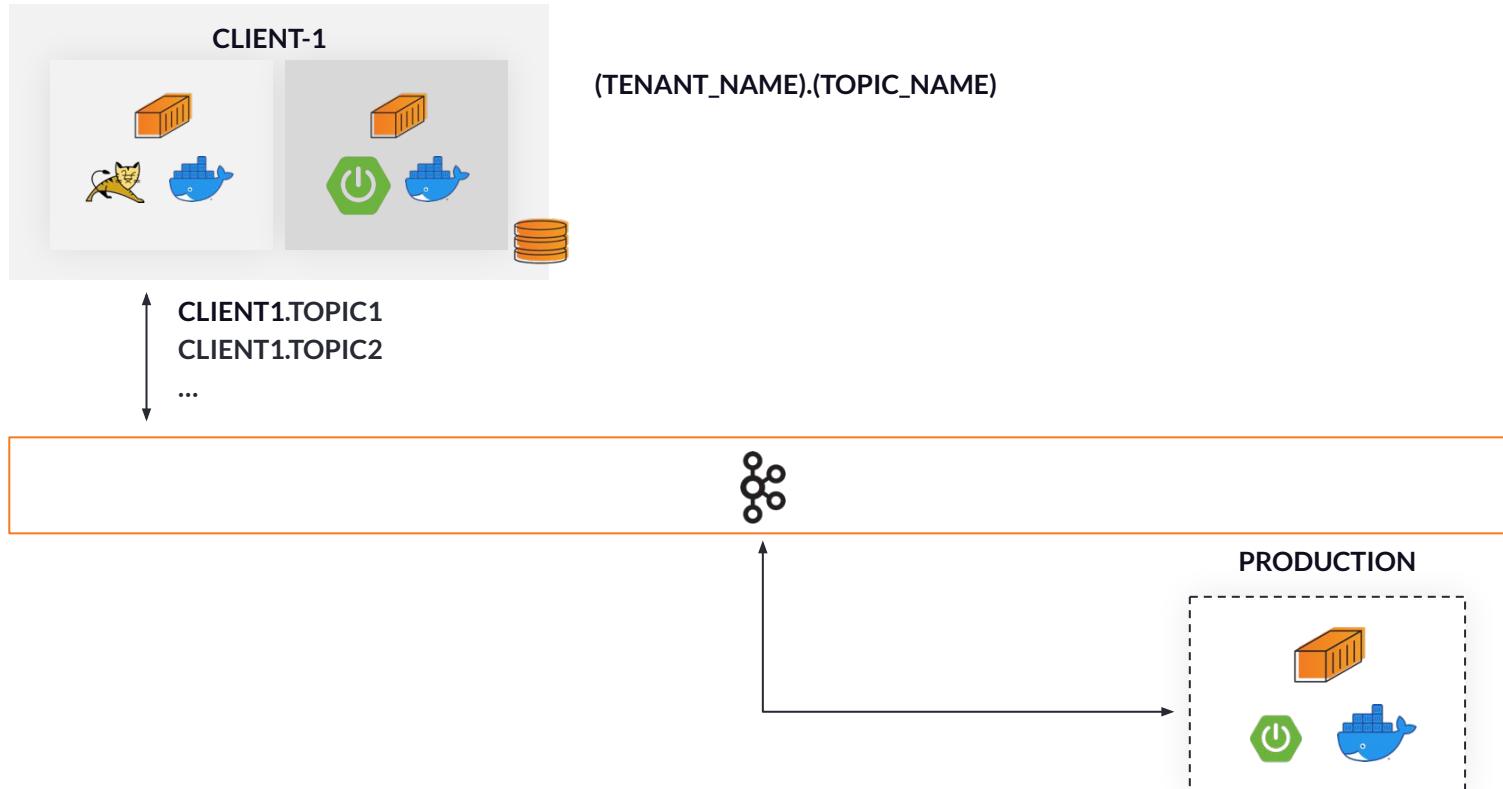
Komunikacja między modułami



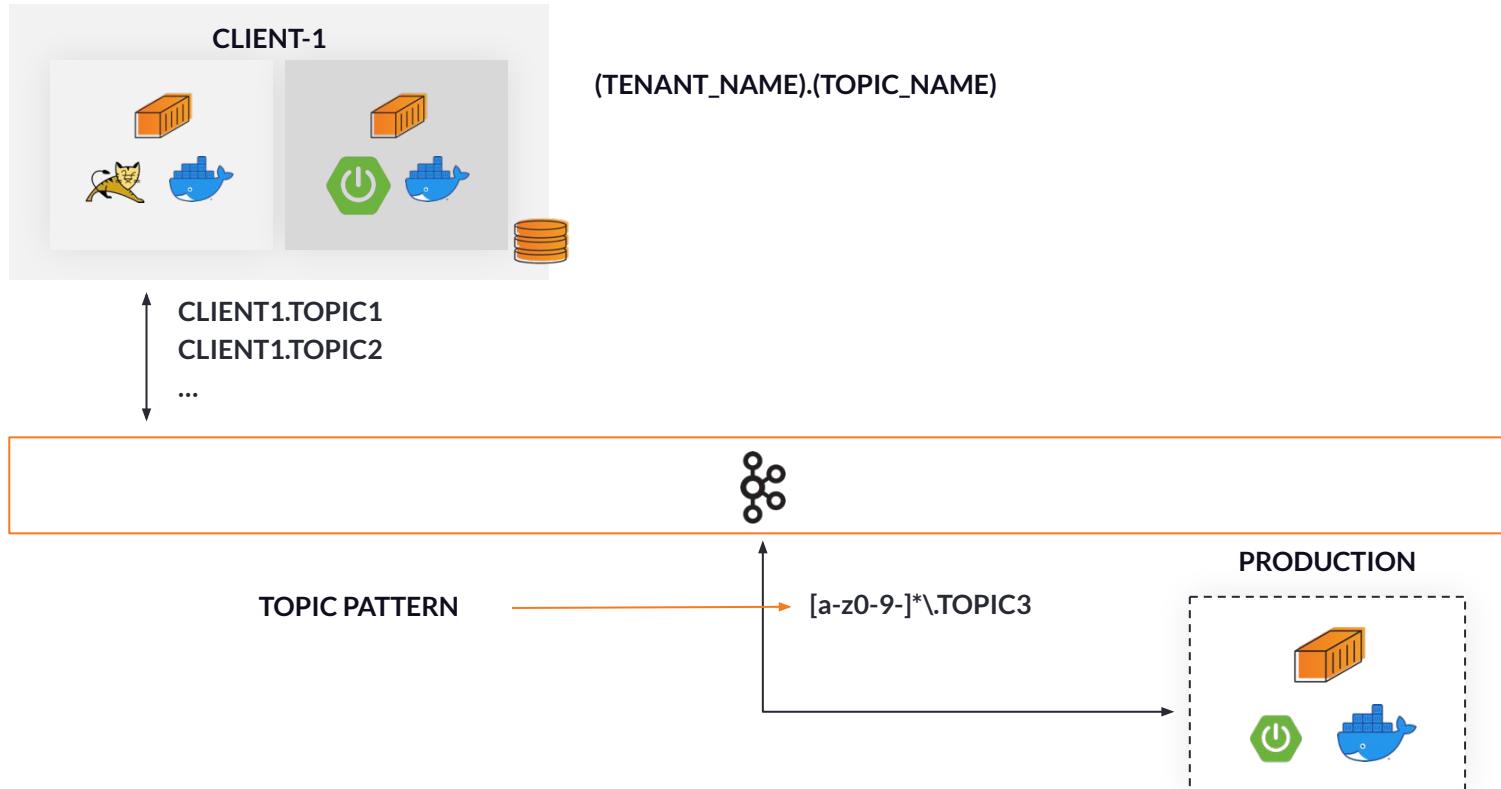
Komunikacja między modułami



Komunikacja między modułami



Komunikacja między modułami



Komunikacja między modułami

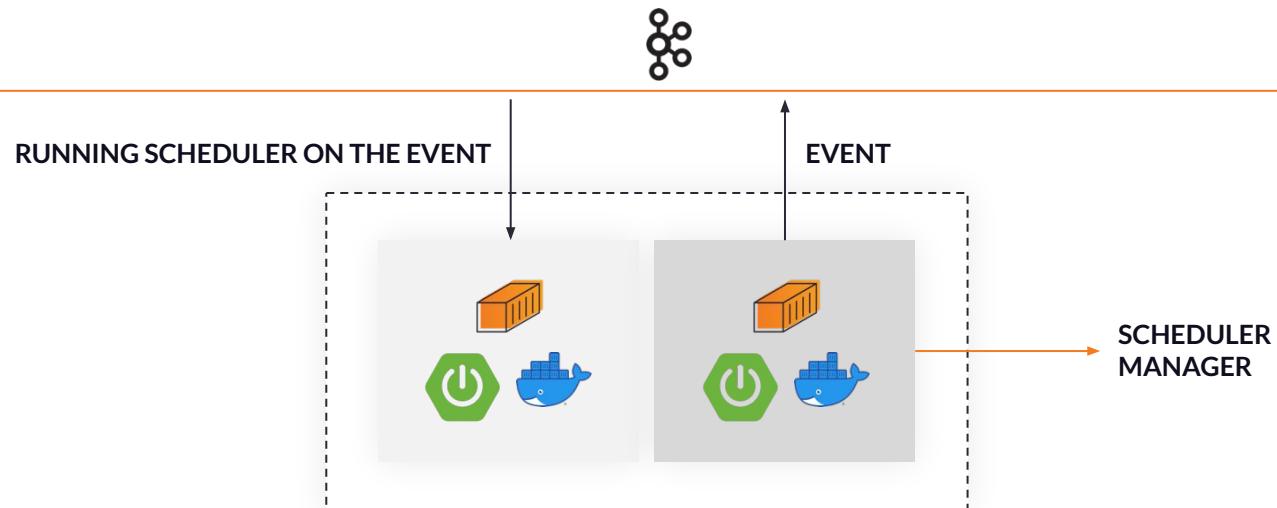


Komunikacja między modułami



```
@Target(ElementType.METHOD)  
@Retention(RetentionPolicy.RUNTIME)  
@EventListener  
public @interface TenantContextEventListener {  
}  
User Context Holder
```

Akcje okresowe – schedulery



Pojawienie się nowego tenanta

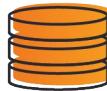


Długi czas stawiania aplikacji dla nowego tenanta



40 minut

DB - NEW TENANT

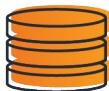


Długi czas stawiania aplikacji dla nowego tenanta



40 minut

DB - NEW TENANT



CONFIGS - NEW TENANT



Długi czas stawiania aplikacji dla nowego tenanta



40 minut

DB - NEW TENANT



CONFIGS - NEW TENANT



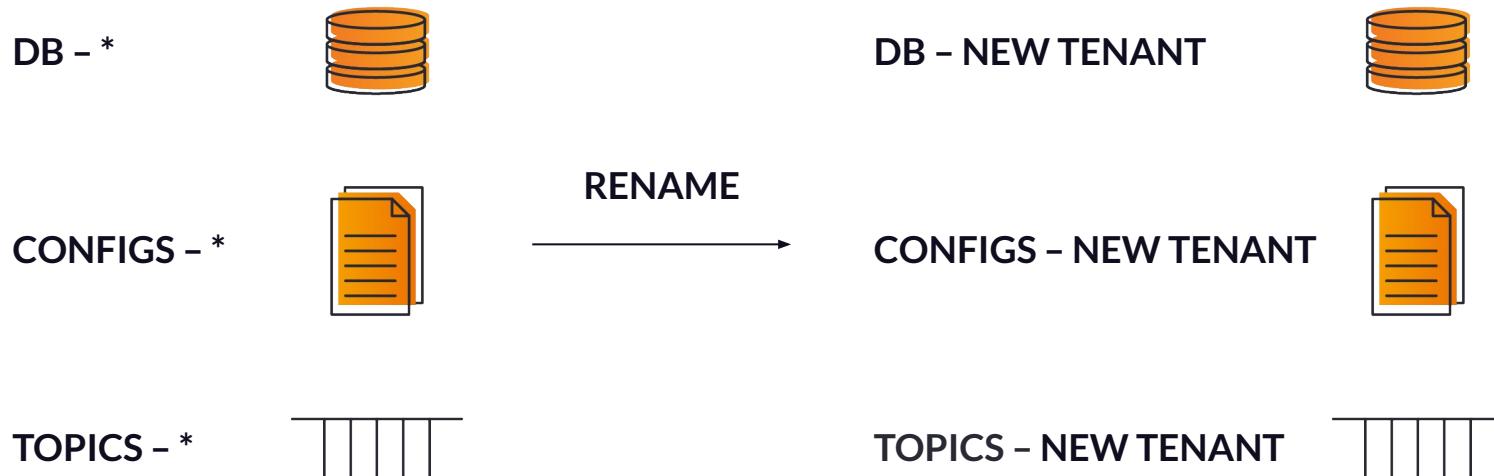
TOPICS - NEW TENANT



Długi czas stawiania aplikacji dla nowego tenanta



40 minut



To nie koniec...



Źródło: <https://giphy.com/gifs/dog-what-confused-7K3p2z8Hh9QOI>

Dziękuję!