Aplicaciones empresariales con los 12 factores cloud native

Víctor Orozco

6 de noviembre de 2019

@tuxtor

NABENIK

NABENIK

¿12 factores?

—

NABENIK

12 factores

- Metodología
- Mejores prácticas
- Manifesto
- https://12factor.net/

NABENIK

12 factores cloud native (Heroku)

Frameworks

- Config
- Backing service
- Disposability

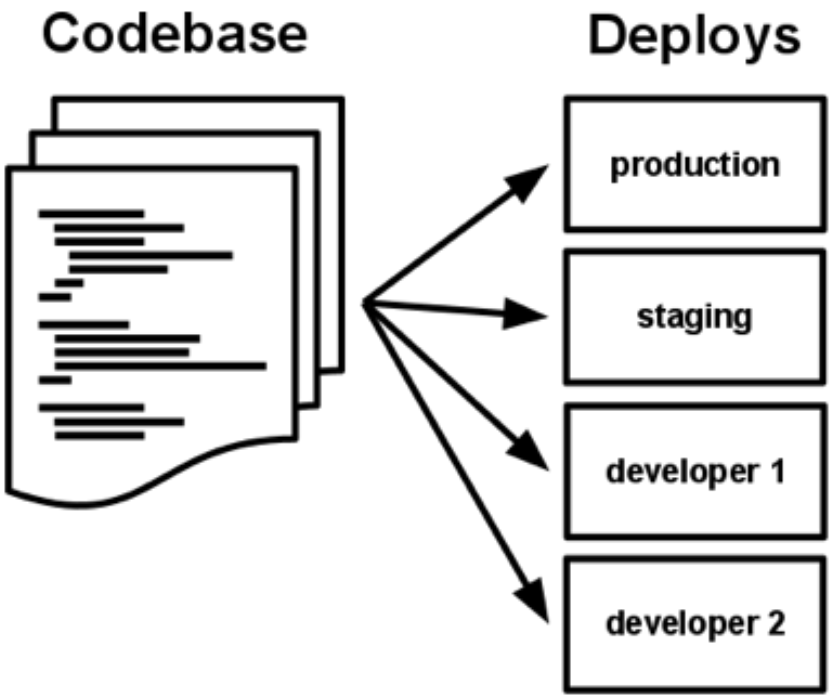
Cloud

- Codebase (Git-Flow)
- Dependencies (Maven)
- Build, Release, Run
- Processes
- Port binding
- Concurrency (Docker - k8s)
- Dev / Prod parity
- Logs
- Admin process

NABENIK

Codebase

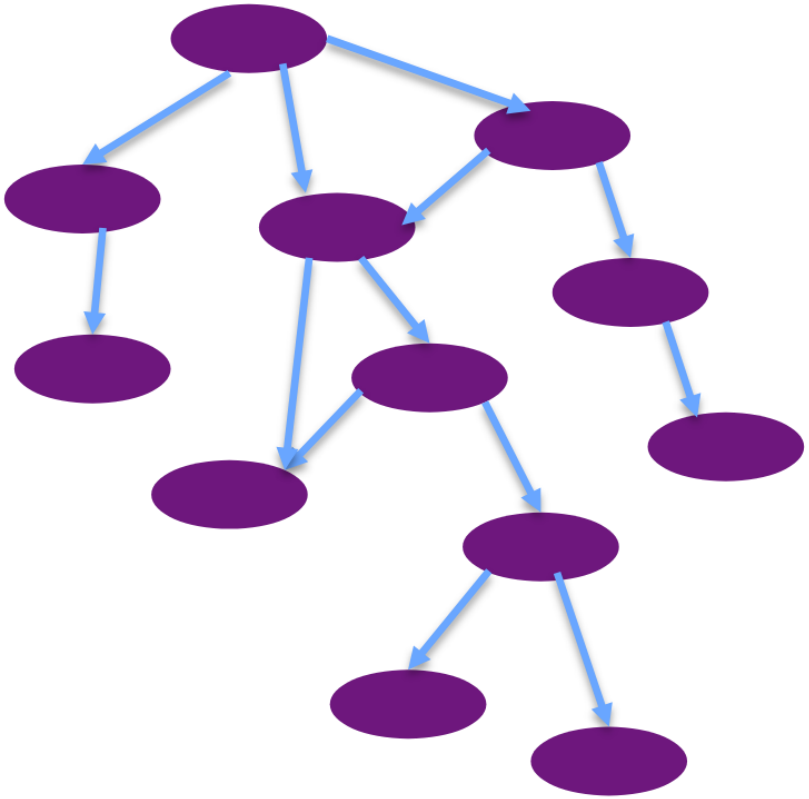
- Una base de código con múltiples entornos de despliegue
- Un repositorio por aplicación / microservicio



NABENIK

Dependencias

- Una aplicación cloud native no "depende" de algo en su entorno
- Dependencias isoladas y compilaciones repetibles



NABENIK

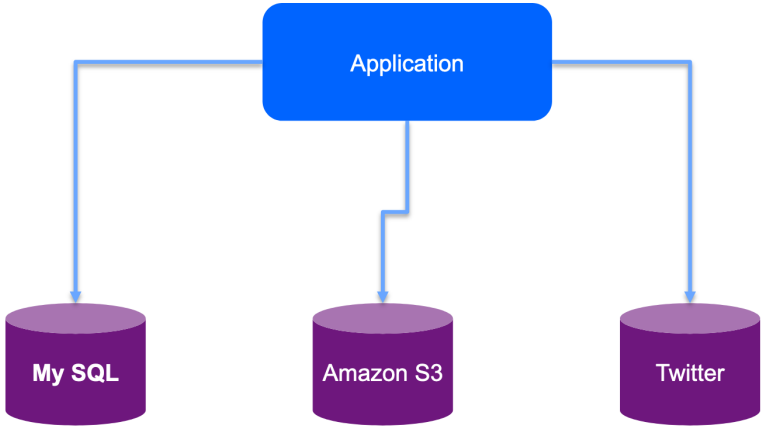
Configuración

- La configuración de una aplicación debe ser dinámica sin re-compilación/re-empaque
- Configuraciones inyectables

NABENIK

Backing services

- Acoplamiento debil. Siempre tratar backing services como componentes intercambiables y/o adjuntos



NABENIK

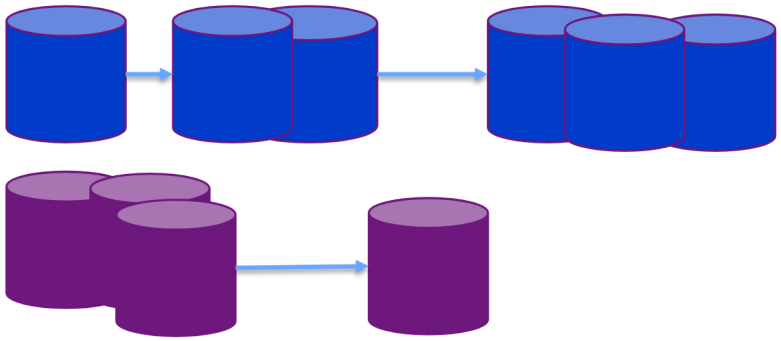
Build, release, run

- Separación de etapas de construcción, ejecución y lanzamiento
- CI/CD se hace obligatorio

NABENIK

Procesos

- Ejecutar la aplicación como uno o más procesos sin estado
- REST, Stateless, sesiones portables con JWT



NABENIK

Port binding

- Exponer los servicios con puertos dinámicos
- Kubernetes, Docker, etc.

NABENIK

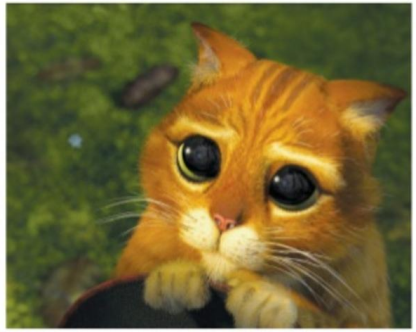
Concurrencia

- Aplicaciones escalan de forma independiente replicándose
- Las nubes escalan mediante copias independientes, sin estado

NABENIK

Disposability

- Procesos arrancar rápido, mueren rápido, reinician rápido
- Procesos son tolerantes a fallas



NABENIK

Dev/Prod parity

- Entornos de desarrollo, certificación, producción lo más homogéneos posible

NABENIK

Logs

- Manipular logs de n copias de n servicios (streams de eventos)
- Permitir el análisis posterior

NABENIK

Admin process

- Manipular logs de n copias de n servicios (streams de eventos)
- Permitir el análisis posterior

NABENIK

12 factores cloud native (Heroku)

Frameworks

- Config
- Backing service
- Disposability

Cloud

- Codebase (Git-Flow)
- Dependencies (Maven)
- Build, Release, Run
- Processes
- Port binding
- Concurrency (Docker - k8s)
- Dev / Prod parity
- Logs
- Admin process

NABENIK

Monolito - Escalabilidad

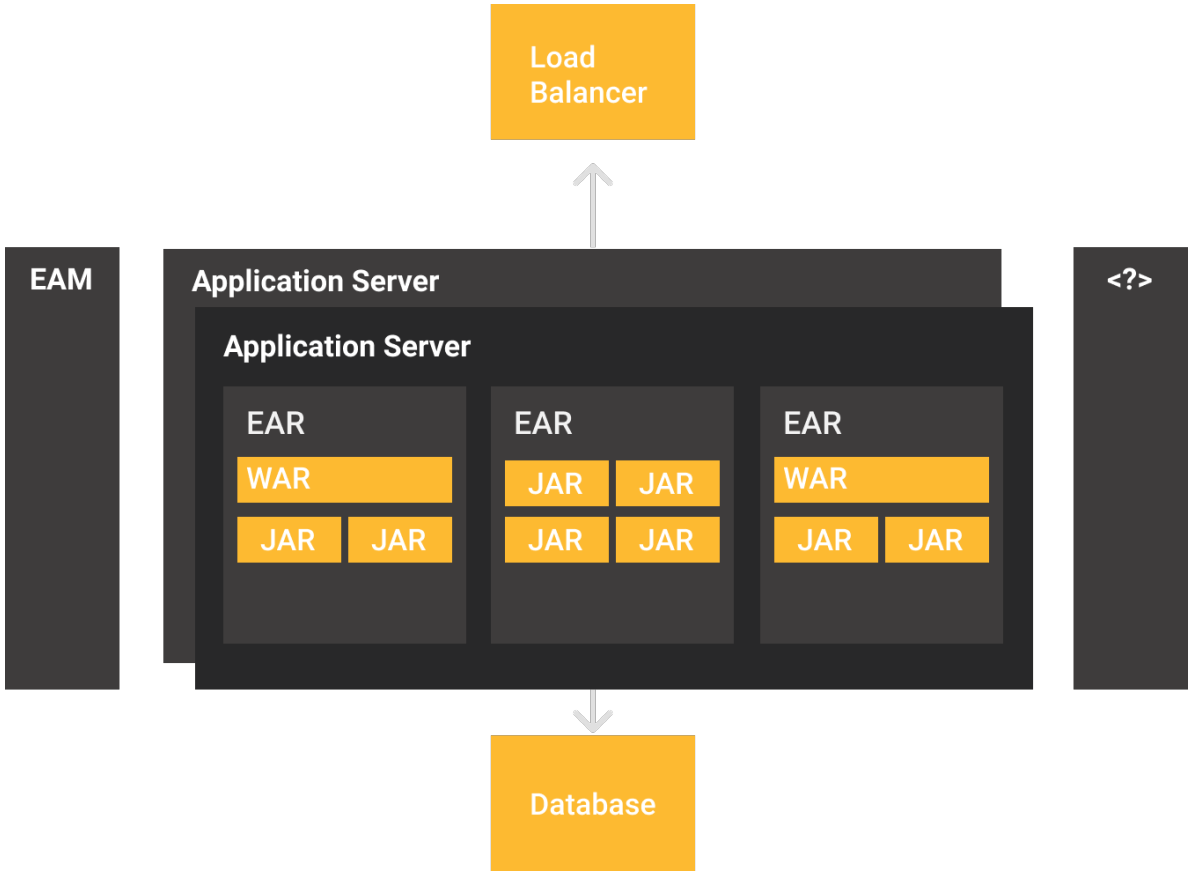


Figura 1: Monólito

NABENIK

Microservicios

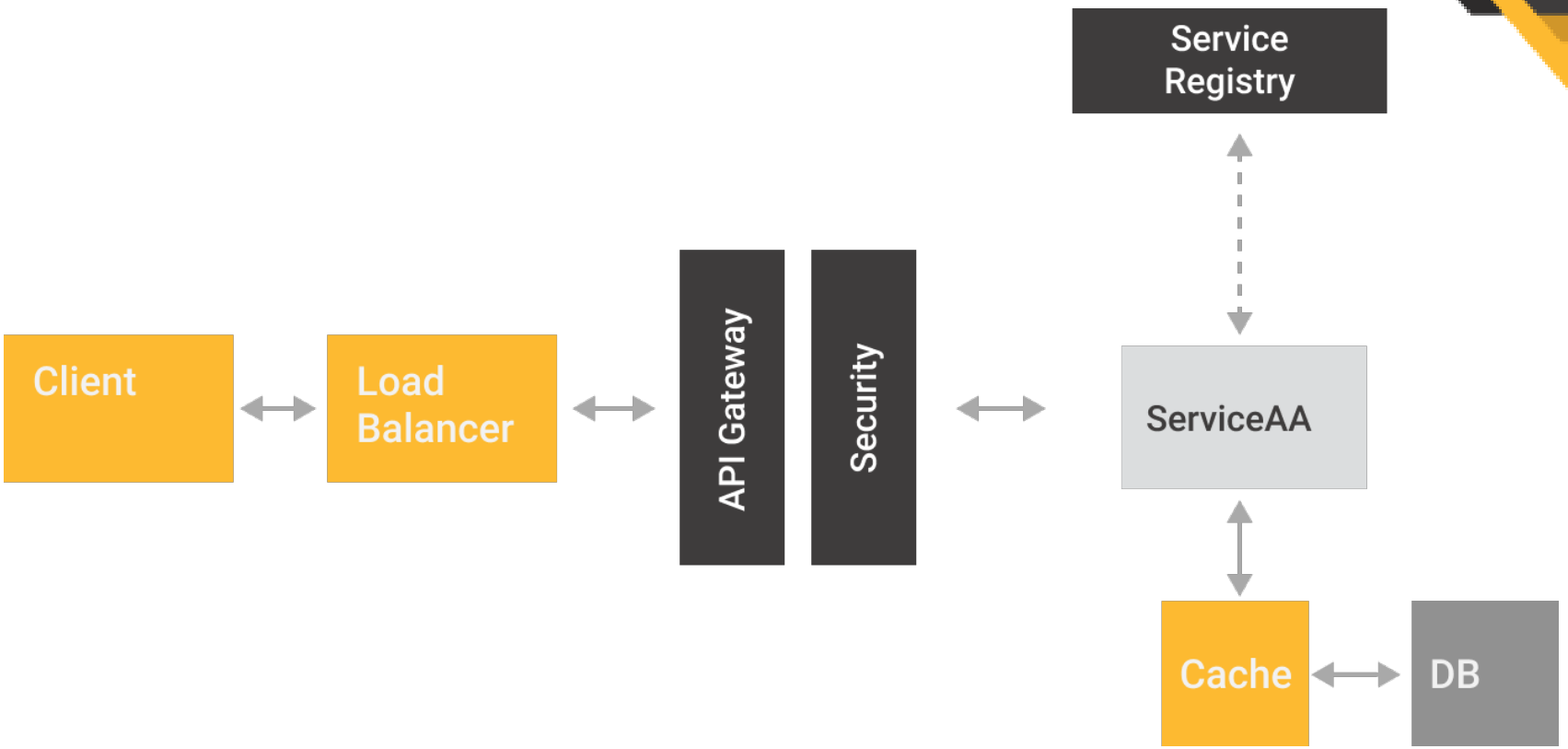


Figura 2: Microservicios

NABENIK

Eclipse MicroProfile

Eclipse MicroProfile

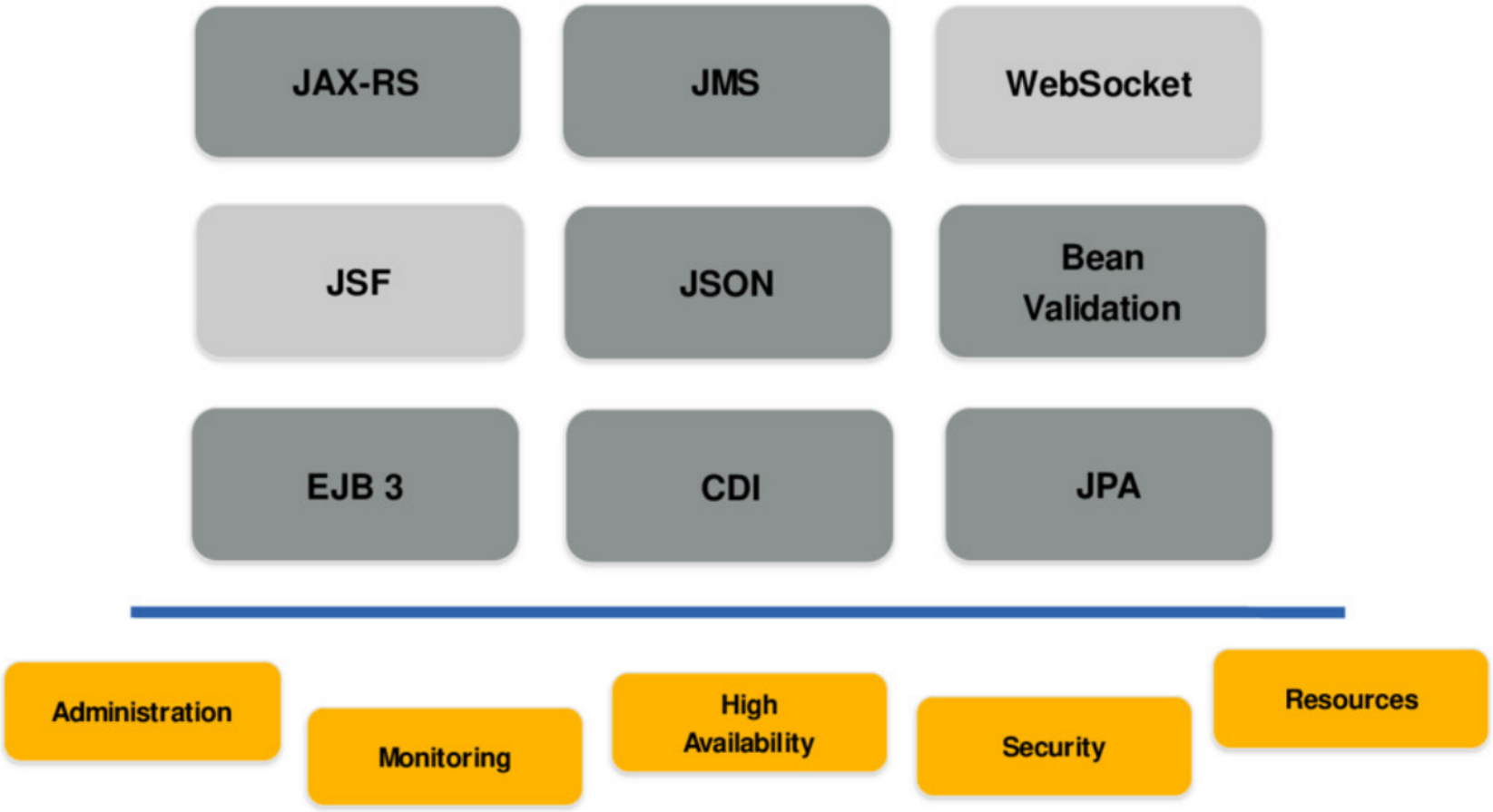
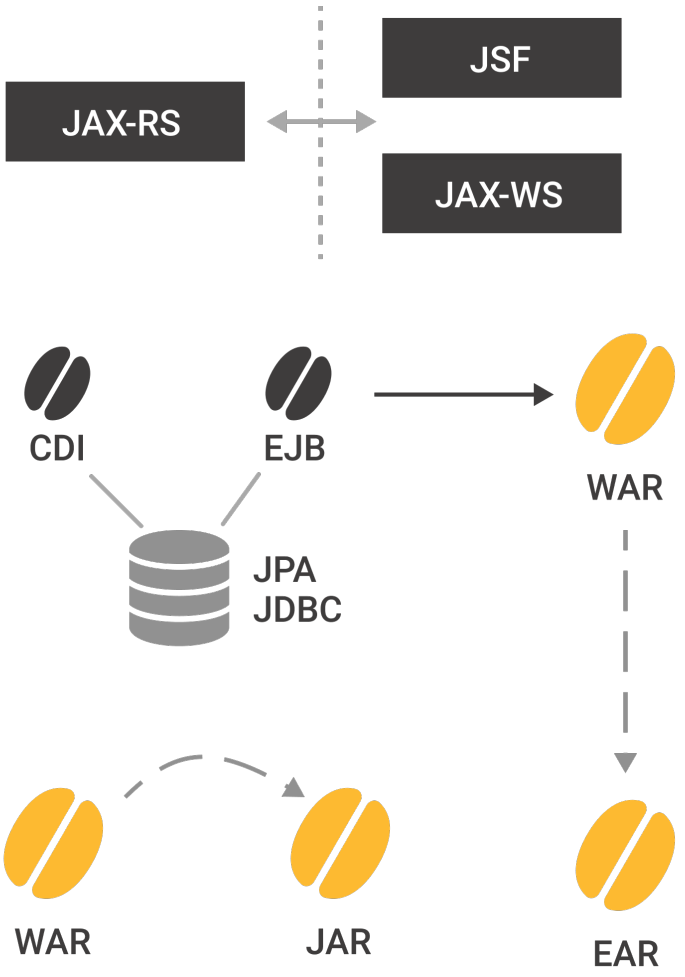


Figura 3: Credito: Reza Rahman

NABENIK

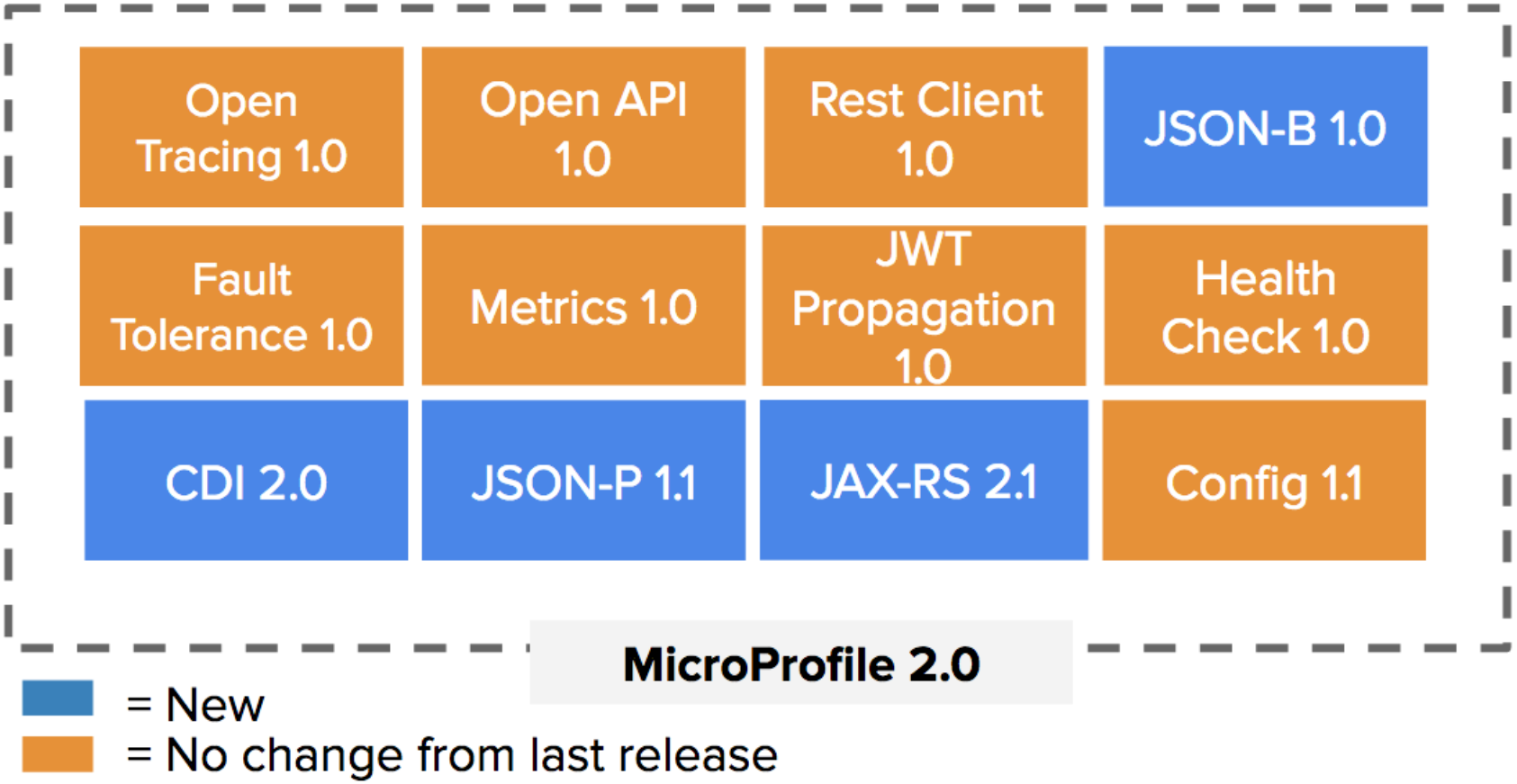
Eclipse MicroProfile



BV + JSON-P JSON-B + Security

NABENIK

Eclipse MicroProfile



NABENIK

Eclipse MicroProfile - Implementaciones

Bibliotecas

- SmallRye (Red Hat)
- Hammock
- Apache Geronimo
- Fujitsu Launcher

JEAS - Fat Jar

- Dropwizard
- KumuluzEE
- Helidon (Oracle)
- Open Liberty (IBM)
- Apache Meecrowave
- Thorntail (Red Hat)
- Quarkus (Red Hat)

https://wiki.eclipse.org/MicroProfile/Implementation

NABENIK

Eclipse MicroProfile - Implementaciones

Micro server

- Payara Micro
- TomEE JAX-RS

Full server

- Payara Application Server
- JBoss Application Server / Wildfly Application Server
- WebSphere Liberty (IBM)

https://wiki.eclipse.org/MicroProfile/Implementation

NABENIK

Eclipse MicroProfile on Payara 5

```
<dependency>
        <groupId>org.eclipse.microprofile</groupId>
        <artifactId>microprofile</artifactId>
        <type>pom</type>
        <version>2.0.1</version>
        <scope>provided</scope>
</dependency>
```

NABENIK

Demo

NABENIK

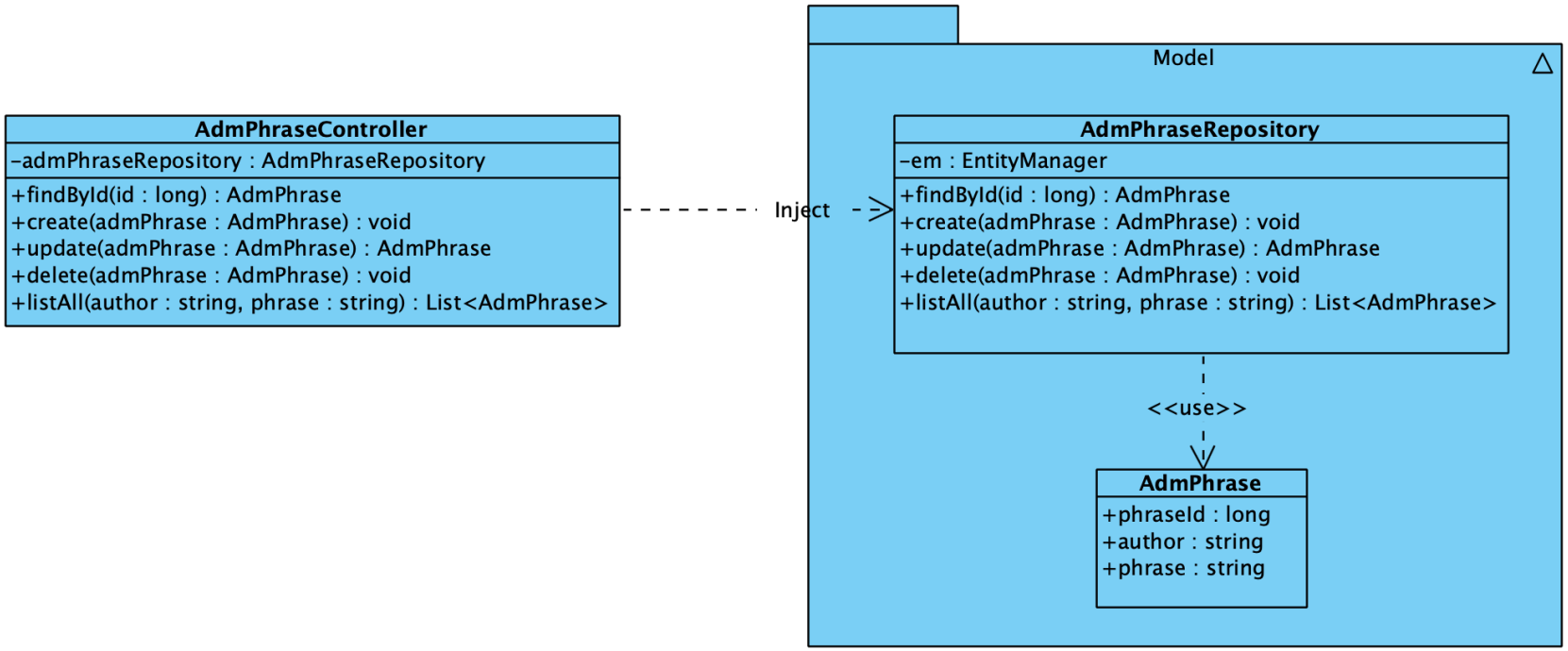
Kotlin + Jakarta EE + MicroProfile - Demo

- Kotlin 1.3
- Libraries - SLF4J, Flyway, PostgreSQL
- Jakarta EE 8 - EJB, JPA
- MicroProfile - CDI, JAX-RS, MicroProfile Config
- Testing - Arquillian, JUnit, Payara Embedded

https://dzone.com/articles/
the-state-of-kotlin-for-jakarta-eemicroprofile-tra
https://github.com/tuxtor/integrum-ee

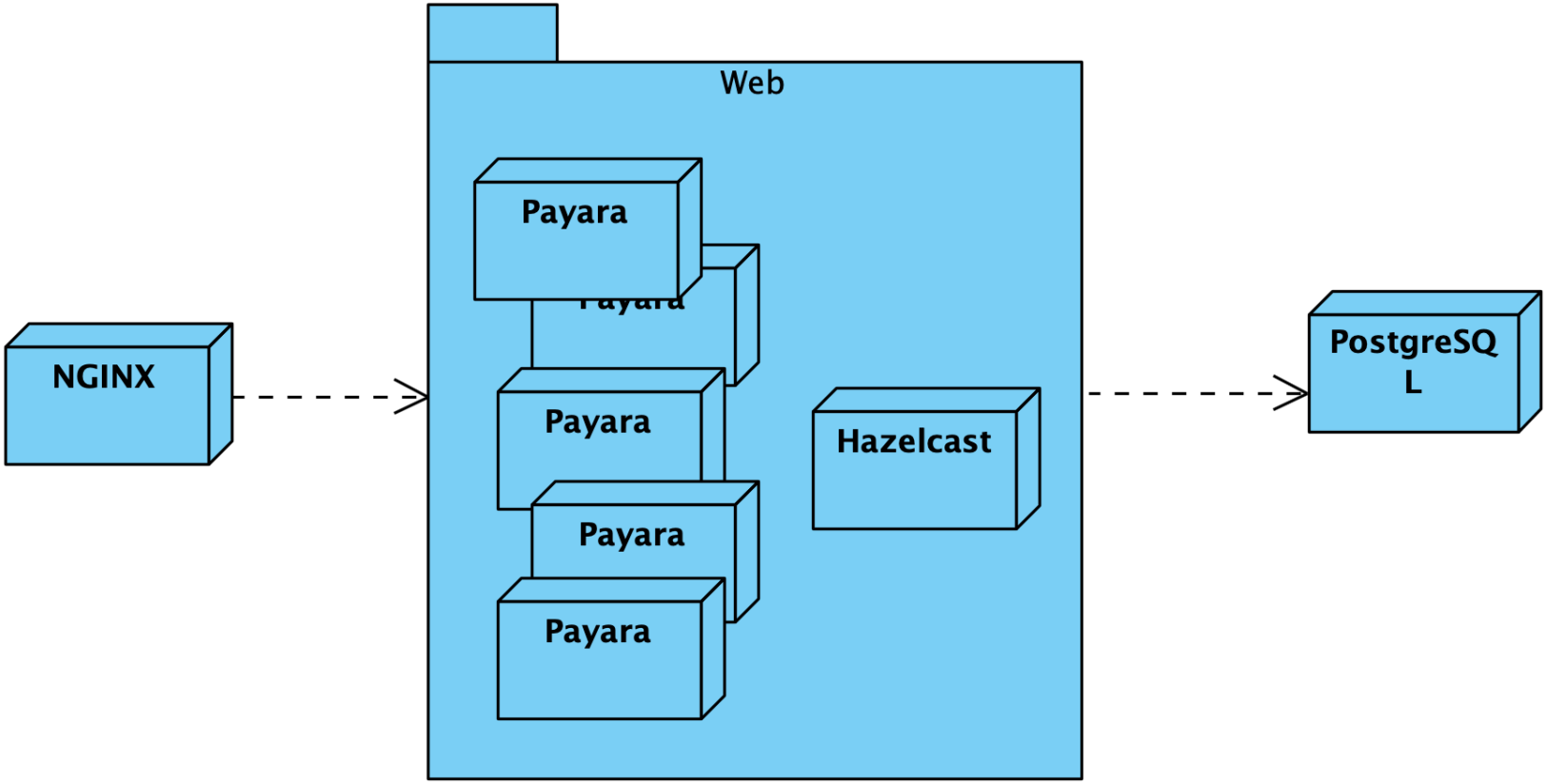
NABENIK

Kotlin + Jakarta EE + MicroProfile - Demo



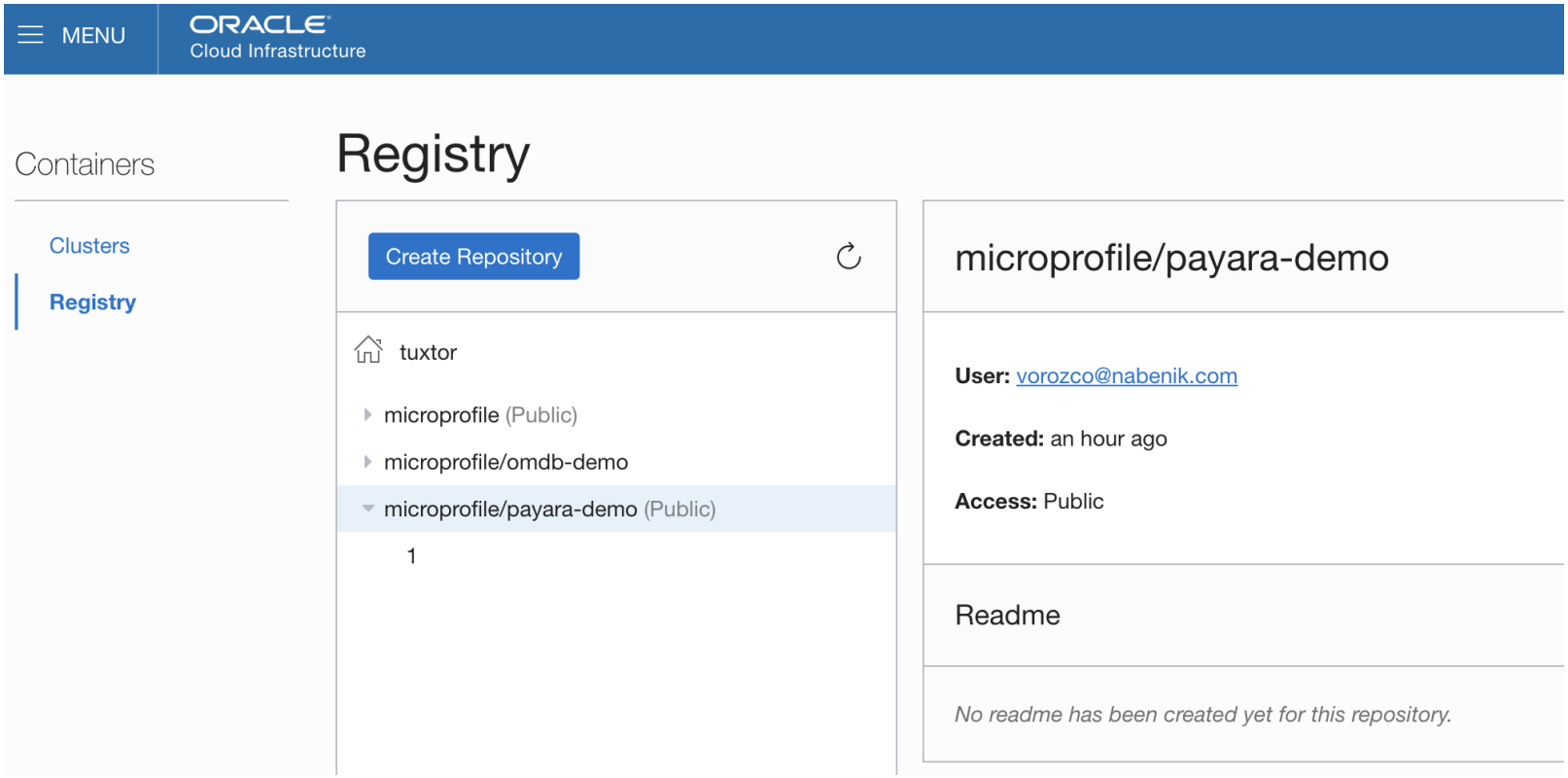
NABENIK

Kotlin + Jakarta EE + MicroProfile - Demo



NABENIK

Oracle Cloud



NABENIK

Oracle Cloud

MENU

ORACLE
Cloud Infrastructure

Compute » Instances » Instance Details

RUNNING

instance-20181206-0243

Create Custom Image | Start | Stop | Reboot | Terminate | Apply Tag(s)

Instance Information | Tags

Instance Information

Availability Domain: hgWe:US-ASHBURN-AD-1

Fault Domain: FAULT-DOMAIN-2

Region: iad

Shape: VM.Standard2.1

Virtual Cloud Network: vcn20181206044411

Maintenance Reboot: -

Primary VNIC Information

NABENIK

Oracle Cloud

```
tuxtor@millenium-falcon-2:~$ docker tag omdb-demo iad.ocir.io/tuxtor/microprofile/omdb-demo:latest
```

```
tuxtor@millenium-falcon-2:~$ docker push iad.ocir.io/tuxtor/microprofile/omdb-demo
The push refers to repository [iad.ocir.io/tuxtor/microprofile/omdb-demo]
31d661ce120c: Pushing [==================================>                ]  952.3kB/1.46MB
5dd1fd455c13: Layer already exists
2aa3decaee68: Layer already exists
685fdd7e6770: Layer already exists
c9b26f41504c: Layer already exists
cd7100a72410: Layer already exists
```

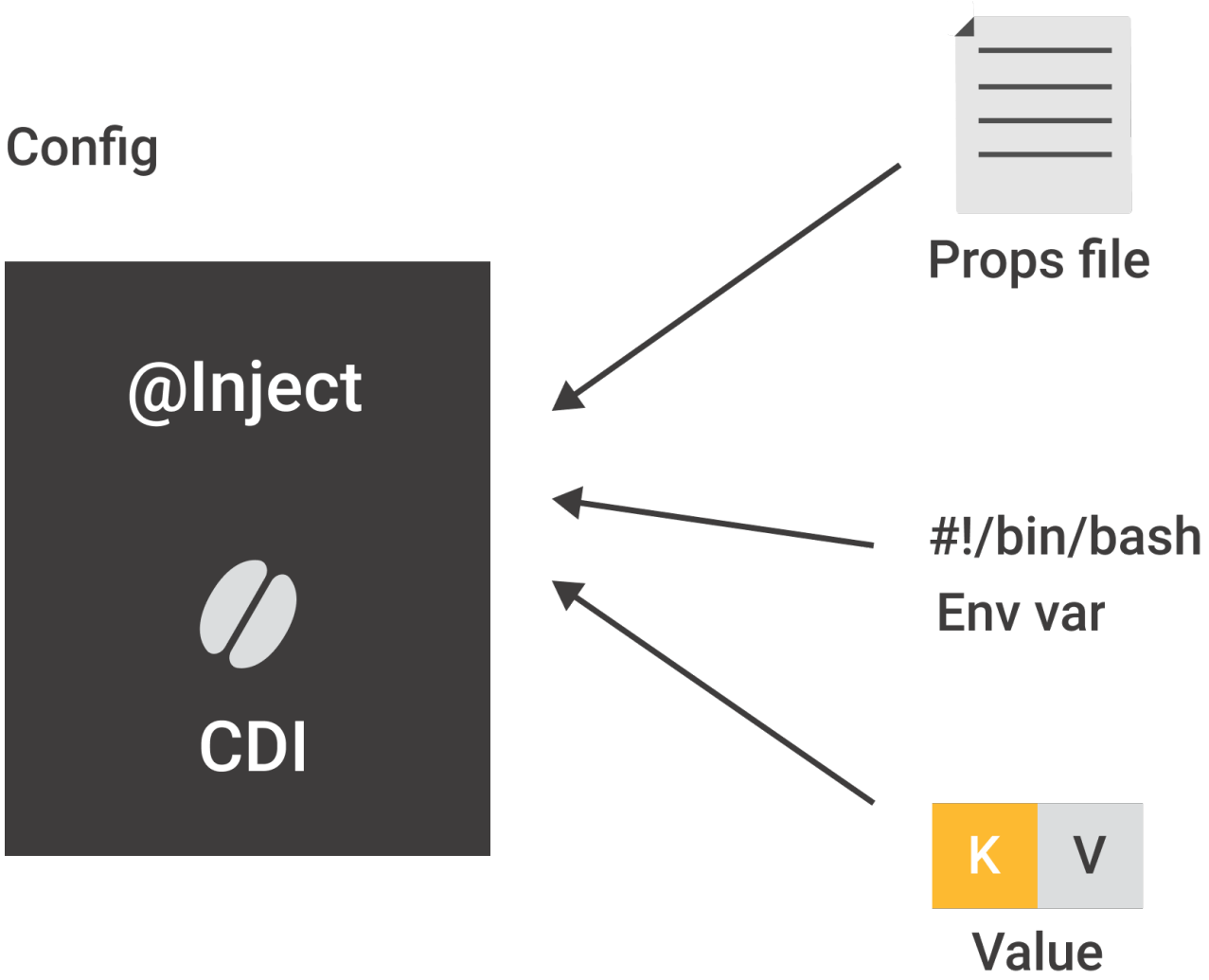
NABENIK

Oracle Cloud

Stateful Rules				
Source: 0.0.0.0/0	**IP Protocol:** TCP	**Source Port Range:** All	**Destination Port Range:** 22	**Allows:** TCP traffic for ports: 22 SSH Remote Login Protocol
Source: 0.0.0.0/0	**IP Protocol:** ICMP	**Type and Code:** 3, 4		**Allows:** ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
Source: 10.0.0.0/16	**IP Protocol:** ICMP	**Type and Code:** 3		**Allows:** ICMP traffic for: 3 Destination Unreachable
Source: 0.0.0.0/0	**IP Protocol:** TCP	**Source Port Range:** All	**Destination Port Range:** 8080-8085	**Allows:** TCP traffic for ports: 8080-8085

NABENIK

Config



NABENIK

Config

```
@Inject
@ConfigProperty(name = .ºmdbservice.url")
String omdbDaemonServiceUrl;
```

Ext. de la configuración (VM, Docker, Kubernetes)

NABENIK

Fault Tolerance

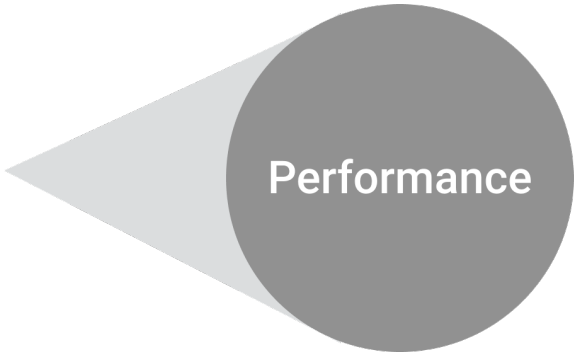
~~a ( )~~ X

aFallback()

NABENIK

Metrics

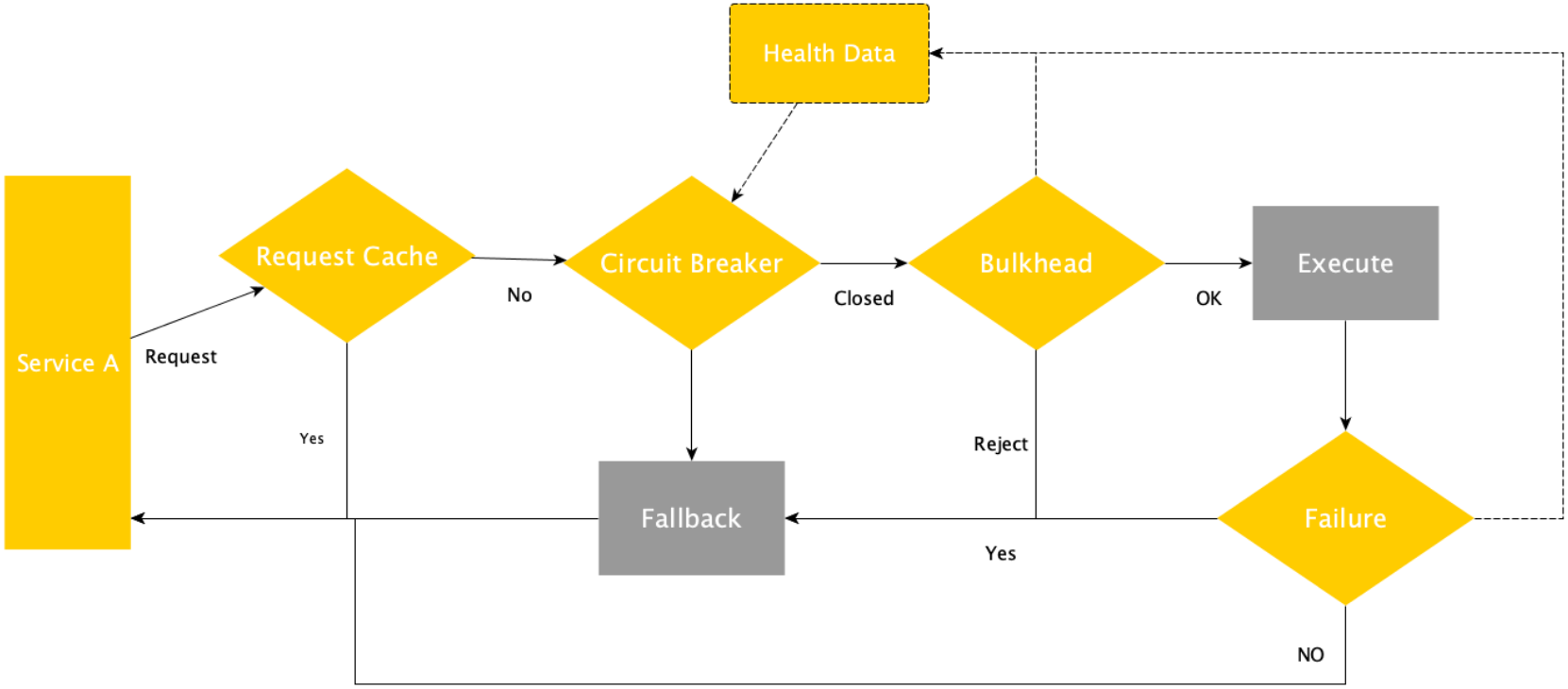
Metrics



NABENIK

Fault Tolerance + Metrics

- *Fault Tolerance* depende de la existencia de metricas, las metricas se exponen mediante *Metrics*



NABENIK

Fault tolerance

Reglas de evaluación y alternativas

- Circuit Breaker
- Bulkhead
- Retry
- Timeout
- Fallback

NABENIK

Fault tolerance - Fallback, Timeout

```
@GET
@Path("/{id:[a-z]*[0-9][0-9]*}")
@Fallback(fallbackMethod = "findByIdFallBack")
@Timeout(TIMEOUT)
public Response findById(@PathParam("id")
final String imdbId) {
...
}

public Response findByIdFallBack(@PathParam("id")
final String imdbId) {
...
}
```

NABENIK

Métricas

- JSON or OpenMetrics (Prometheus)
- Vendor
- Base
- Application

¿Cuales?

- Counted
- Gauge
- Metered
- Timed
- Histogram

NABENIK

Metrics - Counted

```
@Inject
@Metric
Counter failedQueries;

@GET
@Path("/{id:[a-z]*[0-9][0-9]*}")
@Fallback(fallbackMethod = "findByIdFallBack")
@Timeout(TIMEOUT)
public Response findById(@PathParam("id")
final String imdbId) {
...
}

public Response findByIdFallBack(@PathParam("id")
final String imdbId) {
        ...
        failedQueries.inc();
}
```

NABENIK

Metrics - Gauge

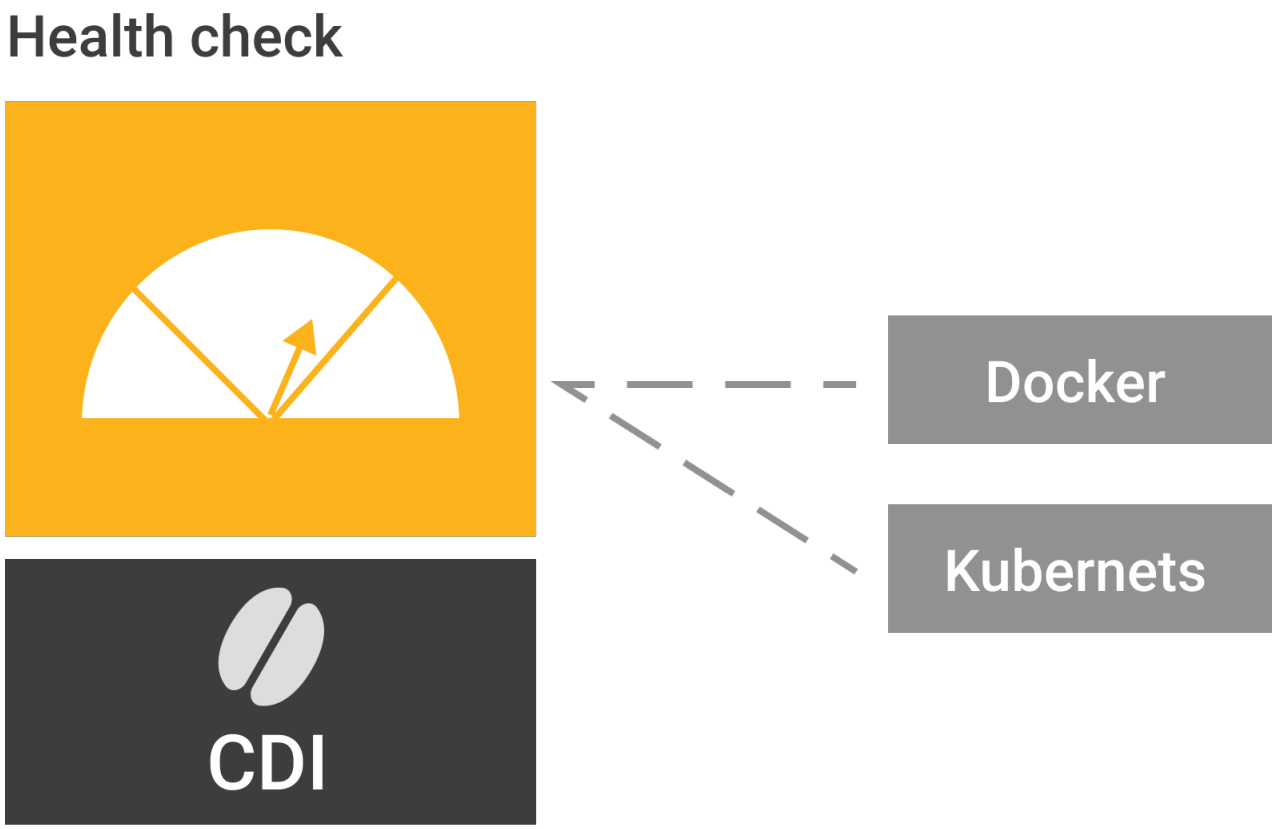
Inc-dec en tiempo real

```
@Gauge(unit = .ExternalDatabases",name = "movieDatabases", absolute
= true)
public long getDatabases() {
        return 99; //Any value
}
```

/metrics/application/movieDatabases

NABENIK

Health Check



- OK?
- How much ok?

NABENIK

Health Check

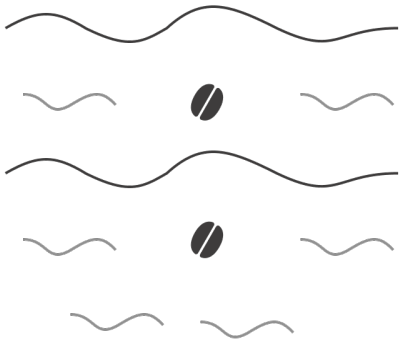
¿Estas vivo?

```
@Override
public HealthCheckResponse call() {
        return HealthCheckResponse.named("TaVivoAinda")
                .withData("key1", "val1")
                .withData("key2", "val2")
                .up()
                .build();

}
```

NABENIK

JWT

JWT



@Inject
Principal

@Inject
Realm

NABENIK

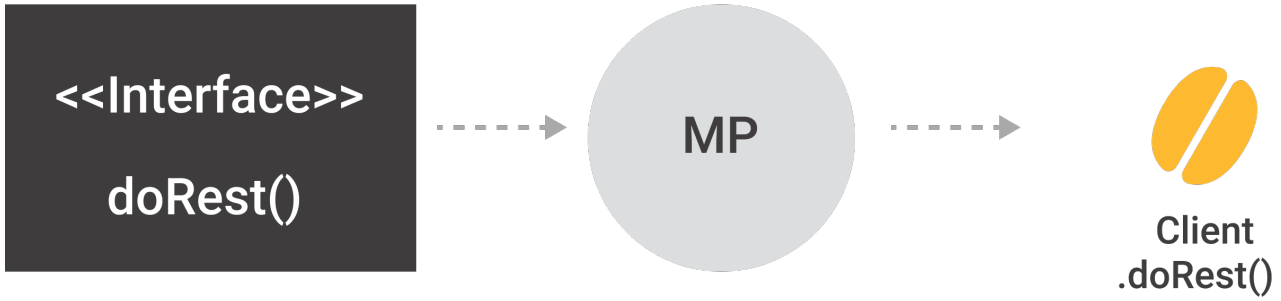
JWT

```
@LoginConfig(authMethod = "MP-JWT")
public class ApplicationConfig extends Application {
}

@Inject
private JsonWebToken jwtPrincipal;

@Inject
@Claim(.email")
private String email;
```

NABENIK

TypeSafe

Type Safe



NABENIK

TypeSafe

```
@Path("/playlist")
@Consumes("application/json")
public interface MusicPlaylistService {

        @GET
        List<String> getPlaylistNames();


        @PUT
        @Path("/{playlistName}")
        long updatePlayList(@PathParam("playlistName")
                String name,
                List<Song> playlist)
                throws UnknownPlaylistException;
}
```

NABENIK

Víctor Orozco

- vorozco@nabenik.com
- @tuxtor
- http://www.nabenik.com

NABENIK