
kumuluzEE

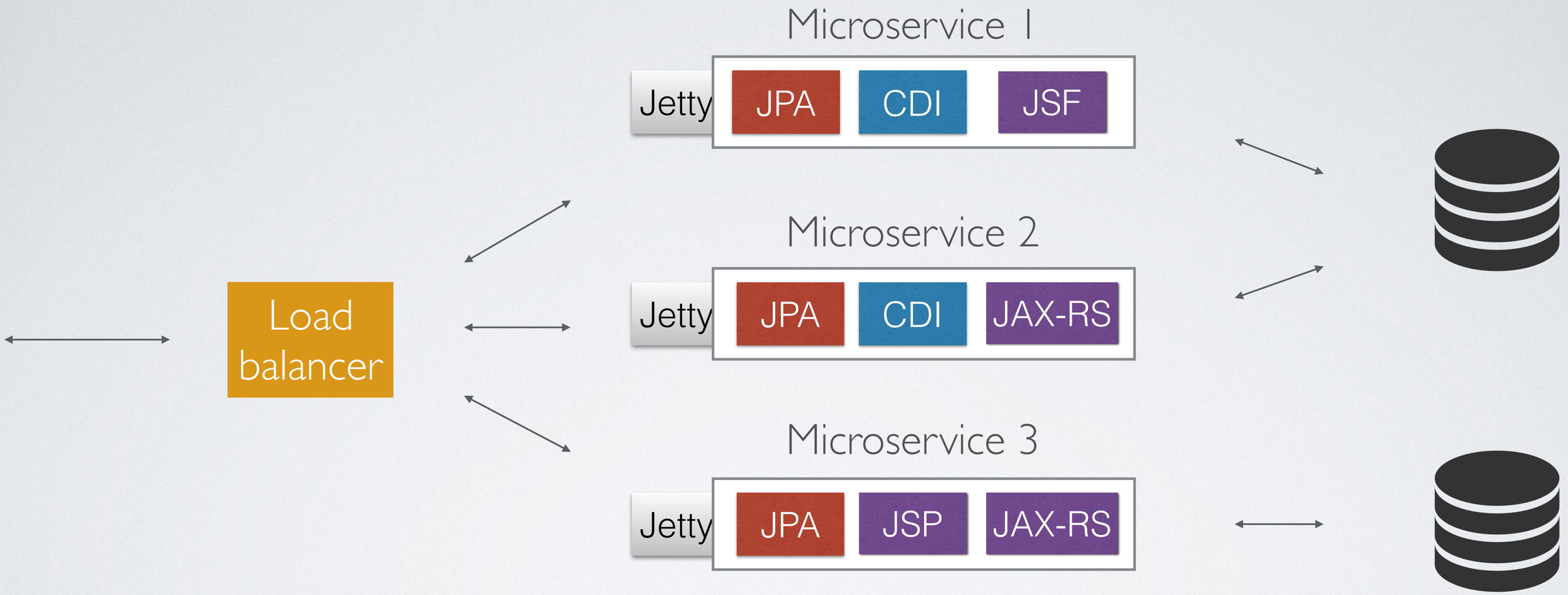
KUMULUZEE

- Lightweight open source framework for developing microservices using Java EE
- Automises configuration and deployment
- Uses standard Java EE APIs
- Produces standalone and independent JARs that run anywhere
- Completely modular and easily extendible

GREAT FIT FOR MICROSERVICES

- Dependency driven - simple to use
- Pick and choose the Java EE components you want
  - Even their implementations
- Final packages (JARs) include only what they need - lightweight
- Ideal for running in PaaS and Docker-like environments

- Support for multiple containers. Choose the one you want
  - Currently Jetty - Tomcat, Undertow, Grizzly planned
- Many Java EE components:
  - Servlet, JSP, EL, CDI, JPA, JAX-RS, Bean Validation, JSON-P, JSF
  - soon to follow: JMS, EJB, JAX-WS



Each microservice can have different components, implementations and versions

SCALING AND DOCKER

- Works nicely with docker as it only requires Java SE
- Microservices act like normal applications/processes
- Simply run the produced JAR or build the microservice in the container and run it directly

Add the required dependencies (only add what you need)

```
<dependency>
    <groupId>com.kumuluz.ee</groupId>
    <artifactId>kumuluzee-core</artifactId>
    <version>2.0.0</version>
</dependency>

<dependency>
    <groupId>com.kumuluz.ee</groupId>
    <artifactId>kumuluzee-servlet-jetty</artifactId>
    <version>2.0.0</version>
</dependency>

<dependency>
    <groupId>com.kumuluz.ee</groupId>
    <artifactId>kumuluzee-jax-rs-jersey</artifactId>
    <version>2.0.0</version>
</dependency>
```

Add standard Java EE components

```
@ApplicationPath("rest")
public class RestApplication extends Application {
}

@Path("catalog")
public class SampleREST {

    @GET
    @Produces({"application/xml", "application/json"})
    public Response hello() {
        return Response.ok(“hello from KumuluzEE”);
    }
}
```

Build it

```
$ mvn clean package
```

And run it dirrectly

```
$ java -cp target/classes:target/dependency/* com.kumuluz.ee.EeApplication
```

Or run from the produced JAR

```
$ java -jar target/kumuluzee-example-1.0.0-SNAPSHOT.jar
```

- Scaling is as simple as multiplying the instances and load balancing
- Can scale using PaaS, and all Docker-like environments (Kubernetes, CoreOS, …) or whichever way you like
- Settings are controlled via environment variables
- Each microservice can be scaled completely separately according to its needs

IN SUMMARY…

- Simply transition your monolith applications to microservices
- No need to learn any new frameworks
- You're in complete control of what components you use
- Run and scale anywhere

- https://ee.kumuluz.com
- https://github.com/tfaga/kumuluzee