

마이크로서비스로 구축하는 경량 고성능 분산채팅 플랫폼

Reactive Web/App 개발을 위한 가장 핫한 방법

- The Reactive Manifesto
- Reactive vs nonReactive
- Microservices vs Monolithic Architecture
- Microservices with MQ
- Deployment/Management Microservices
- Vert.x, eventBus, sockJs, Redis
- Workshop1 - single Vert.x chat server
- Workshop2 - cluster Vert.x chat server
- Workshop3 - Distributed single Vert.x chat server
- Q & A

The Reactive Manifesto

<http://www.reactivemanifesto.org/>

The Reactive Manifesto

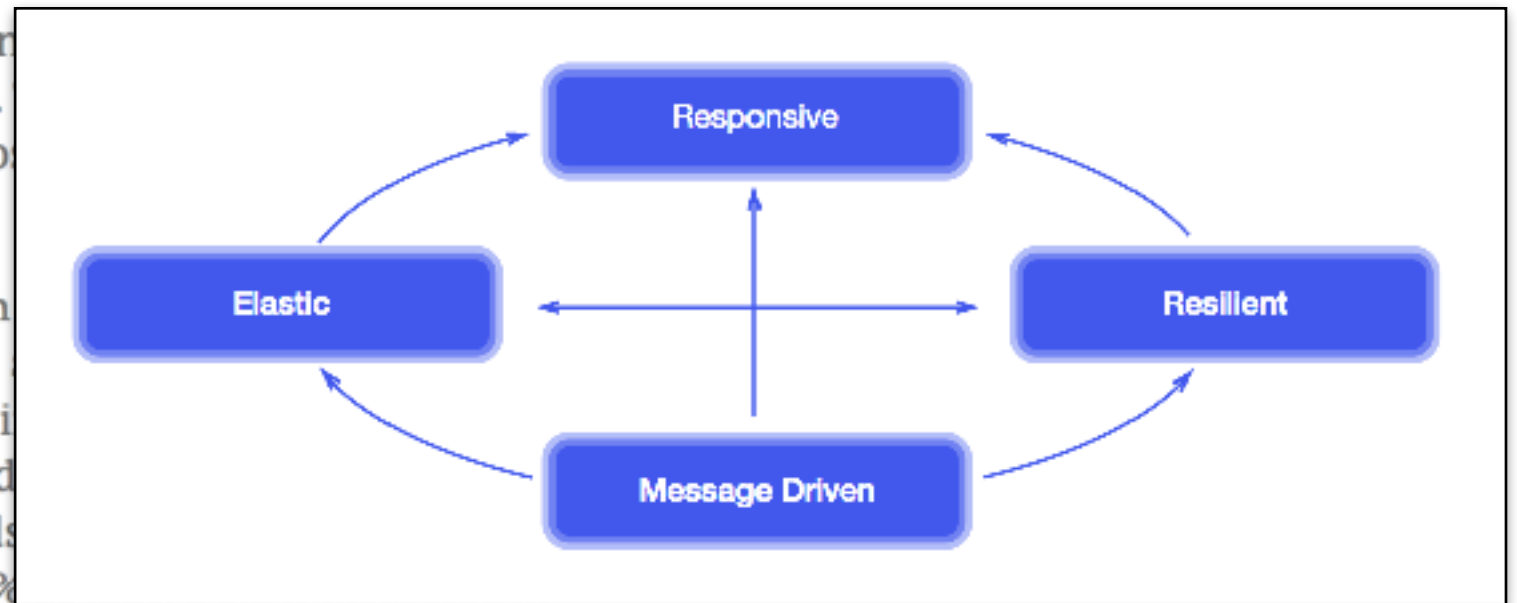
Published on September 16 2014. (v2.0)

Organisations working in disparate domains are in patterns for building software that look the same. robust, more resilient, more flexible and better po demands.

These changes are happening because application dramatically in recent years. Only a few years ago of servers, seconds of response time, hours of offli gigabytes of data. Today applications are deployed devices to cloud-based clusters running thousands Users expect millisecond response times and 100% Petabytes. Today's demands are simply not met by yesterday's software architectures.

We believe that a coherent believe that all necessary as systems that are Responsive these Reactive Systems.

Systems built as Reactive S [scalable](#). This makes them significantly more tolerant with elegance rather than d giving [users](#) effective intera



[Sign the manifesto](#)

17942 people already signed ([Full list](#))

[Tweet](#) [Like 2.0K](#) [Share 814](#) [LinkedIn 1,934](#)

[Glossary](#)

[Download as PDF](#)

[Suggest improvements](#)

Reactive vs nonReactive

Benefits of Reactive

Microservices vs Monolithic Architecture

Benefits of Microservices

Microservices with MQ

!!! WORKSHOP !!!

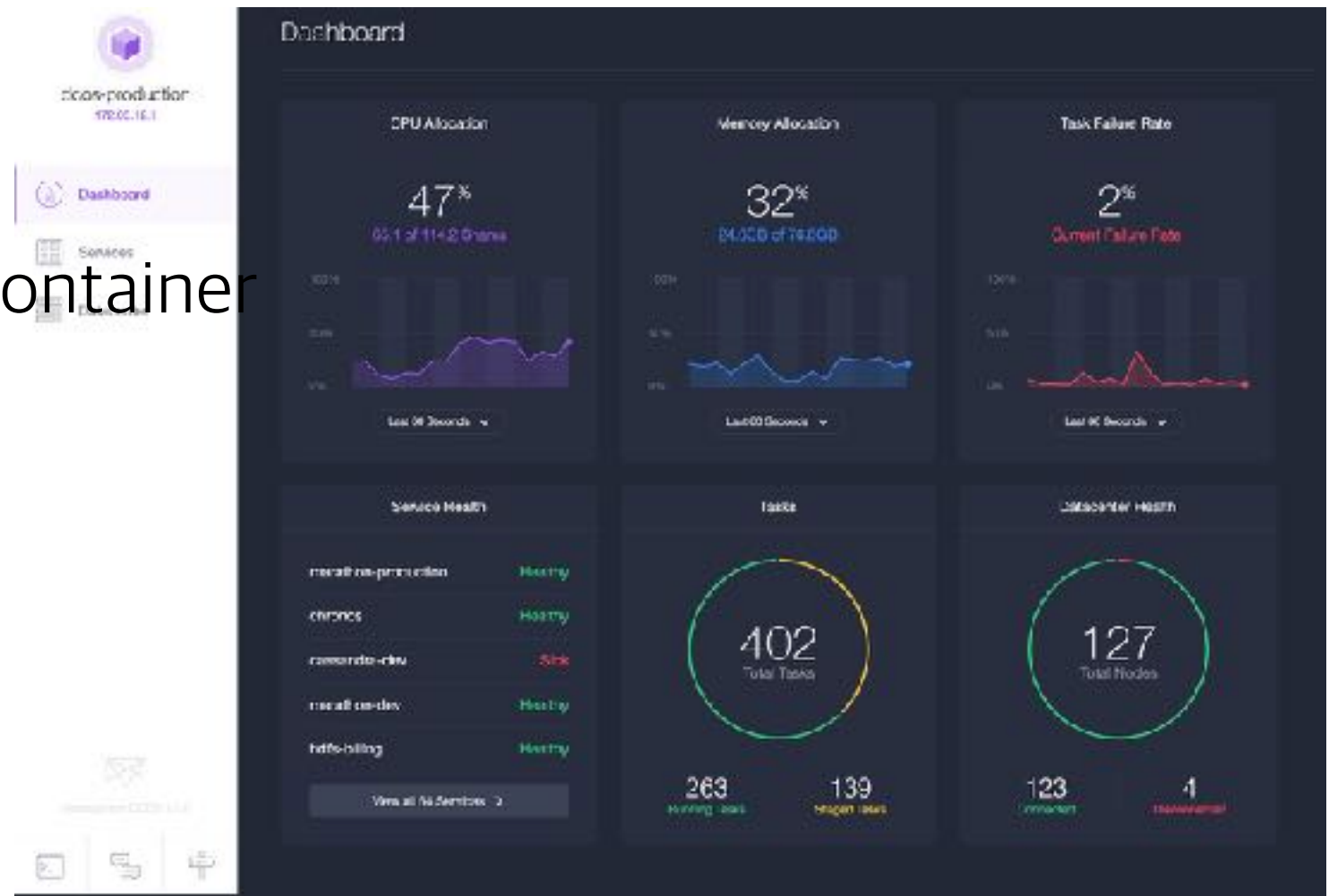
Deployment/Management Microservices

- Service instance per container

- **Docker**

- Swarm

- Kubernetes

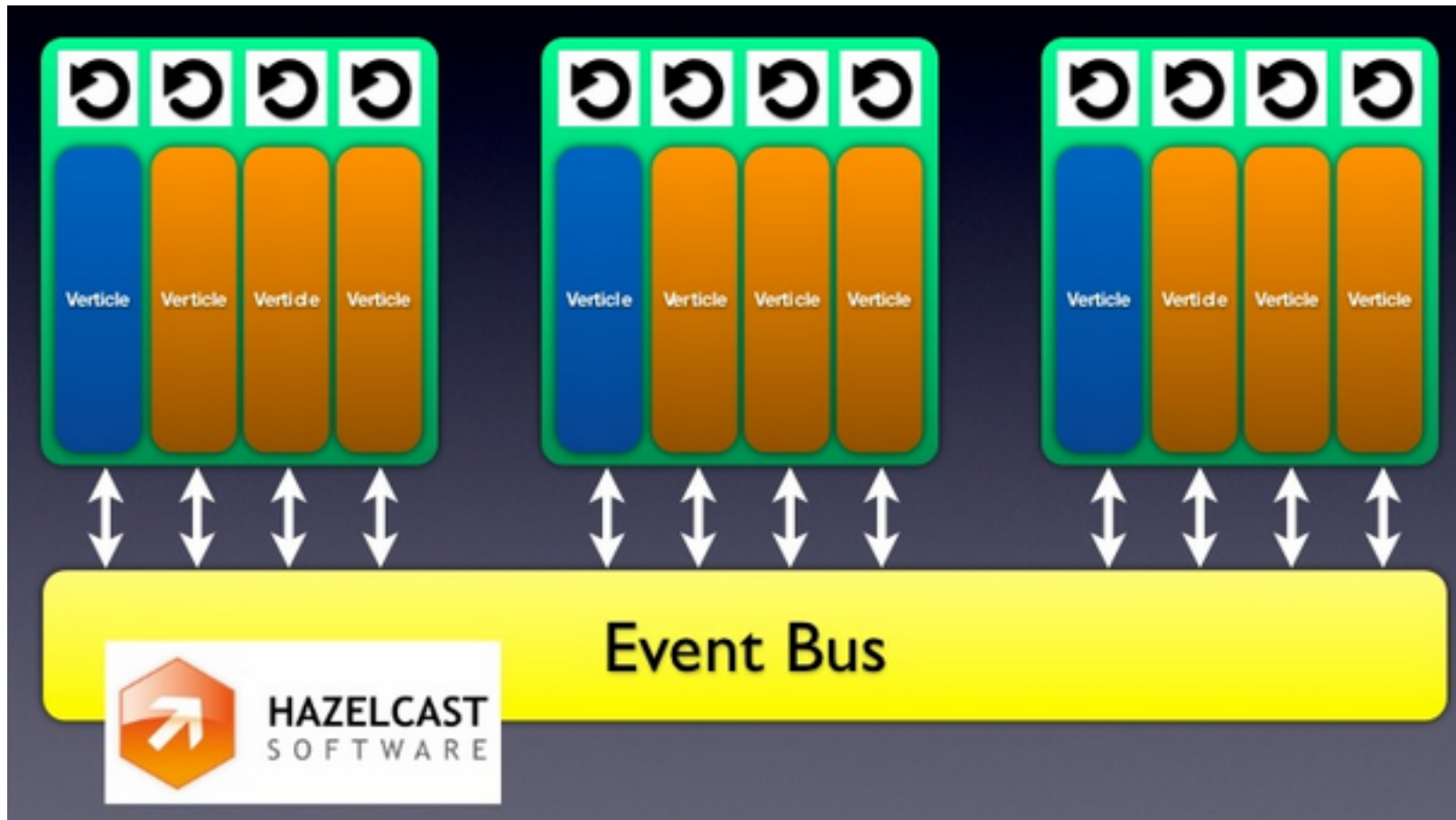


<https://dcos.io/>

- **DCOS - Docker/Marathon/Mesos**

- Amazon EC2 Container Service

Vert.x, eventBus, sockJs, Redis



<http://vertx.io/>

<https://redis.io/>



<https://github.com/sockjs>

Workshop1

Single Vert.x chat server

<https://github.com/unlogicaldev/slime/tree/master/slime-microservice-chat>

- slime-http : http web server & static resource
- slime-sockjs : sockjs socket server

```
#>cd slime-http
```

```
#>mvn clean package
```

```
#>java -jar target/slime-http-0.0.1-SNAPSHOT-fat.jar
```

```
#>cd slime-sockjs
```

```
#>mvn clean package
```

```
#>java -jar target/slime-sockjs-0.0.1-SNAPSHOT-fat.jar
```

Workshop2

Cluster(HAZELCAST) Vert.x chat server

- slime-http : http web server & static resource
- slime-sockjs : sockjs socket server

```
#>cd slime-http
```

```
#>mvn clean package
```

```
#>java -jar target/slime-http-0.0.1-SNAPSHOT-fat.jar
```

```
#>cd slime-sockjs
```

```
#>mvn clean package
```

```
#>java -jar target/slime-sockjs-0.0.1-SNAPSHOT-fat.jar
```

Workshop3

Distributed **Single** Vert.x chat server

- slime-http : http web server & static resource
- slime-sockjs : sockjs socket server
- slime-publish : message publish microservice

```
#>cd slime-http
```

```
#>mvn clean package
```

```
#>java -jar target/slime-http-0.0.1-SNAPSHOT-fat.jar
```

```
#>cd slime-publish
```

```
#>mvn clean package
```

```
#>java -jar target/slime-publish-0.0.1-SNAPSHOT-fat.jar
```

```
#>cd slime-sockjs
```

```
#>mvn clean package
```

```
#>java -jar target/slime-sockjs-0.0.1-SNAPSHOT-fat.jar
```

Q & A

<https://github.com/unlogicaldev/slime/tree/master/slime-microservice-chat>

<https://github.com/sayseakleng/vertx-eventbus-chat>

<https://github.com/cescoffier/vertx-microservices-workshop>

<https://www.facebook.com/groups/jbossusergroup/>

unlogicaldev@gmail.com Rhys Lee