



RED HAT  
**FORUM**  
Europe, Middle East & Africa



# Déploiement des Microservices à l'échelle de l'entreprise

**Malik SAHEB**

*Senior Solutions Architect - Red Hat*

**Antoine Ferté**

*Directeur Technique Europe du Sud - Dynatrace*

*PARIS, le 11 octobre 2017*

# Disruptive IT



“Applications created today using the good-old traditional architecture will be a business-constraining legacy even before they are completed.”

*Nefim Yatis, Gartner*

“Organisations that haven’t modernized their application architecture and infrastructure to support digital business will fall further behind new entrants and fast-moving competitors”

*Gartner Jan 2017*

“In today’s era of volatility, there is **no other way but to re-invent**. The **only sustainable advantage** you can have over others is **agility**, that’s it. Because nothing else is sustainable, everything else you create, somebody else will replicate.”

*Jeff Bezos, Founder Amazon*

# Les modèles de développement et de déploiement changent



# 2015



Arun Gupta [@arungupta](#) Following

Microservices is like teenage sex, every body talking about it, but no body is doing it! #TGIF

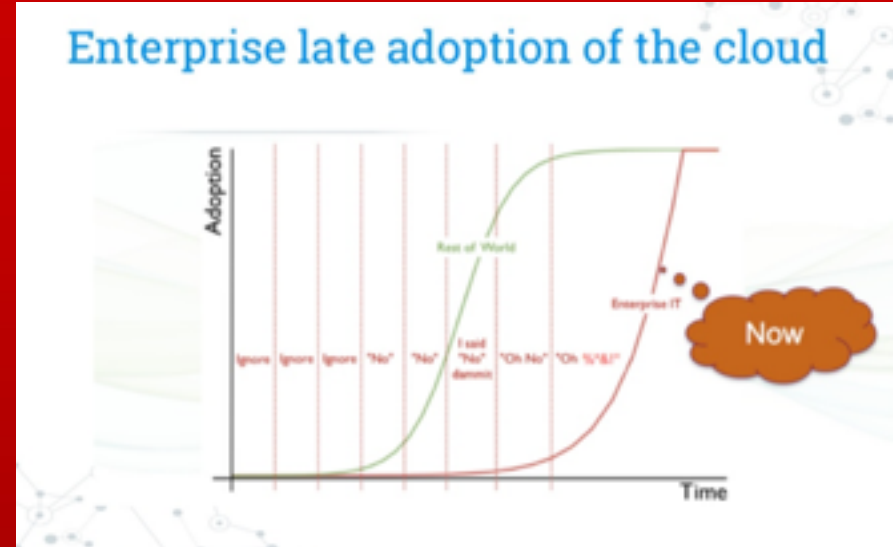
RETWEETS 103 LIKES 55

12:22 PM - 26 Jun 2015

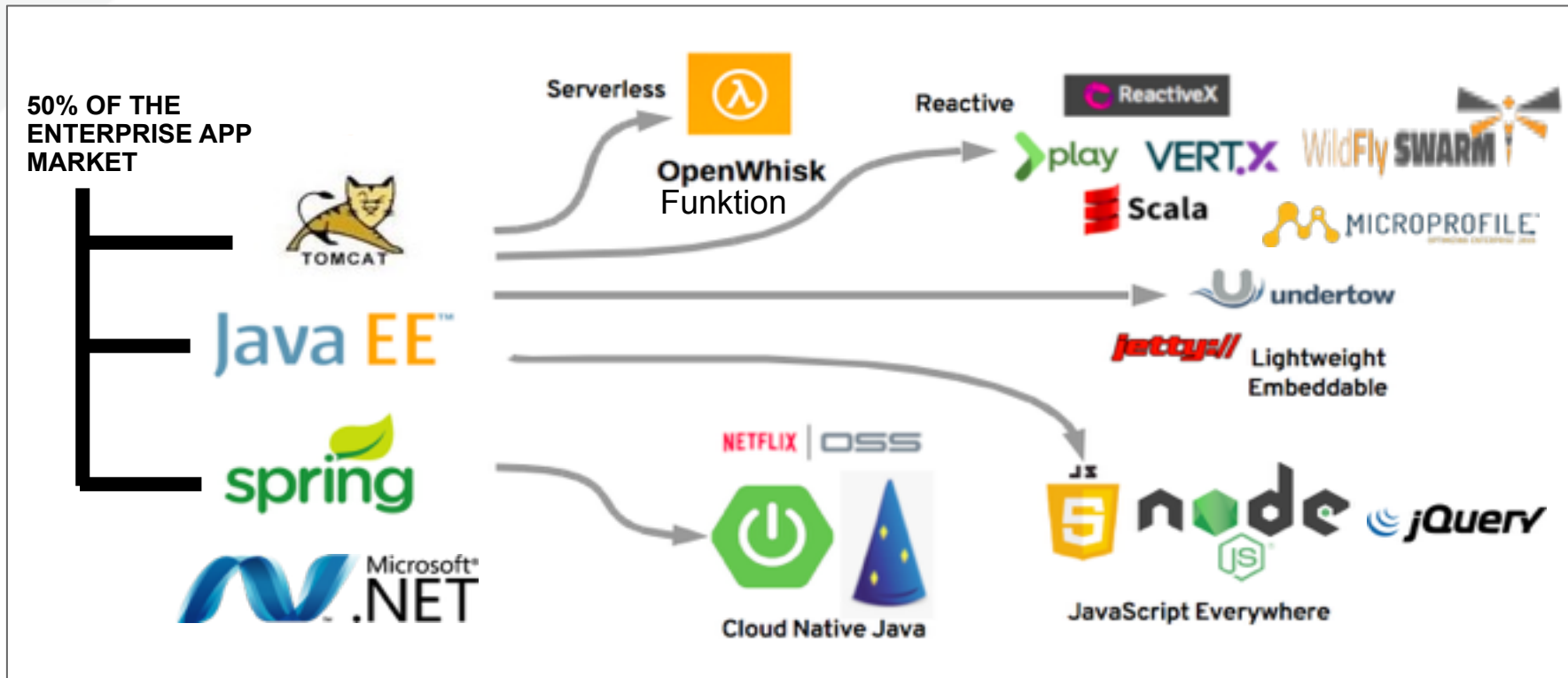
7 103 55

Reply to [@arungupta](#)

# 2017

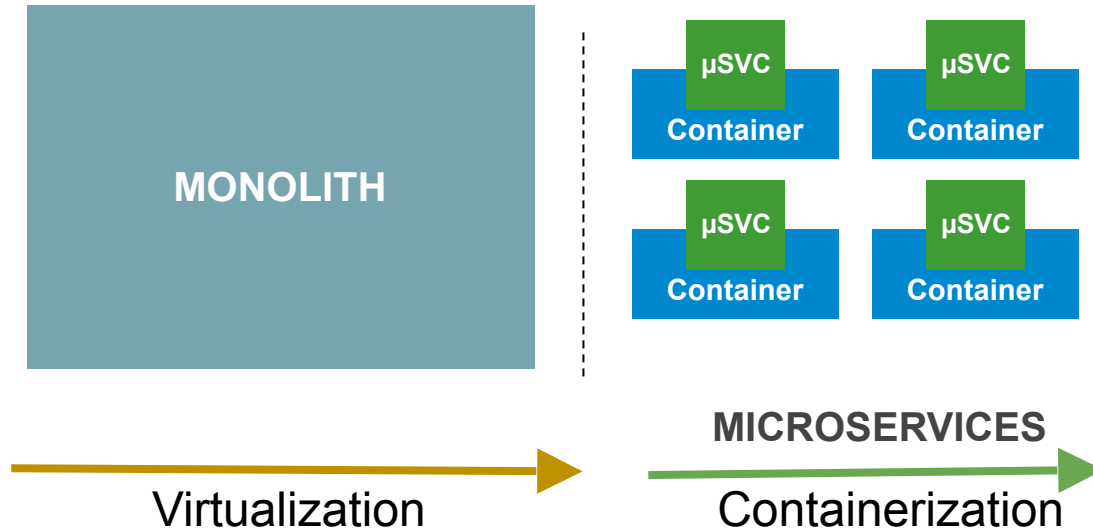


# Les Développeurs ont beaucoup plus de choix



# Microservices

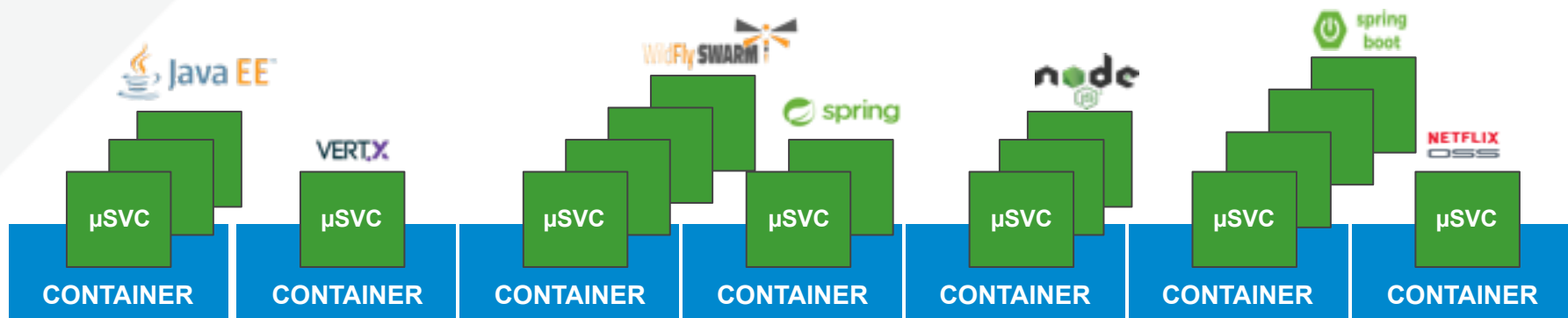
Applications Modernes pour Architectures Hybrides



## μSVC :

- Agile, DevOps
- Polyglot
- Architectural Flexibility
- Easier to manage in the cloud

# Gestion des containers à l'échelle



Data

Security

IMDG

Messaging

Cloud platform

Build | Deploy | Scheduling | Scaling | Elasticity | Metrics | Logging



RED HAT  
OPENSSHIFT



RED HAT  
ENTERPRISE  
VIRTUALIZATION

vmware



Microsoft  
Hyper-V



Microsoft  
Azure



Google Cloud Platform

amazon



# RED HAT® OPENSIFT

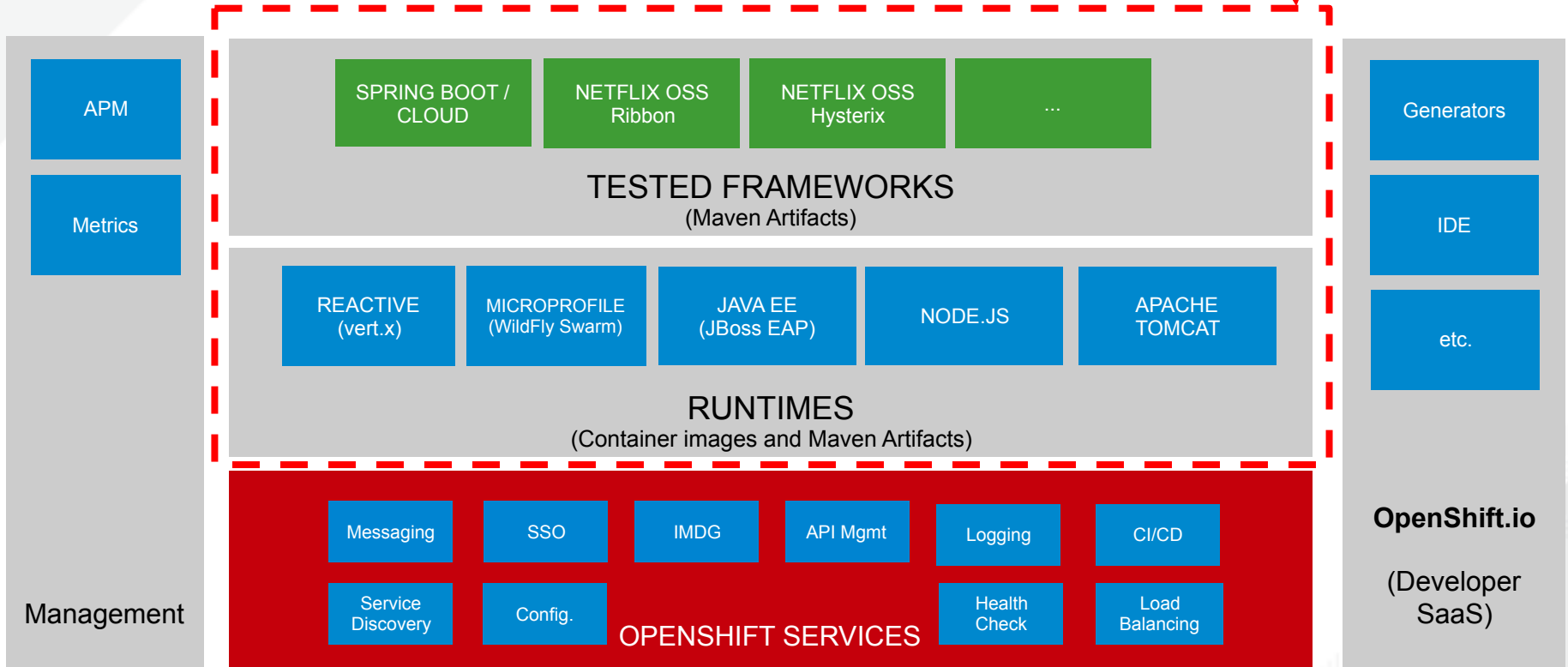
## Application Runtimes

Modern, cloud-native application runtimes and an opinionated developer experience for organizations that are moving beyond 3-tier architectures and embracing cloud-native application development.



# Openshift Application Runtime

(RH OAR)



# Eléments différenciants

## Deployment & Packaging

### Physical Servers



### Virtual Servers



## Containers

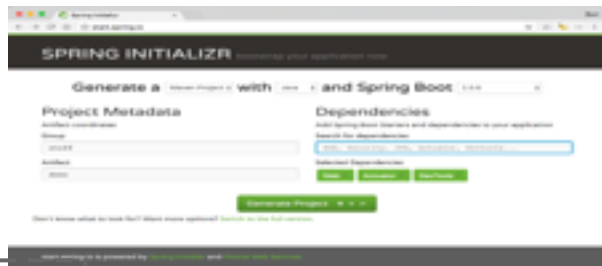


- Plateforme Complète de développement et de déploiement d'applications
- Multi-langages
- Basé sur des containers
- Facilite la portabilité
- Permet des déploiements rapide et consistant du Développement à la Production
- Poly-architecture: microservices, applications traditionnelles
- Conçue pour un mode Cloud à l'échelle

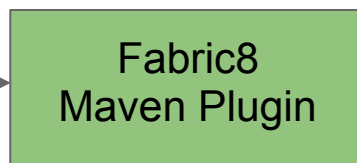
# OUTILS

# “Containérisation” de vos applications

## Fabric8 maven plugin



Ex. Spring Initializr



Fabric8  
build  
s2i



Fabric8  
deploy



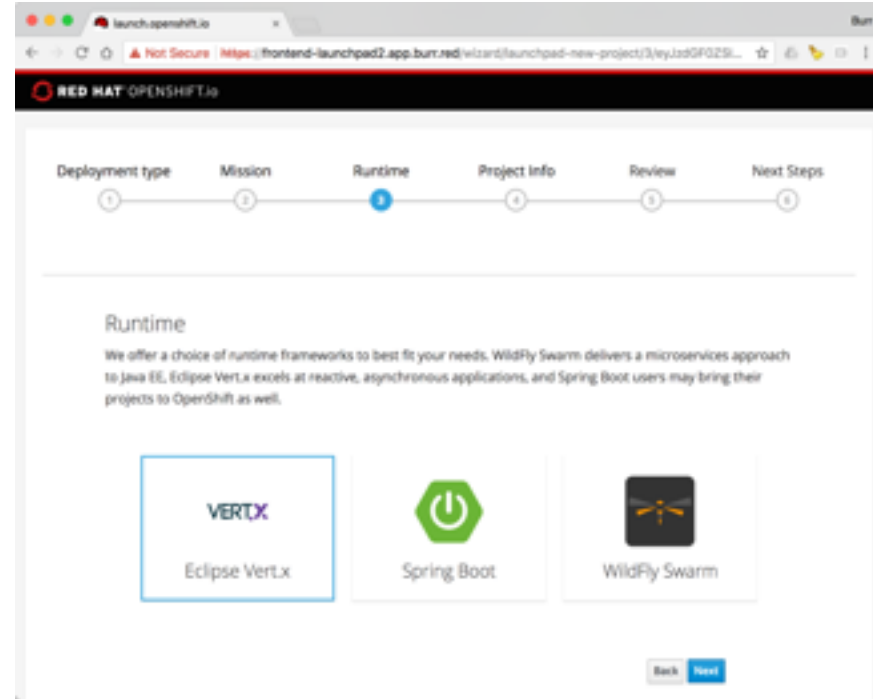
OPENSIFT

```
<groupId>io.fabric8</groupId>
<artifactId>fabric8-maven-plugin</artifactId>
<version>3.5.28</version>
```

# Environnement en ligne avec openshift.io

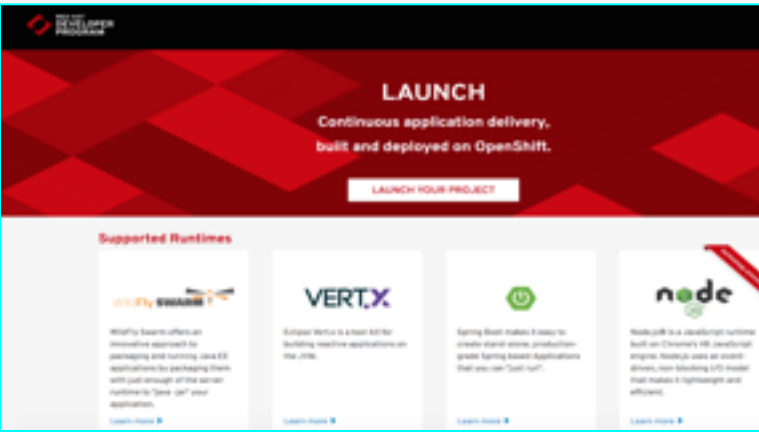
*Environnement de développement, de construction, de planification et de déploiement de bout-en-bout d'applications cloud-native*

- Accélère l'apprentissage et l'expérience
- Collection d'exemples cloud native
- Fonctionne entièrement sur OpenShift
- Combine plusieurs projets innovants:
  - Fabric8, Eclipse Che, JBoss Forge,
  - Spring Boot, Vert.x, WildFly Swarm



• [launch.openshift.io](https://launch.openshift.io)

# launch.openshift.io

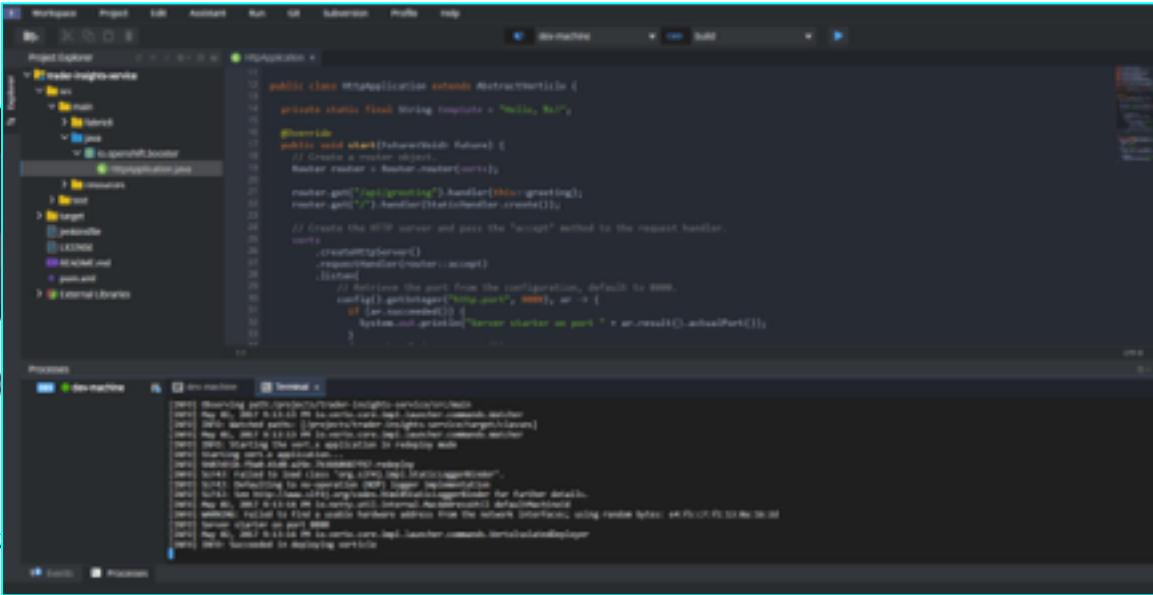


**LAUNCH**  
Continuous application delivery,  
built and deployed on OpenShift.

[LAUNCH YOUR PROJECT](#)

**Supported Runtimes**

- WildFly Swarm**: WildFly Swarm offers an innovative approach to packaging and running Java EE applications by packaging them with container images and running them on OpenShift.
- Vert.x**: Support Vert.x is a great option for building reactive applications on the JVM.
- Spring Boot**: Spring Boot makes it easy to create stand-alone, production-grade Spring-based Applications that you can "just run".
- Node.js**: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.



```
public class MyAppApplication extends AbstractArtifact {
    private static final String CONFIG = "hello, hi!";

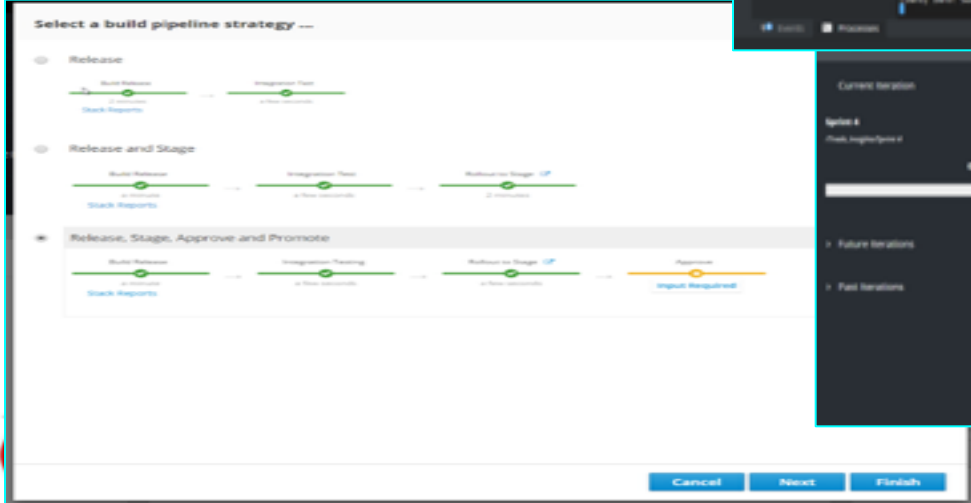
    @Override
    public void start(EvaluationContext context) {
        // Create a reactor object.
        Reactor reactor = Reactor.builder(context);

        reactor.gem("http://localhost:8080", handler((status, greeting) {
            reactor.gem("?", handler((status, handler) {
                // Create the HTTP server and pass the "accept" method to the request handler.
                .createHttpServer()
                .requestHandler(reactor.accept())
                .build();

                // Retrieve the port from the configuration, default to 8080.
                Integer port = context.getConfig("http.port", 8080);
                System.out.println("Server started on port " + port);
            });
        });
    }
}
```

Process: [logs]

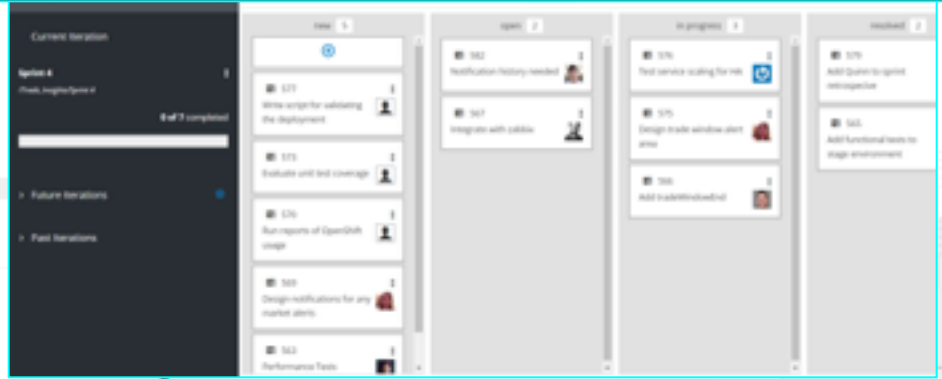
```
Starting java: java -jar /usr/share/containers/oci-runtime/bin/containers-runtime
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Running java: [ /usr/share/containers/oci-runtime/bin/containers-runtime
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Starting the java: application in relative mode
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Starting java: application
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Failed to load class 'org.springframework.boot.autoconfigure
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Retrieving the OpenShift (OS) support configuration
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Failed to find a public hostname address from the network interface, using random bytes: 48.F0.CD.F5.52.86.28
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Server started on port 8080
May 01, 2017 11:11:00 AM io.openshift.launcher.commands.deploy
INFO: Server started on port 8080
```



Select a build pipeline strategy --

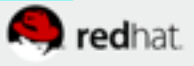
- Release
- Release and Stage
- Release, Stage, Approve and Promote

Cancel Next Finish



Current iteration: 1 of 7 completed

Step	Status	Progress
101	Notification history needed	100%
102	Test service waiting for v1	100%
103	Design trade window alert view	100%
104	Add functional tests to stage environment	100%
105	Add KubernetesPod	100%
106	Performance Tests	100%



# Challenges pour la gestion des containers à grande échelle



**EFFICACITE  
OPERATIONELLE**



**SANTE DES  
SERVICE**

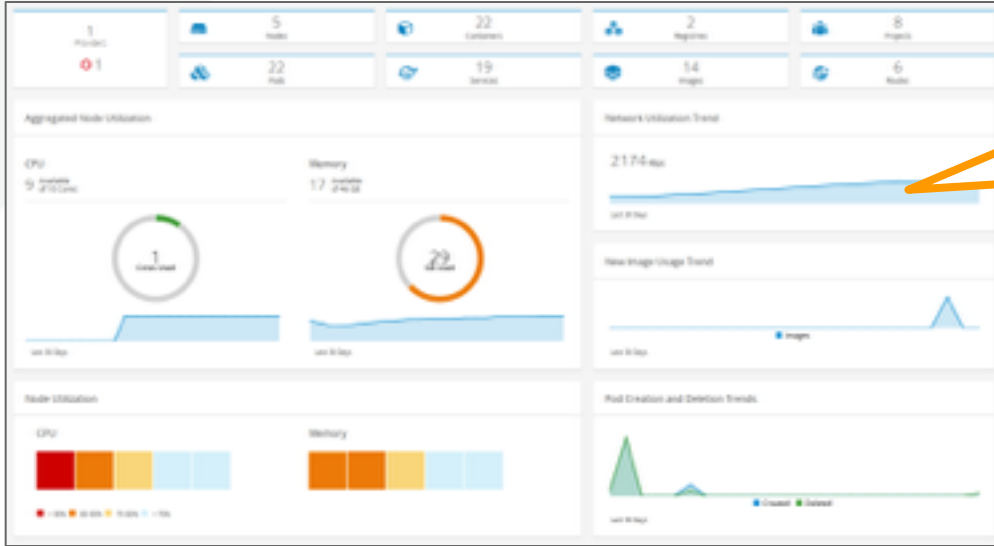


**SÉCURITÉ &  
CONFORMITE**



**GESTION  
FINANCIERE**

# Vue enterprise



How many pods?  
How many Images?  
How many Pods created/deleted trend?  
Which Nodes is running Out?  
Do I still have some capacity?

Properties

Name	ocp-node01-4f6b-986a-986a-986a-986a
IP Address	192.168.17.12
MAC Address	5c:7d:3c:6a:4f:9a
Container	1 approach: 4 CPUs, 8 GB memory, 1 core, 100MB disk
Platform Tools	ok
Operating System	rhel, rhel
Deployment	none
Advanced Settings	0
Resources	Available
Management Engine GUID	4147750a-5003-17a7-8029-000000000000
OS User	root
OS Vendor	45021708-950-4a48-850c-1a8f10-840000

ocp-n-1-0001

Where my Nodes are running?





# Inventaire et monitoring

## cicd (Summary)

Properties	
Name	cicd
Display name	CI/CD
Creation timestamp	Thu, 02 Feb 2017 18:23:40 +0000
Resource version	548745

Relationships	
Containers Provider	OpenShift Container Platform
Routes	4
Container Services	7
Replicators	6
Pods	6
Nodes	2
Container Images	5
Container Templates	0

Smart Management  
Red Hat Tags No Red Hat Tags have been assigned

How many pods in this project?  
How many Images?  
What are these images?  
Any Quota and Limits?

Name
gogs-1-e50px
jenkins-1-6hc03
nexus-1-fahrd
postgresql-gogs-1-4ttip
postgresql-sonarqube-1-wt31z
sonarqube-1-m6ikm

On Which Nodes run my Project?

	Name	Provider	Ready	Operating System	Kernel Version	Runtime Version
<input type="checkbox"/>	ocp-node-1.lab.example.com	OpenShift Container Platform	True	Employee SKU	3.10.0-514.6.1.el7.x86_64	docker://1.10.3
<input type="checkbox"/>	ocp-node-2.lab.example.com	OpenShift Container Platform	True	Employee SKU	3.10.0-514.6.1.el7.x86_64	docker://1.10.3



Vous pouvez aussi compter  
sur nos partenaires comme  
Dynatrace

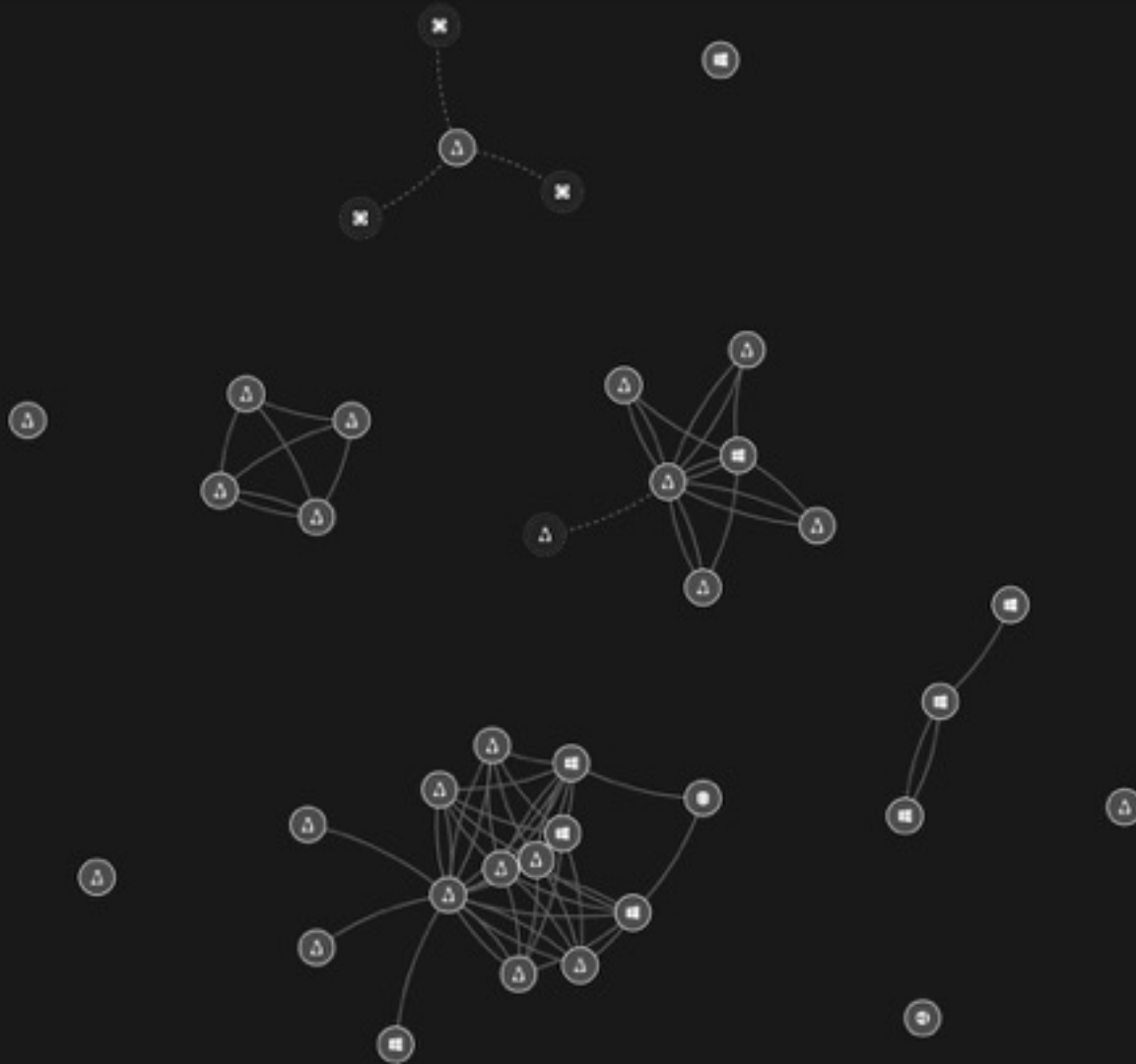


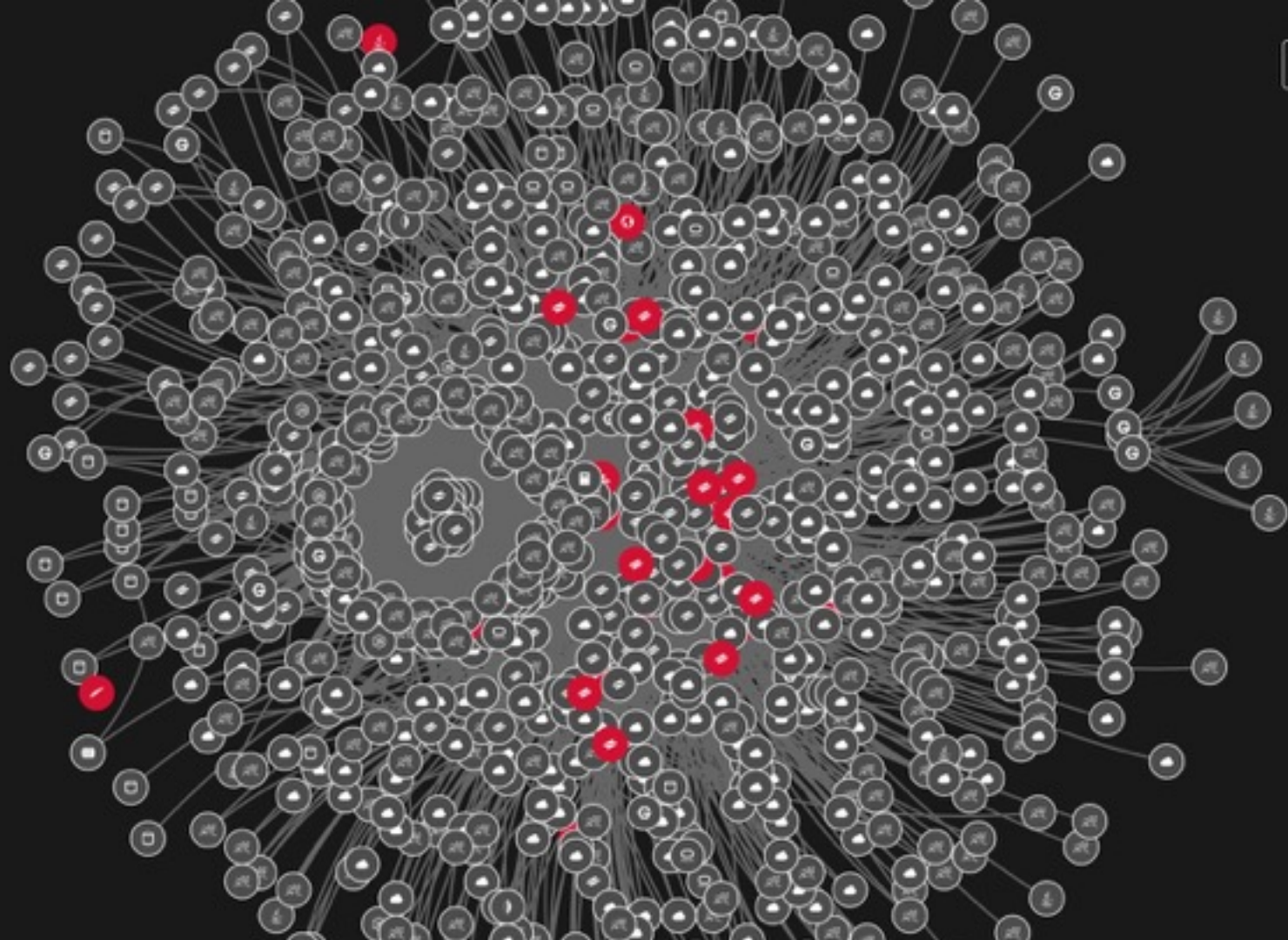
**SANTE DES  
SERVICE**



# dynatrace

Maitriser votre  
**qualité de service**  
sur vos déploiements  
**Openshift**



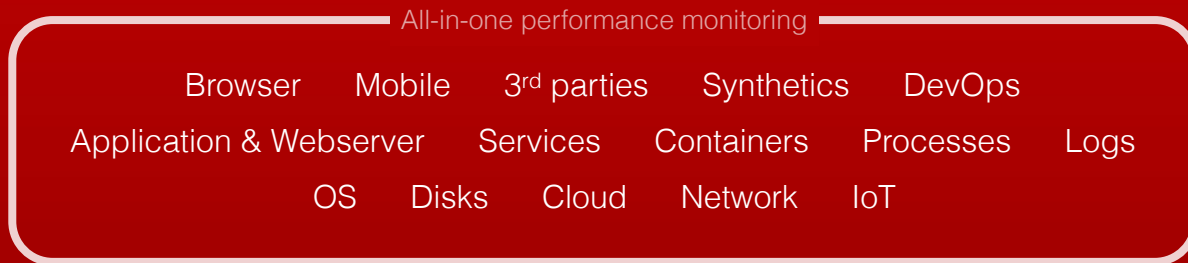
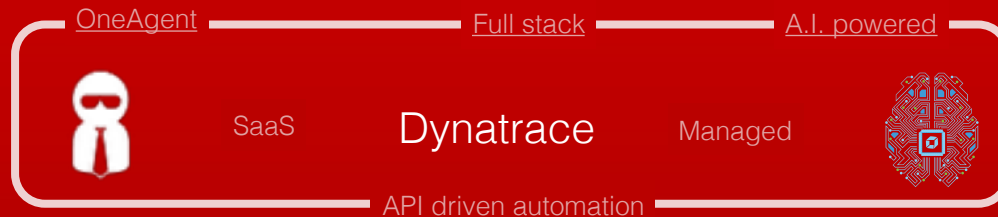
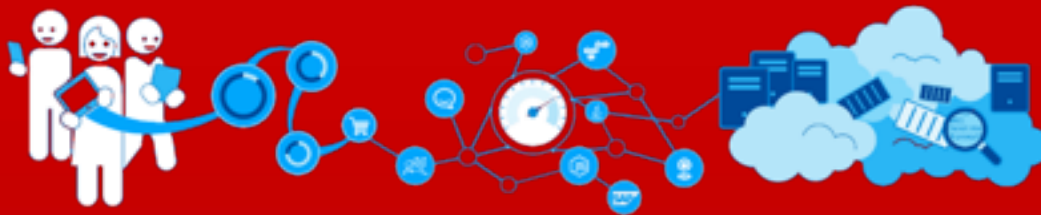


# Monitoring redefined

Digital experience analytics

Application performance

Cloud, container, infrastructure





# One Agent

- Browser
- Mobile
- 3<sup>rd</sup> parties
- Synthetics
- DevOps
- Application &
- Webserver
- Services
- Containers
- Processes
- Logs
- OS
- Disks
- Cloud
- Network
- IoT





Browser



Mobile



3<sup>rd</sup> parties



Application- & Webserver



Services



Containers, Processes,  
Logs



OS, Disks



Cloud



Network

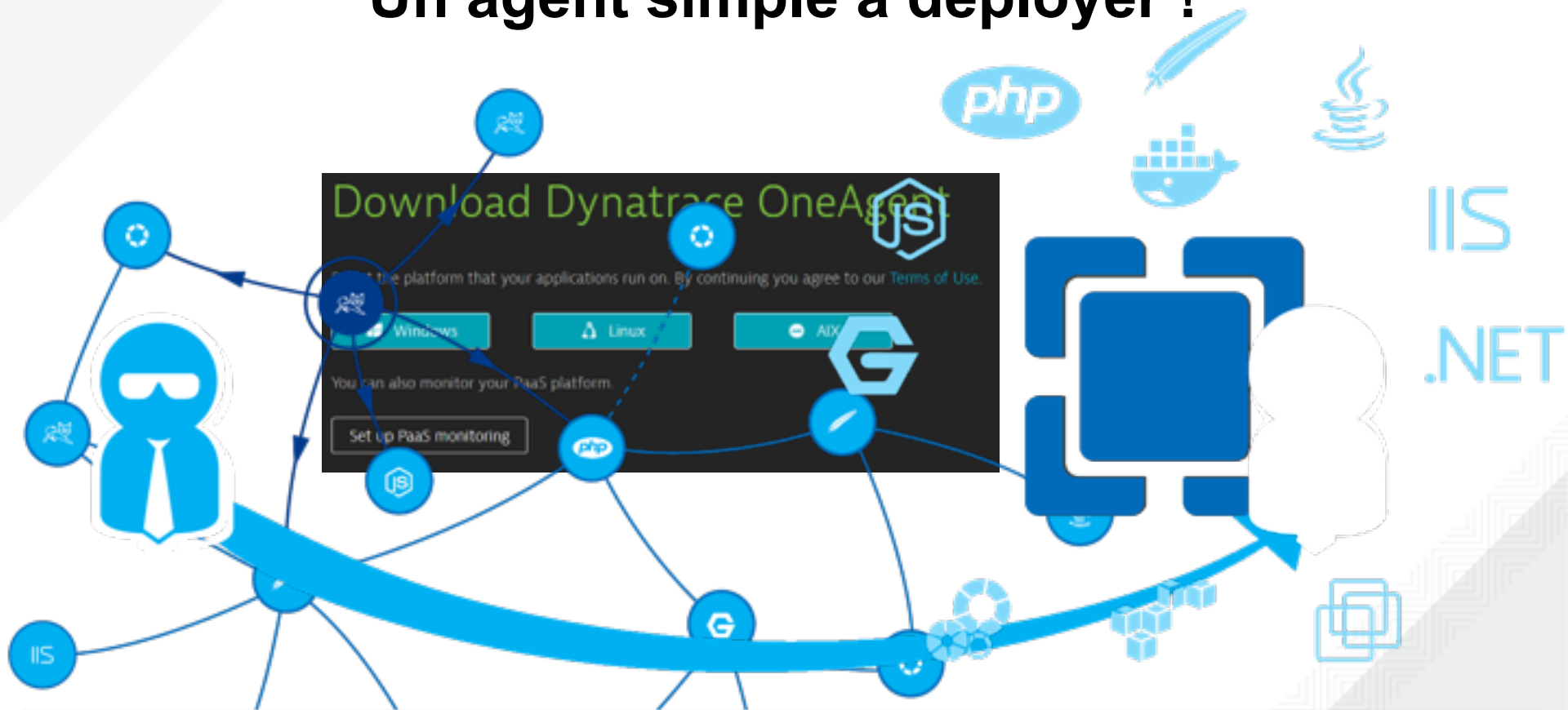


# Full Stack

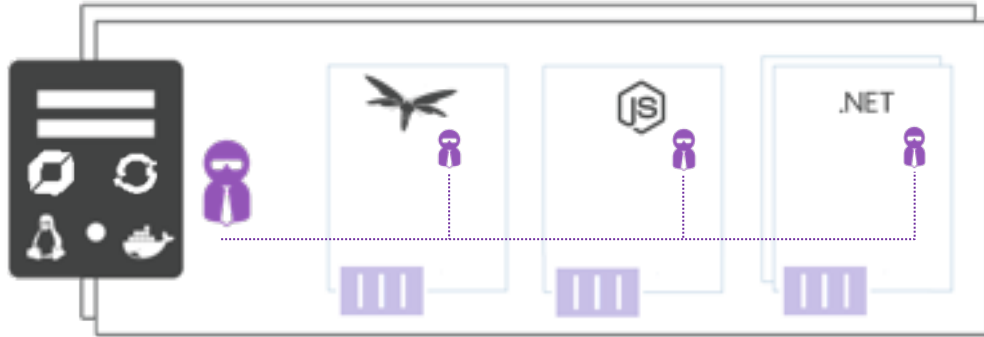
Discovery, Modeling & Analytics



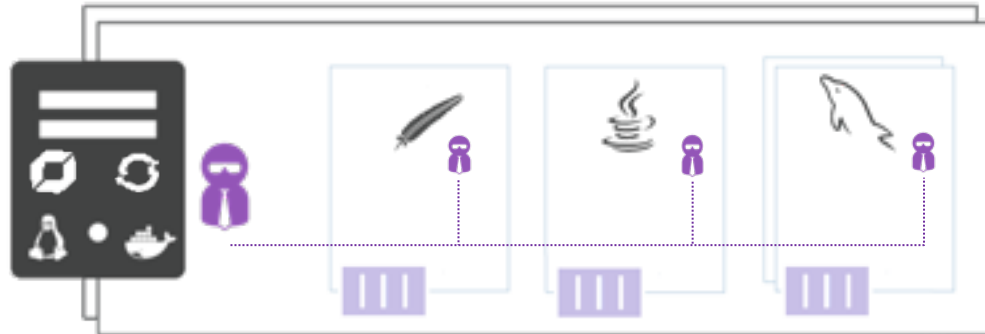
# Un agent simple à déployer !



# Même dans les environnements « conteneurisés »

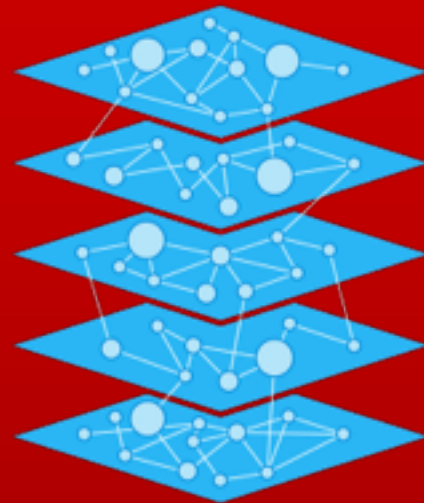


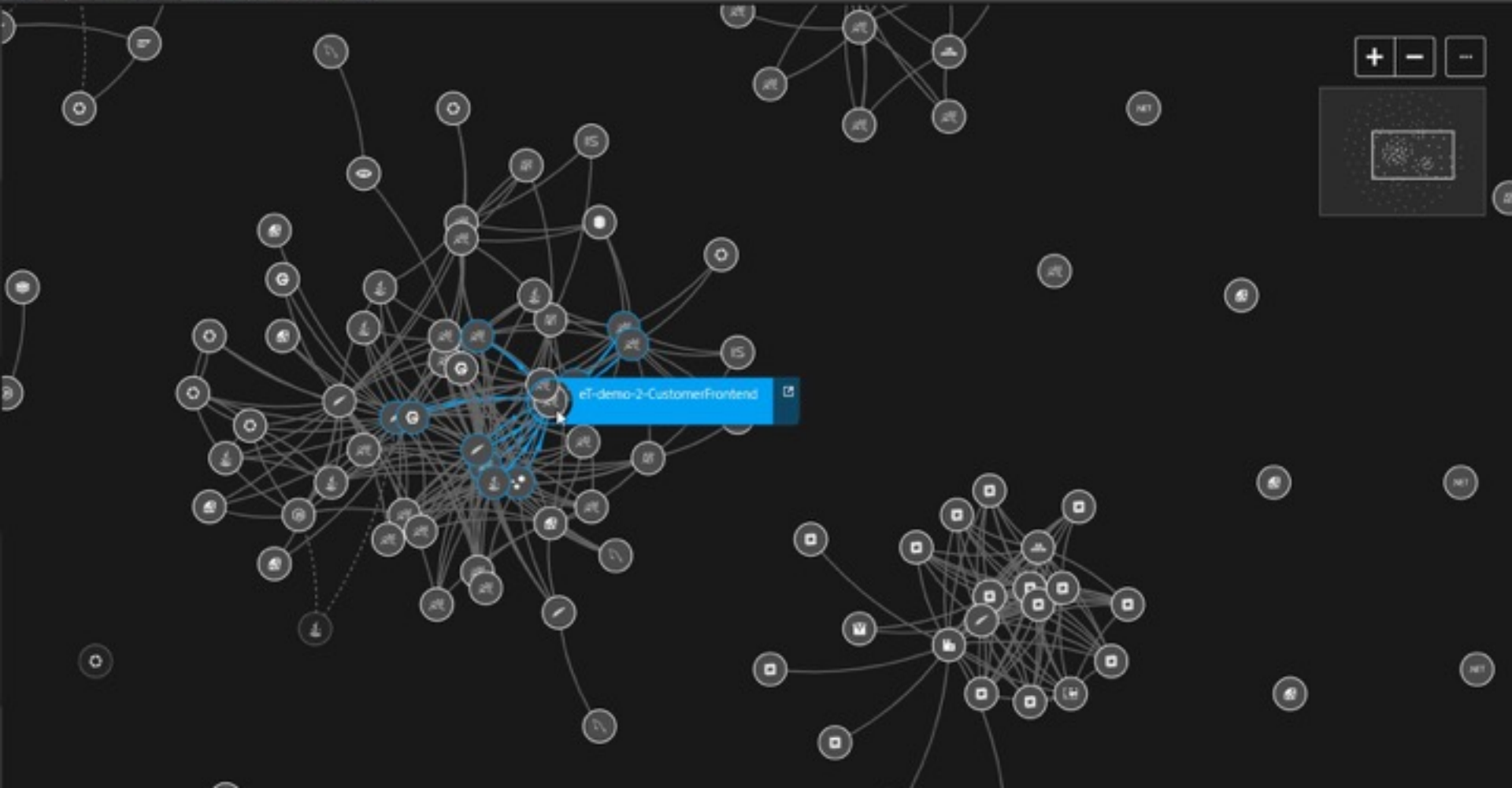
1. Installation sur le Node
2. Injection automatique dans tous les conteneurs





# Full-Stack Monitoring





Applications 10

easytravel Demo  
Mobile Application

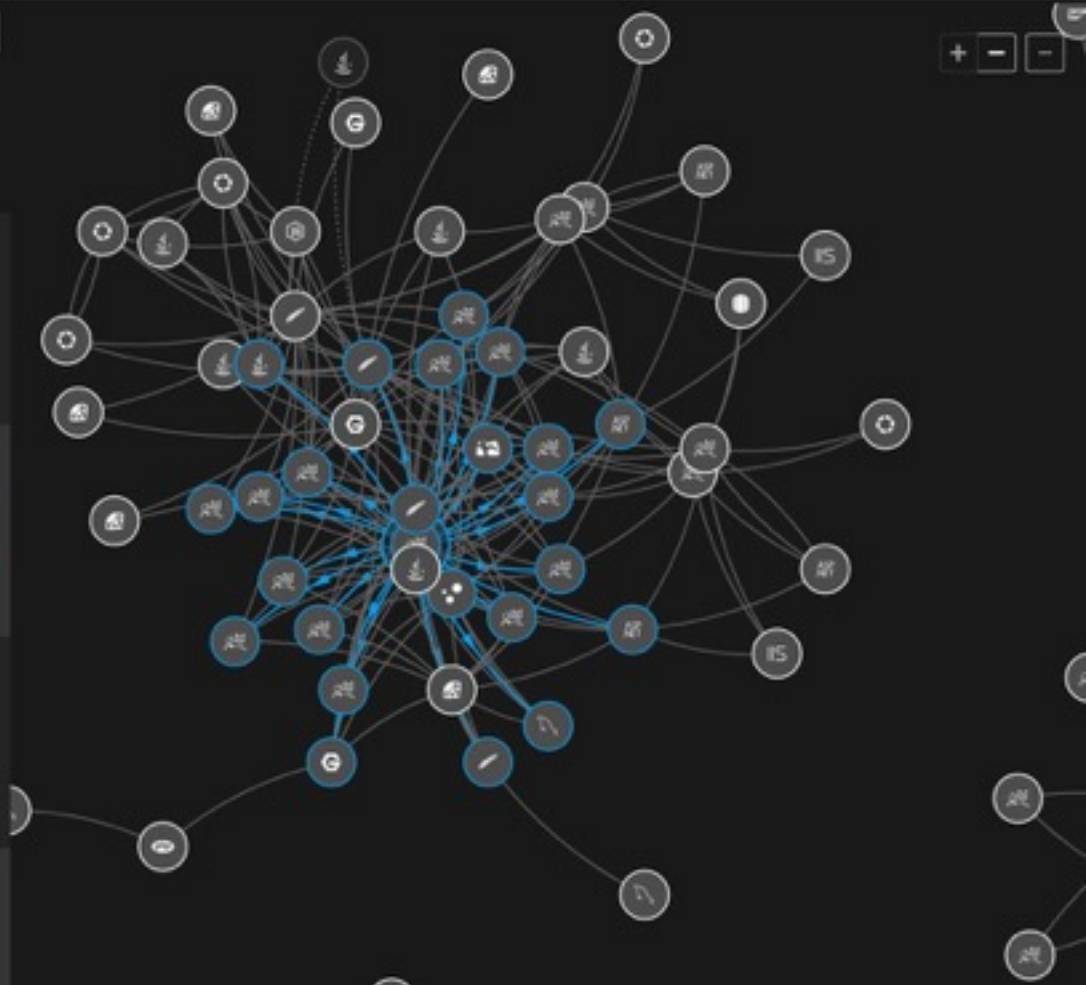
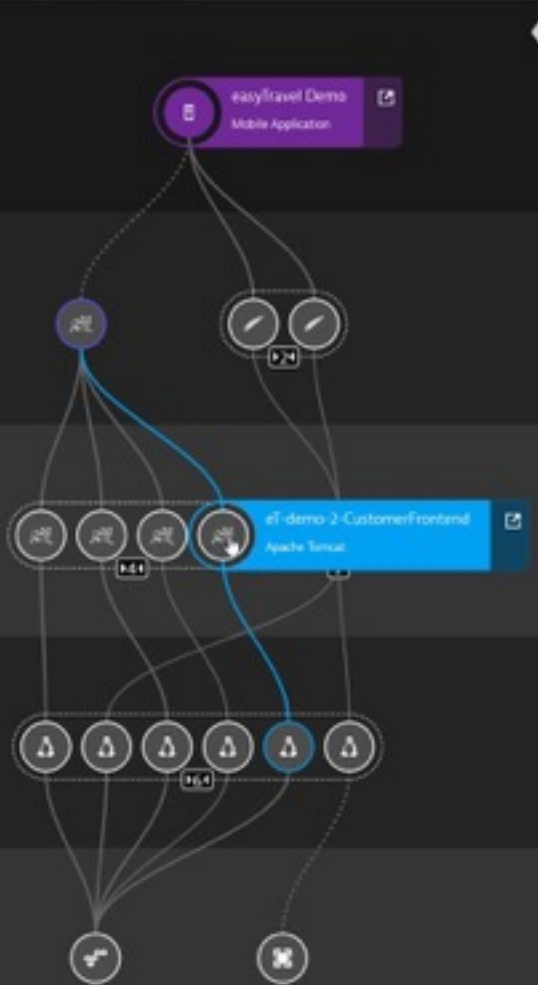
Services

Processes

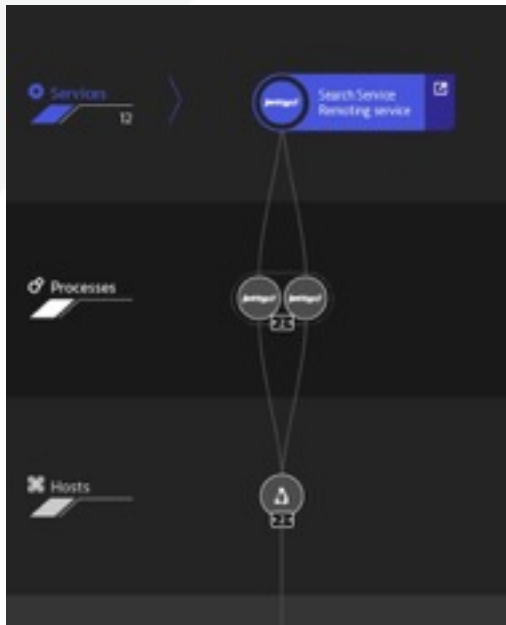
eT-demo-2-CustomerFrontend  
Apache Tomcat

Hosts

Data centers

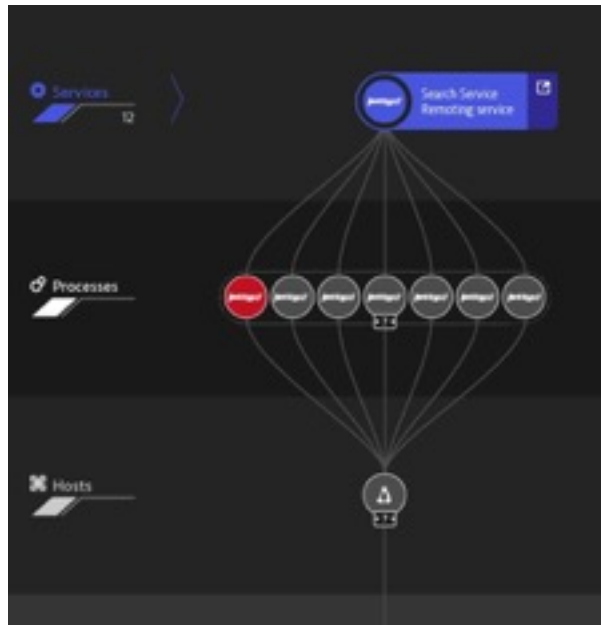


# Adapté aux déploiements élastiques



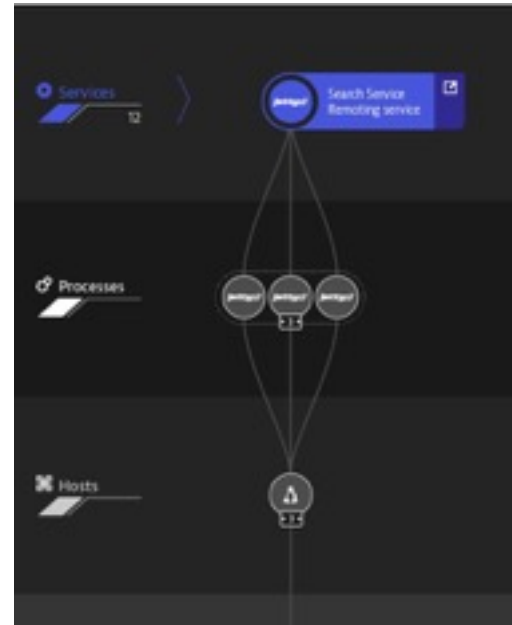
7:00

Peu de charge, le service tourne sur un minimum de redondance



12:00 p.m.

Le service s'appuie sur 7 processus afin de garantir une bonne répartition de charge et du failover pour les processus "plantés"

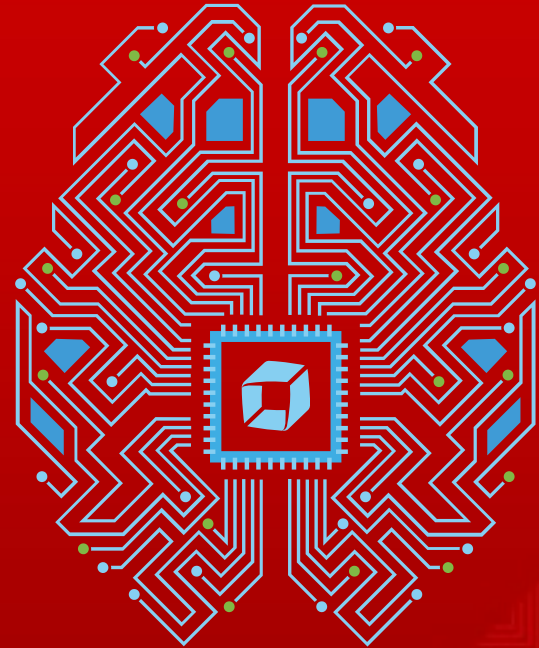


7:00 p.m.

Diminution de la charge donc du nombre de processus utilisés

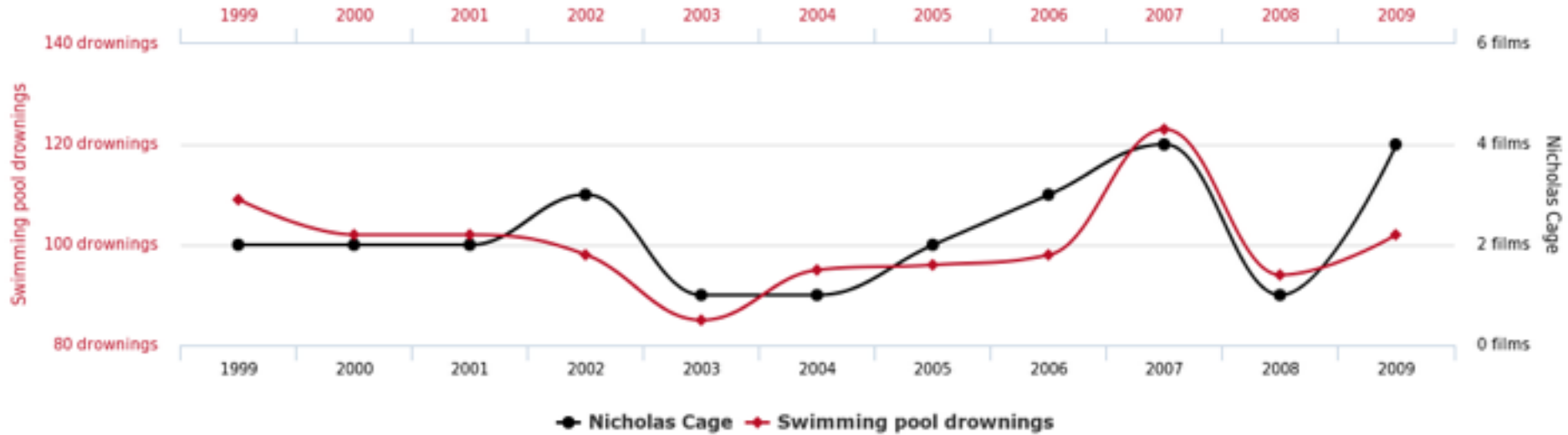


# Artificial intelligence Powered



# La corrélation a ses limites !

Number of people who drowned by falling into a pool  
correlates with  
Films Nicolas Cage appeared in



tylervigen.com



# Dynatrace = Causalité

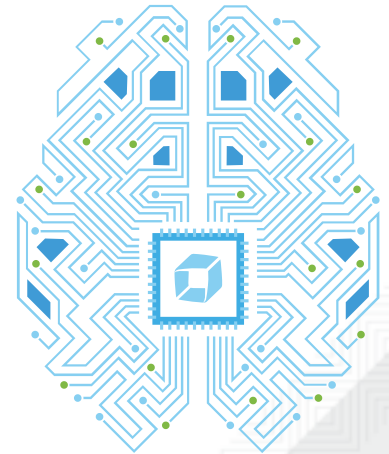
Gapless Data



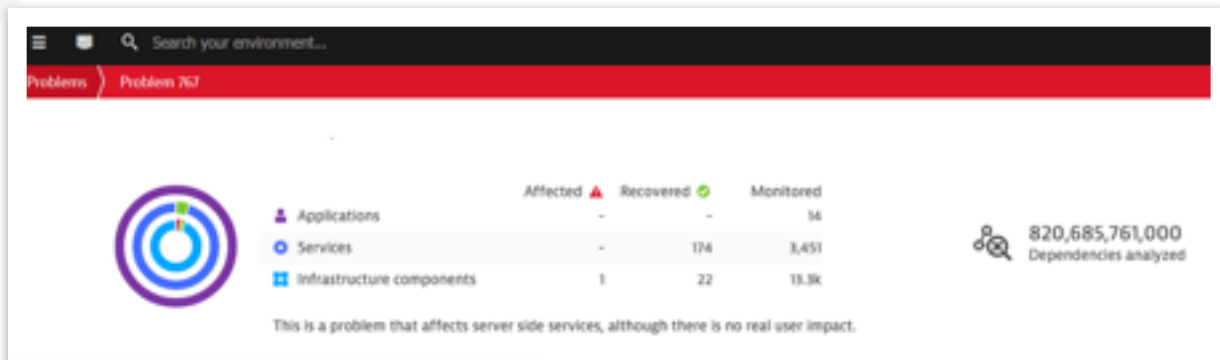
Smartscape Model



Artificial Intelligence



# Découvertes automatique des problèmes



Mushroom cloud effect

## Root cause

Based on our dependency analysis all incidents have the same root cause:

**docker**

Process

Analyze Root Cause

**Connectivity problem** ▲

TCP connectivity rate for process docker on host b2w-persistencia-glets-03 has decreased to 95 %

**docker**

Process. Since 10 h 43 m healthy again

Analyze Root Cause

**Connectivity problem** ●

TCP connectivity rate for process docker on host b2w-persistencia-glets-02 has decreased to 0 %

Problème réseau



return

**ANSWERS**

alt

control

**VS**



**DATAS**

# DEMONSTRATION



dynatrace





Top 3 containers by resource consumption

- 1. engine, 171 MB memory
- 2. image\_gallery, 291 MB memory
- 3. mongo-f9a7781-f88e-495d-918c-e37f06481c8d-51c0160142-7652-4a80-9403-6b082c04412, 198 MB memory

Latest container started

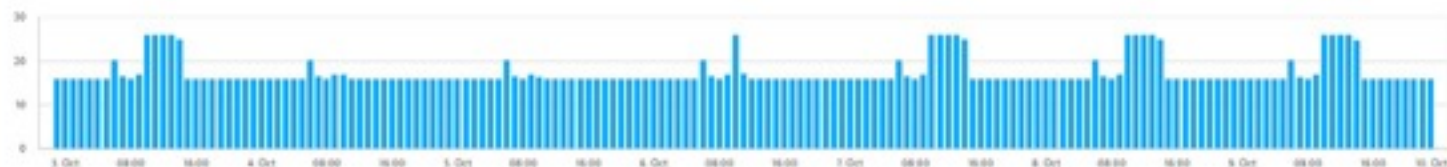
No data.

Top 3 active images with running containers

- 1. easytravel/nodejs-proxy, 10 containers
- 2. redis-weather, 2 containers
- 3. mongo, 1 container

Average number of running containers

Last 7 Days



16  
Containers running on hosts  
No change  
Compared to Mon, Oct 2

4 Docker hosts

Search

Name	CPU usage	Memory usage	Disk latency	Network traffic	Number of containers	State
CCA-docker-node-engine-mongo	10 %	53 % of 3.68 GB	0.41 ms	2.38 MB/s	11	Running



## 6 Docker images

Image name	Containers	Created	Virtual size	Details
easytravel/hodap-proxy	10	-	0.0	▼
redis-weather	2	-	0.0	▼
mongo	1	-	0.0	▼
easytravel/nginx-micro-base	1	-	0.0	▼
e-engine-fpm-mysql-magento	1	-	0.0	▼
image_gallery	1	-	0.0	▼



## 11 Dockerized services

Service	Number of containers	Response time	Requests	Details
PHP on FPM pool www	1	205 ms	6.5 /min	▼
MicrojourneyService	10	124 ms	328 /min	▼
agnaforMicroservices	1	121 ms	328 /min	▼
PHP on 112.110.280	1	19.3 ms	2 /min	▼
redis-weather	1	3.68 ms	26.5 /min	▼
112.110.280	1	0.82 ms	39 /min	▼
madison_island0079	1	8.69 ms	145 /min	▼
easyTravel/insure-Business	1	0.54 ms	857 /min	▼
media.0100	1	0.52 ms	15 /min	▼
img_gallery	1	0.32 ms	18 /min	▼



## Dynamic web requests of MicroJourneyService

Request type: Dynamic requests ✕ [Add filter](#)

Response time

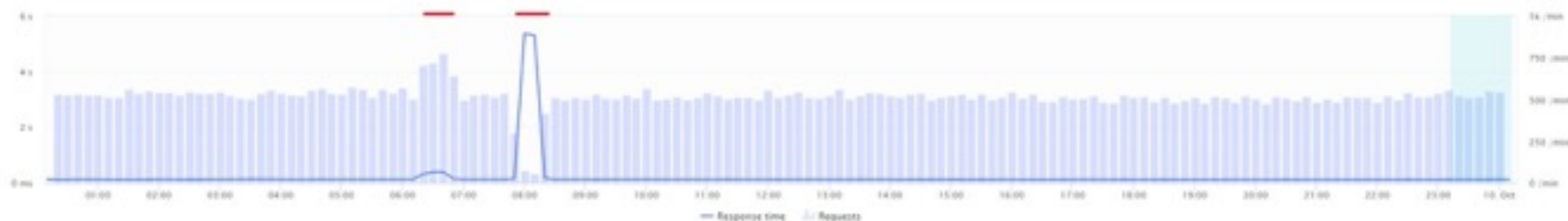
Failures

CPU

Throughput

View [Median](#) [Slowest 10 %](#) [Slowest 5 %](#)

## Server response time

Analyze requests during [1 hour](#) Yesterday, 23:02 - Today 00:02

Analyze code level, database calls, and outgoing requests.

[View response time hotspots](#)

See how response times vary across requests during the selected period.

[Analyze outliers](#)

Looking for more? More analyses and drill-downs...

[More...](#)

Response time of requests sent by monitored services

1 service instance



## Dynamic web requests of MicroJourneyService

Request type: Dynamic requests ✕ [Add filter](#)

Response time

Failures

CPU

Throughput

View [Median](#) [Slowest 10 %](#) [Slowest 5 %](#)

### Server response time



Analyze all requests during this event's time frame, Yesterday, 07:51 - 08:26



Analyze code level, database calls, and outgoing requests.

[Analyze response time degradation](#)

See how response times vary across requests during the selected period.

[Analyze outliers](#)

Looking for more? More analyses and drill-downs...

[More...](#)

Response time of requests sent by monitored services

## www.easytravel.com: User action duration degradation

Problem 30 detected at Oct 9 06:21 - Oct 9 07:02 (was open for 52 minutes).  
This problem affects real users.

	Affected	Recovered	Missed
Applications	-	1	14
Services	-	10	95
Infrastructure components	-	-	681



53,438,070  
Dependencies analyzed

### Business impact analysis

An analysis of all affected service calls and impacted real users during the first 10 minutes of the problem shows the following potential impact.

689 Impacted users 8.4k Affected service calls

[Show more](#)

### 1 impacted application

88.2 User actions per minute impacted

www.easytravel.com  
Application

#### User action duration degradation

The current response time (5.26 s) exceeds the auto-detected baseline (335 ms) by 1385 %

Affected user actions	User action	
88.2/min	3 User actions	
Browser	Session	OS
All	Multiple	All

### One comment

### Root cause

Based on our dependency analysis all incidents have the same root cause:



easytravel-Business  
Database service

#### Response time degradation

The current response time (2.5 s) exceeds the auto-detected baseline (3.78 ms) by 66,025 %

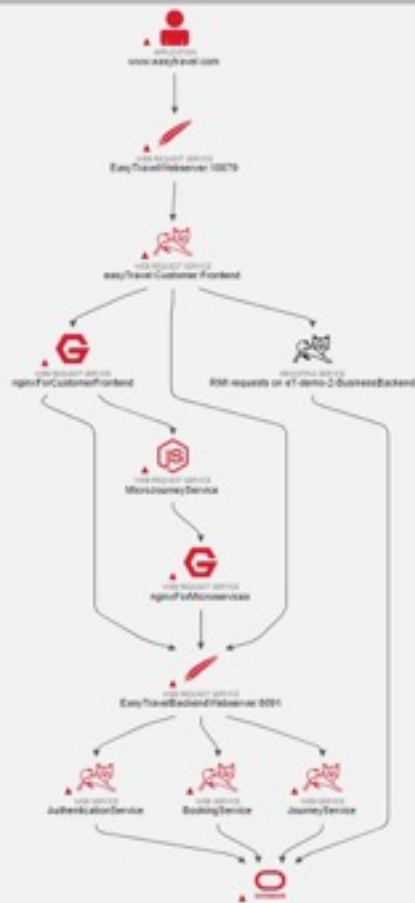
Affected requests: 2,994 /min  
Service method: 8 Service methods



#### Visual resolution path

Click to see how we figured this out.




[Replay](#)

### Problem evolution

Automatically replay problem timeline



2020-10-09 06:20 - 06:25 54 ongoing events in 11 components

Time: 3.56 1/0

#### Application

www.easytravel.com

User action duration degradation (x3)

#### Database

easyTravel-Business

Service method (get verify\_location()) slow down

Service easyTravel-Business slow down (x2)

Service method select journey3\_id as id3, je

Service method select location3\_name as name3, l

Service method select journey3\_id as id3, je

#### Web client service

easyTravel Customer Frontend

Service method services/bookingService/index.js

Service method services/booking/index.js slow d

Service method services/journeyService/index.js

Service method index.js slow down

Service method special-offers.js slow down

Service method services/authenticationService.js

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

Service method services.js slow down

**www.easytravel.com: User action duration degradation**

Problem 30 detected at Oct 9 06:21 - Oct 9 07:02 (was open for 32 minutes).  
This problem affects real users.

	Affected	Recovered	Resolved
Applications	-	1	14
Services	-	10	95
Infrastructure components	-	-	681



53,438,070  
Dependencies analyzed

**Business impact analysis**

An analysis of all affected service calls and impacted real users during the first 10 minutes of the problem shows the following potential impact.

689 Impacted users

8.4k Affected service calls

[Show more](#)

**1 impacted application**

88.2 User actions per minute impacted

**www.easytravel.com**  
Application

**User action duration degradation**

The current response time (3.26 s) exceeds the auto-detected baseline (316 ms) by 1385 %

Affected user actions	User action	
88.2/min	3 User actions	
Browser	Session	OS
All	Multiple	All

**One comment****Root cause**

Based on our dependency analysis all incidents have the same root cause:



**easytravel-Business**  
Database service

**Response time degradation**

The current response time (3.5 s) exceeds the auto-detected baseline (3.78 ms) by 94,025 %

Affected requests: 2,994 /min  
Service method: 8 Service methods

**Visual resolution path**

Click to see how we figured this out.



## www.easytravel.com: User action duration degradation

Problem 30 from Oct 9 06:21 - Oct 9 07:10 for 52 minutes.  
This Service has been identified as root cause.

[Close problem analysis](#)

### Response time degradation (3 events)

#### Response time degradation

The current response time (251 ms) exceeds the auto-detected baseline (224 ms) by 120%.

Affected requests	Service method
101 reqs	SQL Modifications

## easyTravel-Business (Oracle)

Seen recently

[Smartscope view](#)

### Properties and tags



## SQL Modifications


[View database statements](#)

## Database availability



Analyze all requests during this event's time frame. Yesterday: 06:21 - 06:55

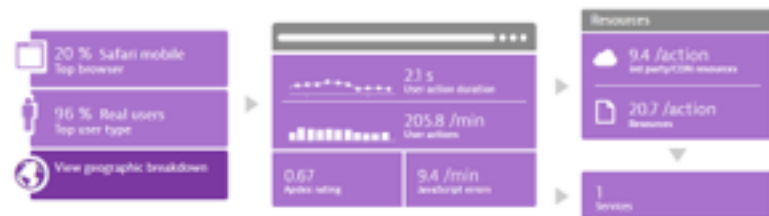
[View backtrace](#)

## Requests contributing to this problem

Slow response time 229 ms  
(call verifyLocation?)

## Performance analysis

Actions, time, Action duration, Apdex rating, JavaScript errors, 3rd party providers and Services.



## User behavior

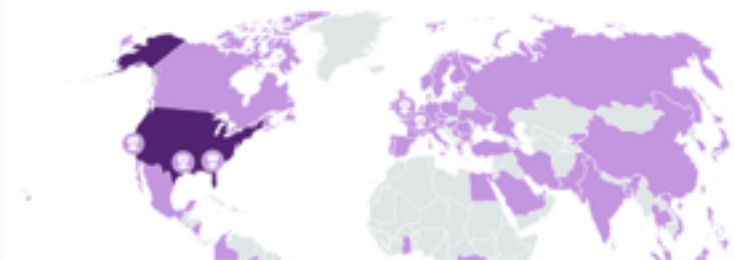
Active sessions, Actions per session, Entry/Exit actions, Bounce rate, and Conversion goals.



## Geo locations

World

By User actions

User actions 205.8 /min  
User actions 0.0 /min 205.8 /min

## 17 Problems Oct 06 2017 07:13 - yesterday 07:13

## 100 % Availability

5 Synthetic checks | 23 Locations | 0 min Total downtime



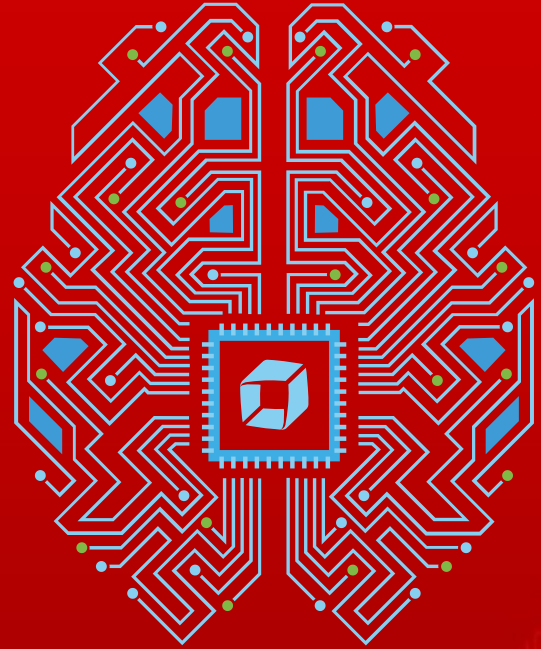
View full details

# Dynatrace

Auto discovery

Auto baselining

Auto problem analysis



# Plus d'informations sur le stand n° 16

« Au fond à gauche »





# 2015



# 2017

## Foncez !

## Mais prenez vos précautions !

**POWERED**  
*by Openshift !*



**SECURED**  
*by Dynatrace !*



RED HAT

**FORUM**

Europe, Middle East & Africa