

# Évaluation de l'impact des fonctionnalités

un outil pour évaluer la consommation de ressources d'une application internet en pré-production

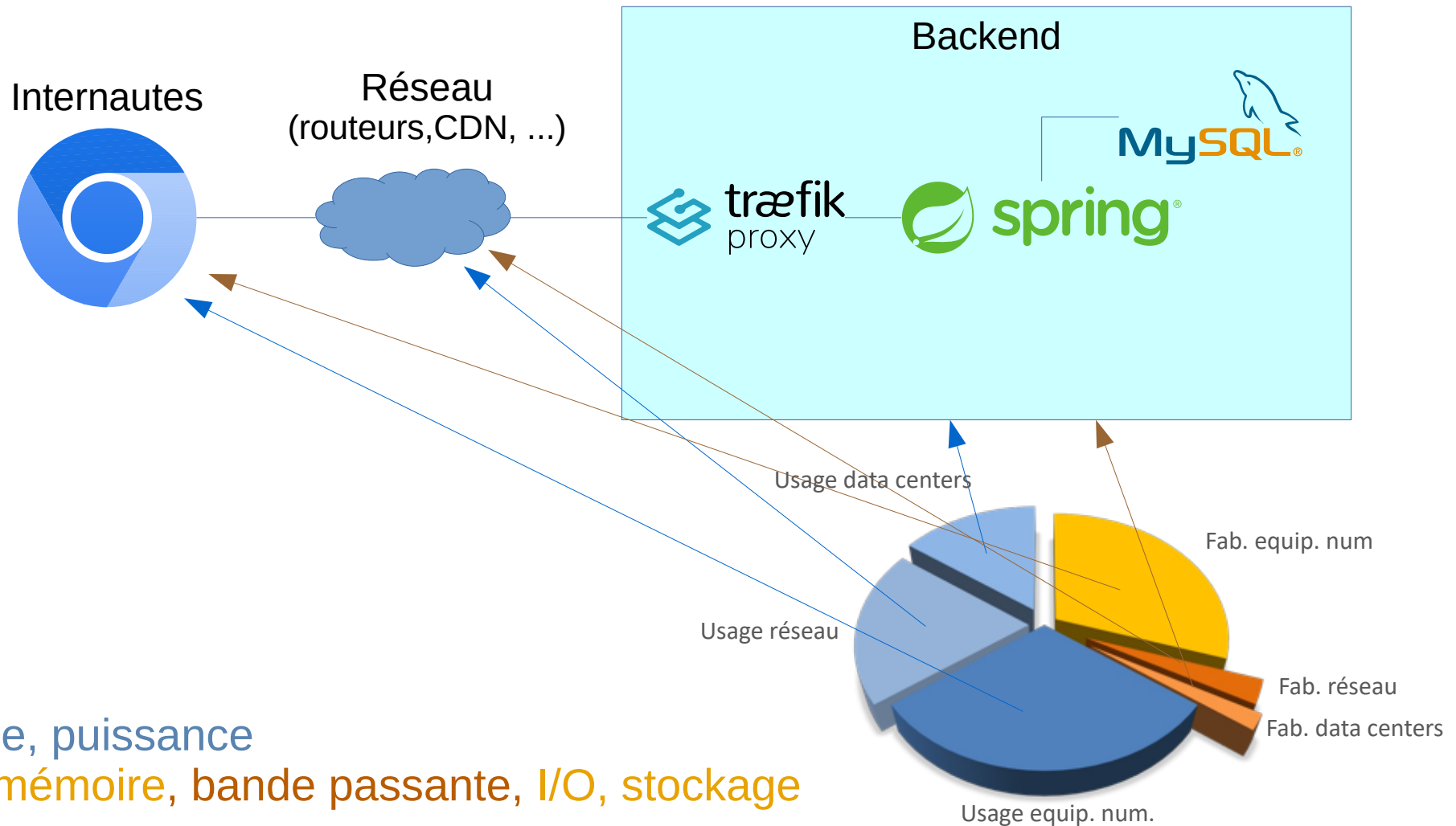
Frédéric Fondement

# Contexte

- Enseignement d'écoconception
  - Système à optimiser
    - Référentiels
    - Outils
  - Application web en Java
- Mesurer les résultats  
*en phase de développement*



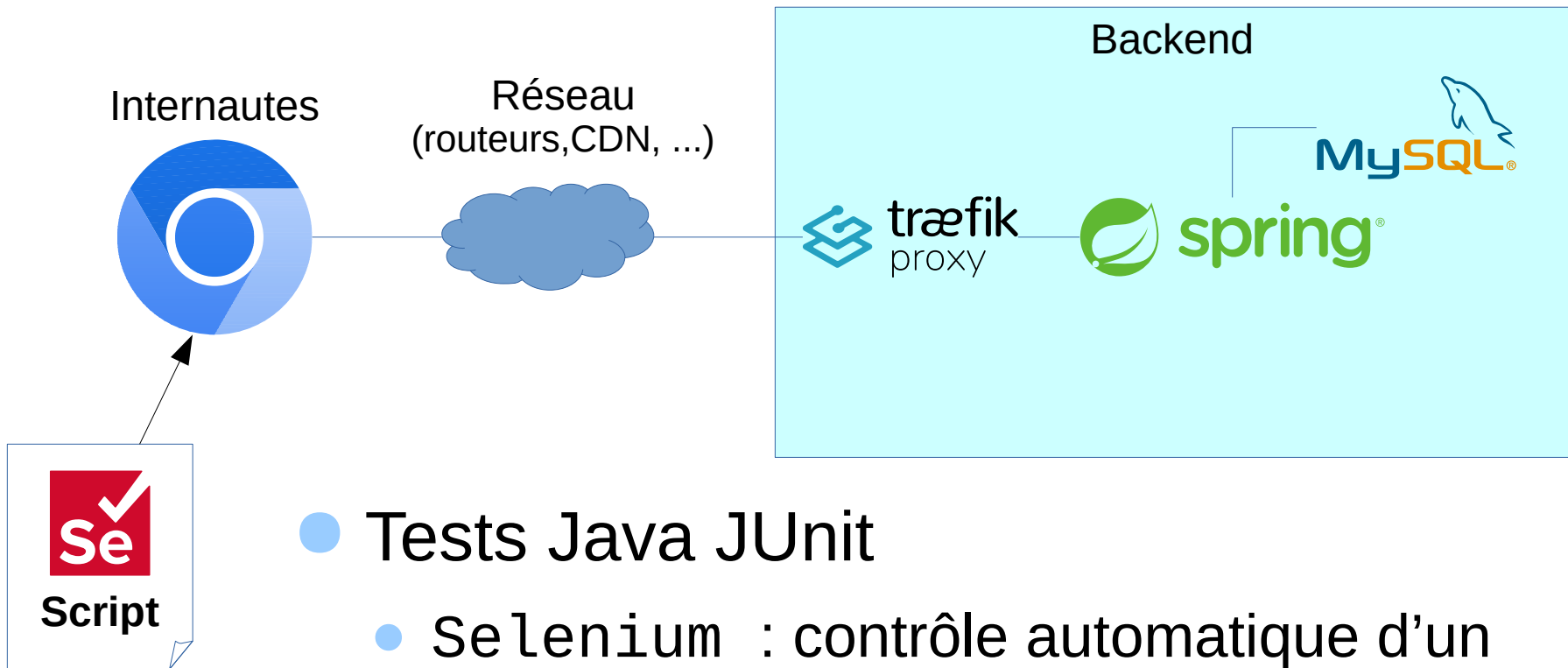
# Mesurer toute la chaîne



Énergie, puissance

CPU, mémoire, bande passante, I/O, stockage

# Tout part de l'internaute



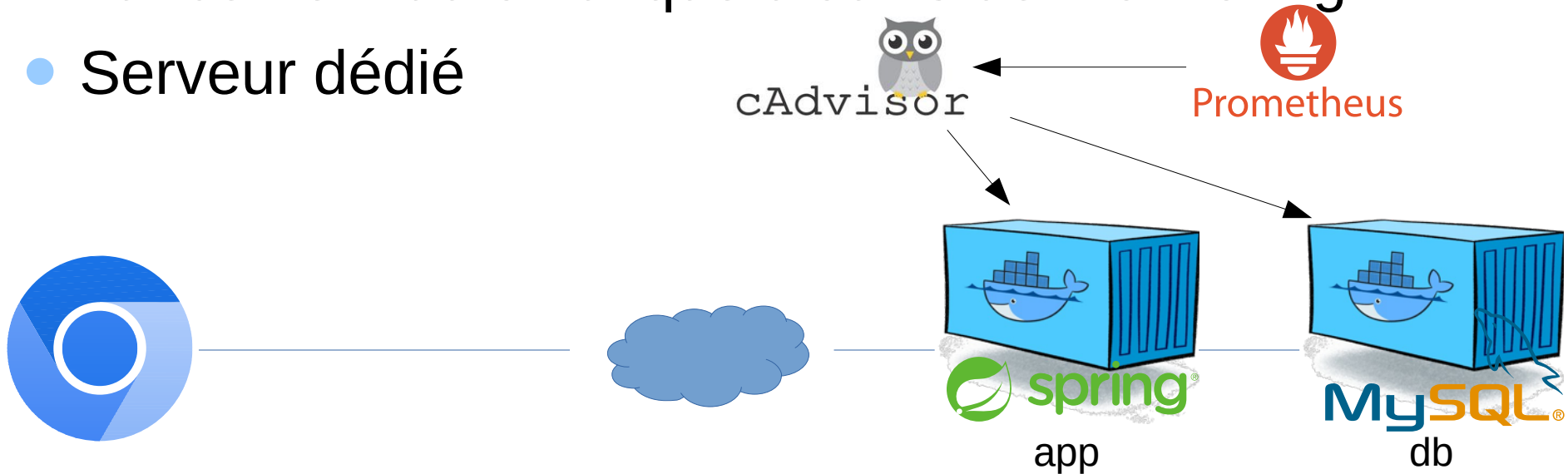
- Tests Java JUnit
  - Selenium : contrôle automatique d'un navigateur (1 internaute)
  - Représentatifs de l'usage de la fonctionnalité mesurée

# Exemple de script

```
public class PremierTest {  
  
    @Test  
    public void papillonnage(){  
        WebDriver wb = new ChromeDriver();  
        wb.get("http://localhost:8080/");  
        wb.executeScript("window.scrollTo(-2000,0)", "");  
        List<WebElement> articles =  
            wb.findElements(By.className("article-btn"));  
        articles.get(3).click();  
        Thread.sleep(TEMPS_PAR_ARTICLE*300);  
        js.executeScript("window.scrollTo(0,500)", "");  
        ...  
    }  
}
```

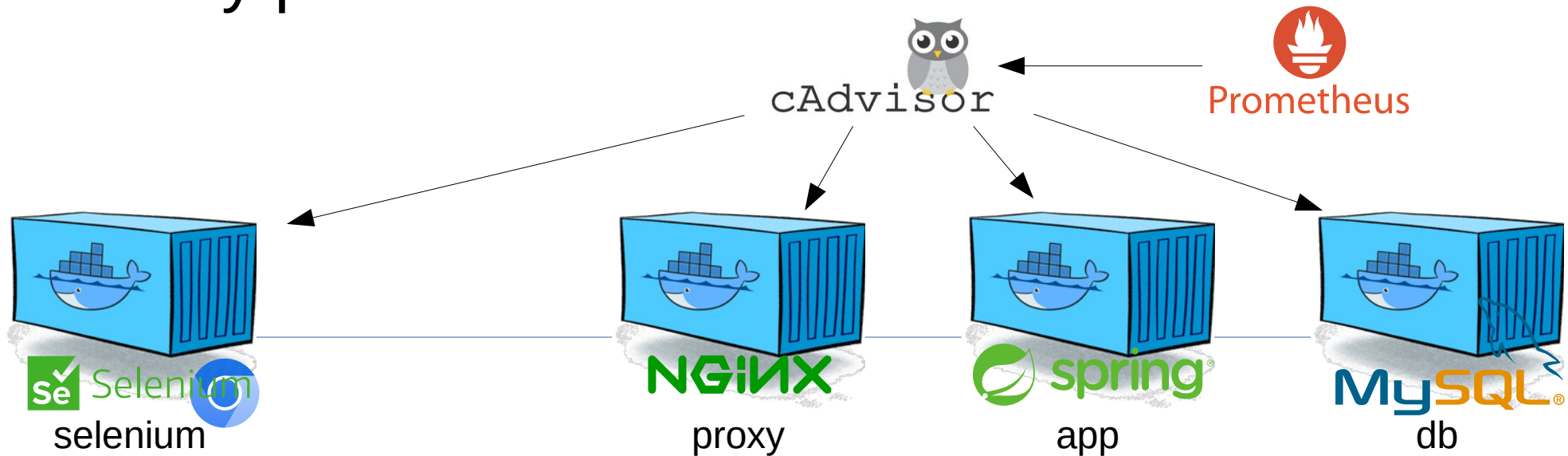
# Backend dans Docker

- Commun chez les développeurs
- Facile à mesurer
  - Lancement automatique d'outils de monitoring
  - Serveur dédié



# Tout dans Docker

- Navigateur
  - Lancé automatiquement par l'outil
- Proxy pour simuler le réseau



# Lancement automatique

```
@EcoWebDriver(remote = true)
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @Test // wb fourni par l'extension
    public void papillonnage(WebDriver wb){
        wb.get("/");
        ...
    }
}
```



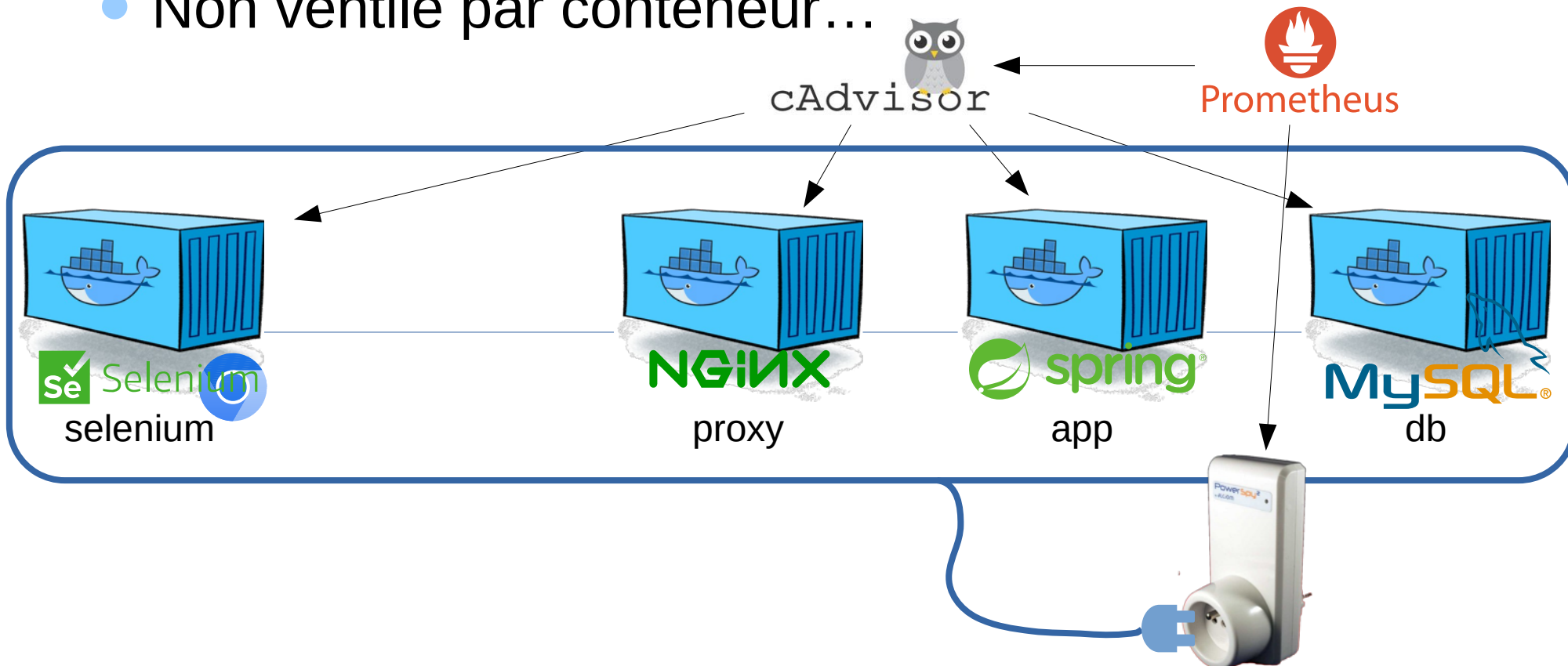
# Conteneurs à surveiller

```
@EcoDockerContainer(id = "anti-demo-proxy-1", port = 8082)
@EcoMonitor(containerId = "anti-demo-app-1")
@EcoMonitor(containerId = "anti-demo-db-1")
@EcoWebDriver(remote = true)
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @Test // wb fourni par l'extension
    public void papillonnage(WebDriver wb){
        wb.get("/");
        ...
    }
}
```

# Pour l'énergie

- Wattmètre connecté
  - Démon docker sur serveur dédié
  - RAPL à venir
  - Non ventilé par conteneur...




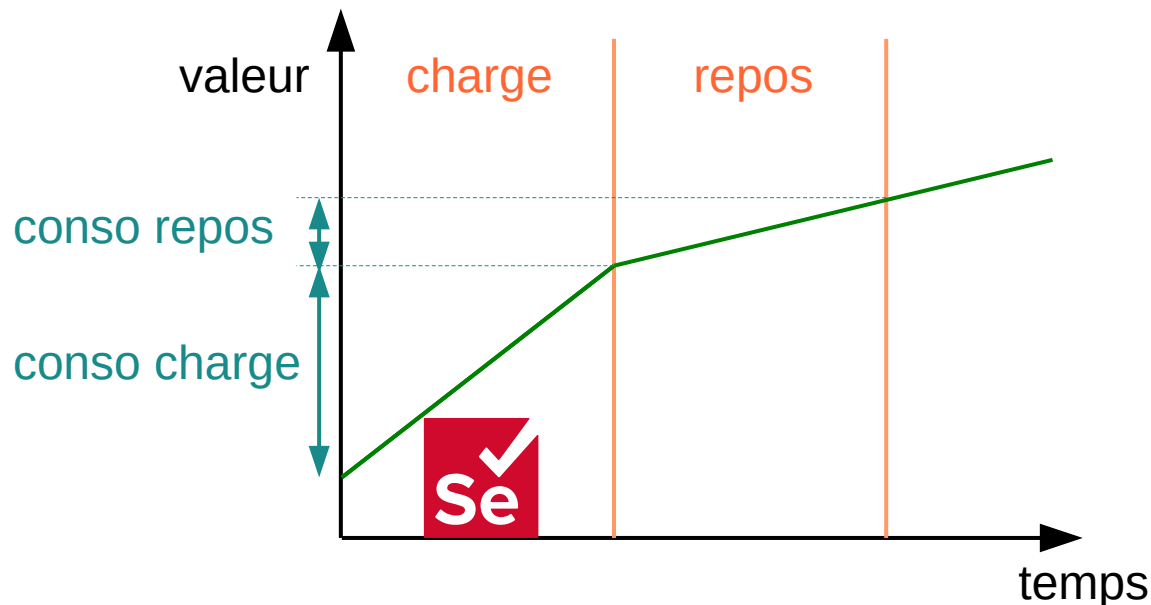
# Compteurs d'énergie

```
@EcoDockerContainer(id = "anti-demo-proxy-1", port = 8082)
@EcoMonitor(containerId = "anti-demo-app-1")
@EcoMonitor(containerId = "anti-demo-db-1")
@EcoWebDriver(remote = true)
@EcoEnergyCounter(type = EcoEnergyCounterType.POWERSPY,
                  name = "Main", endPoint = "http://...")
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @Test
    public void papillonnage(WebDriver wb){
        wb.get("/");
        ...
    }
}
```

# Coût marginal de la fonctionnalité

- Usage de la fonctionnalité dans le script 
- Hypothèse : la fonctionnalité ne consomme que lorsqu'elle est sollicitée
- Le système consomme aussi au repos
- Mesure en charge et au repos
  - Différence de consommation dans les rapports



# Fiabiliser les mesures

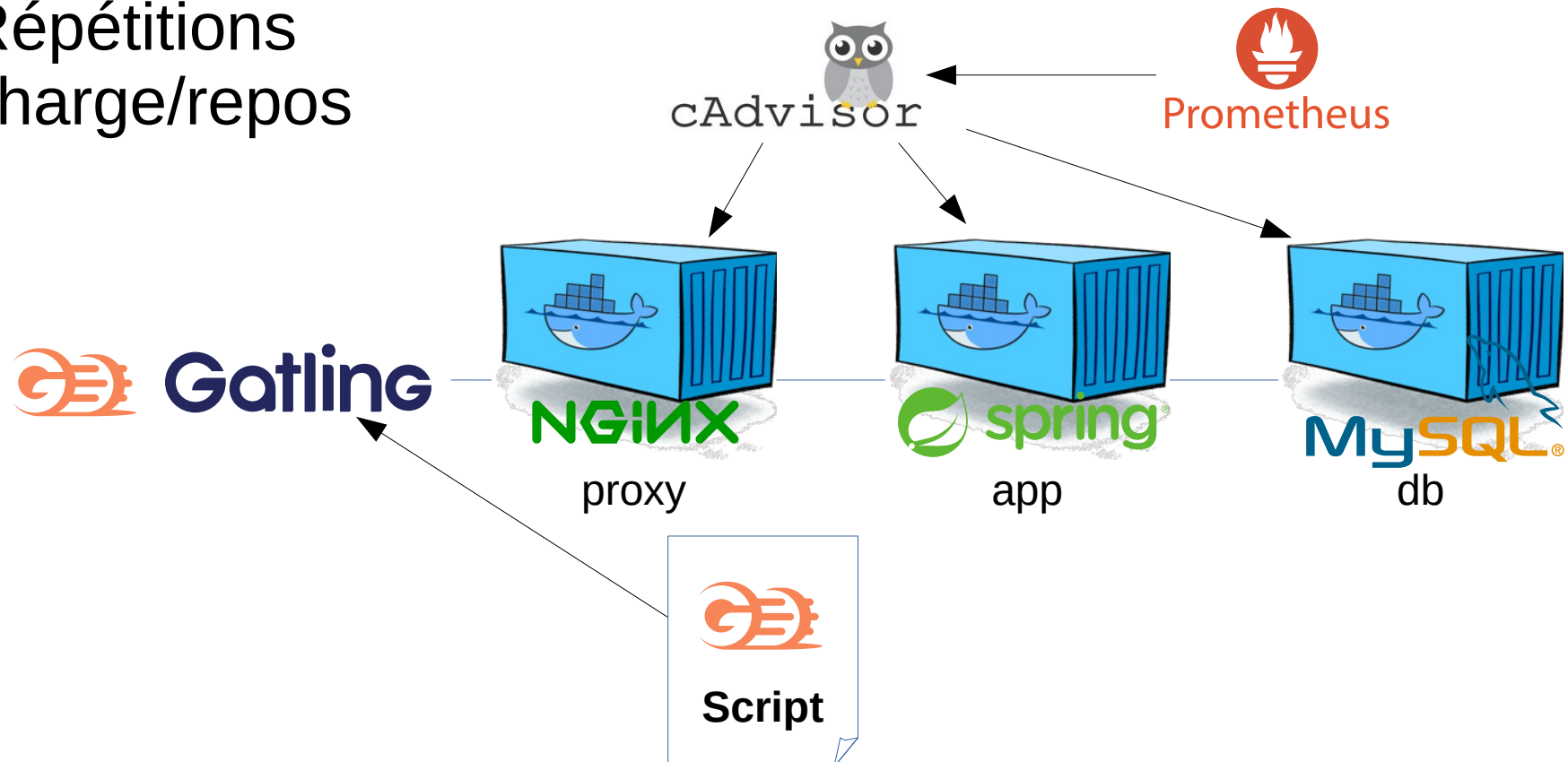
```
@EcoDockerContainer(id = "anti-demo-proxy-1", port = 8082)
@EcoMonitor(containerId = "anti-demo-app-1")
@EcoMonitor(containerId = "anti-demo-db-1")
@EcoWebDriver(remote = true)
@EcoEnergyCounter(type = EcoEnergyCounterType.POWERSPY,
                  name = "Main", endPoint = "http://...")
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @RepeatedTest(10) // agrégation des résultats à la fin
    public void papillonage(WebDriver wb){
        wb.get("/");
        ...
    }
}
```



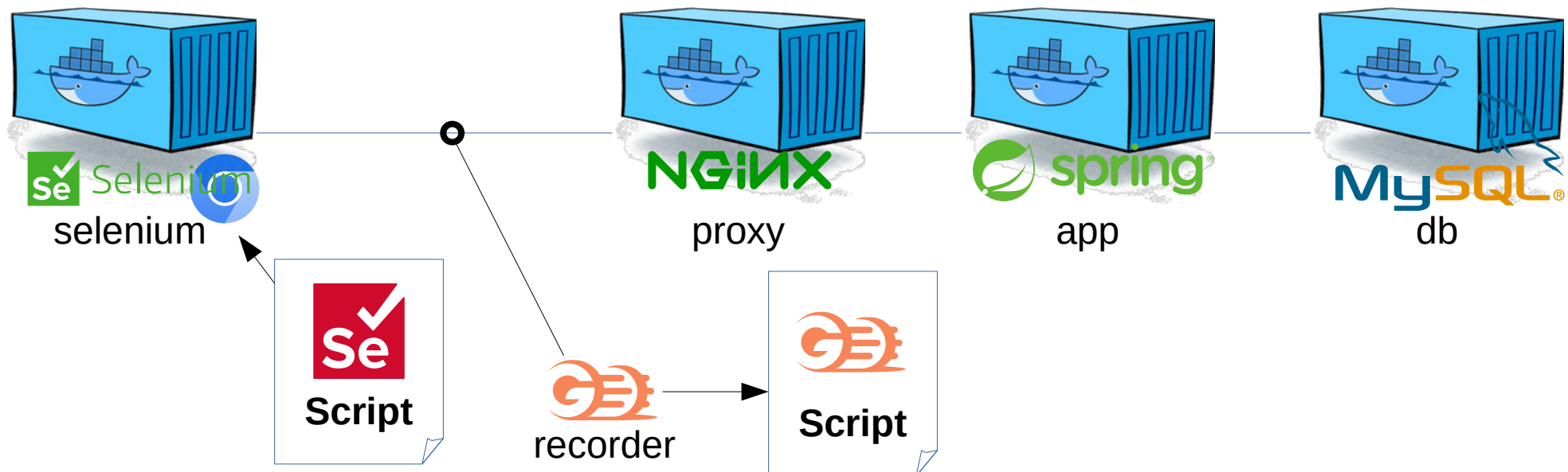
# Backend pas fait pour un seul usager

- Pré-prod : simuler une charge réelle
  - ... et lutter contre le bruit de mesure
- Lancement de tests de montée en charge
  - Répétitions charge/repos



# Générer le script de montée en charge

- Un nouveau script ?
  - Transformation automatique du script Selenium par enregistrement de la première exécution



# Utilisateurs simultanés

```
@EcoDockerContainer(id = "anti-demo-proxy-1", port = 8082)
@EcoMonitor(containerId = "anti-demo-app-1")
@EcoMonitor(containerId = "anti-demo-db-1")
@EcoEnergyCounter(type = EcoEnergyCounterType.POWERSPY,
                  name = "Main", endPoint = "http://...")
@EcoWebDriver(remote = true)
@EcoGatling(userCount = 100, rampDuration = 10)
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @RepeatedTest(10)
    public void papillonage(WebDriver wb){
        wb.get("/");
        ...
    }
}
```



# Résultats : Exemple de la CPU



# Conclusion

- Outil automatisé
  - Applications Web...
  - ... en conteneurs
    - Architectures complètes

## ● Résultats

- Coût marginal d'un changement
  - Optimisation, fonctionnalité, maintenance, ...
- Remontés par CI
  - Développeur prévenu !



## ● Exigences

- Scripts JUnit/Selenium
- Gatling local
- Serveur docker
  - Distant et dédié
- (Wattmètre connecté)

# Perspectives

- Meilleurs rapports
  - Agrégation des métriques en une seule
    - Ex : eqCO<sub>2</sub> → Base de faits
  - Comparaison
- Prise en considération de systèmes physiques contrôlés
  - Jumeaux numériques

# Liens

- Extension JUnit

<https://gitlab.cluster.ensisa.uha.fr/ecoconception/metrologie>

- Projet exemple

[https://gitlab.cluster.ensisa.uha.fr/ecoconception/antidemo\\_public](https://gitlab.cluster.ensisa.uha.fr/ecoconception/antidemo_public)

- Remerciements

Maxime KUHN

Alexis GERBER

Albi GRAINCA

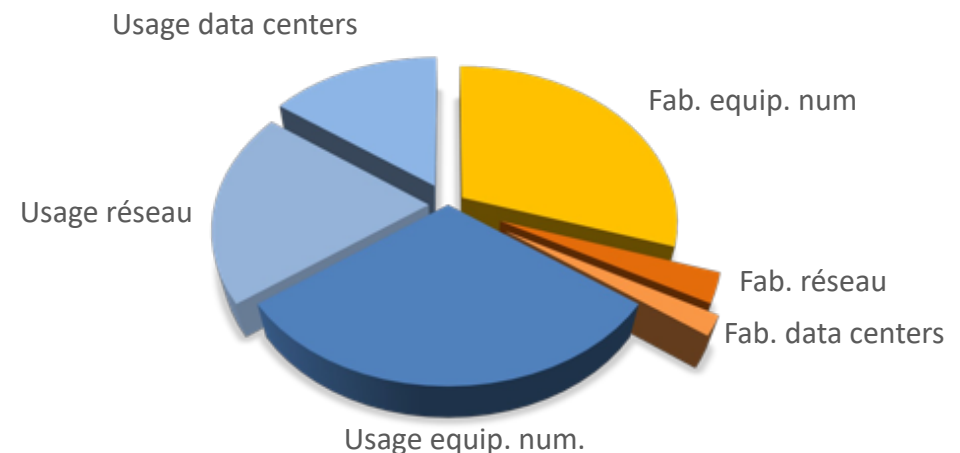
Batuhan GÖKER

Imad YASSIM

# Annexes

# Métriques considérées

- Utilisation CPU, mémoire, bande passante, I/O, stockage
  - Proxys pour l'impact de la fabrication
  - Proxys pour la durabilité du matériel
- Énergie, puissance
  - Proxys pour l'impact de l'usage



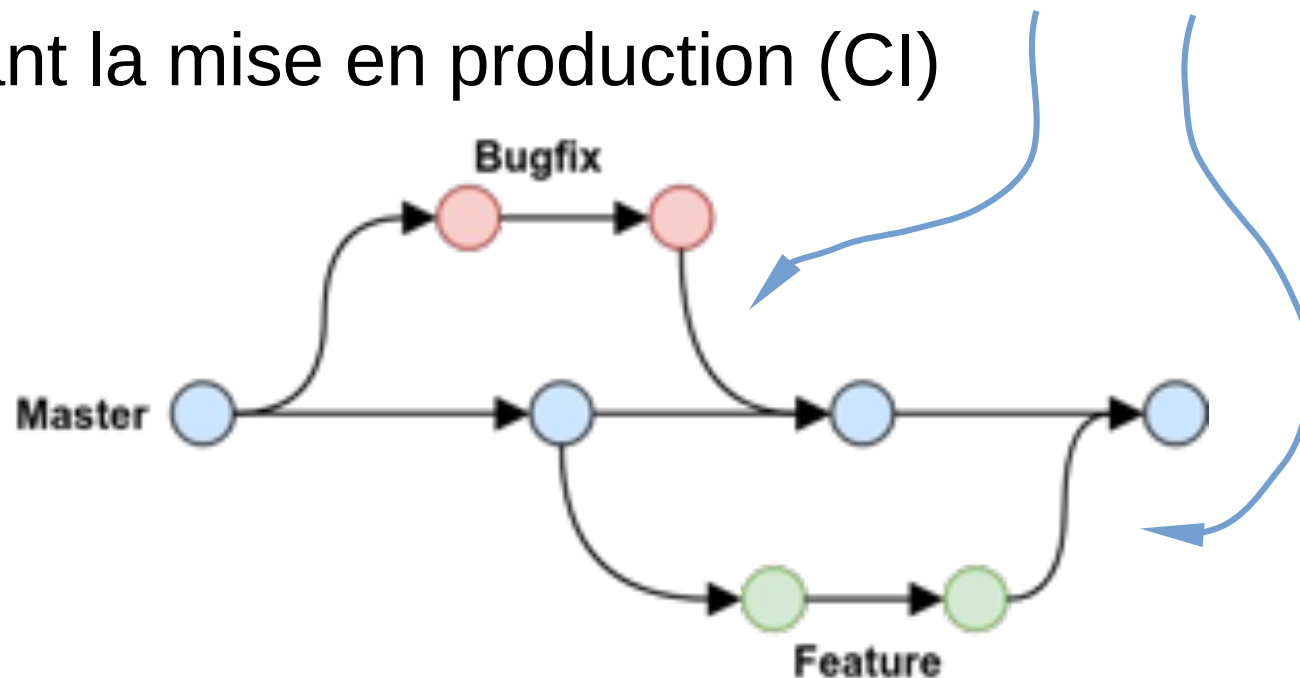
# Répliquabilité de la mesure d'un système

- Mesures bruitées
  - Répéter les mesures
    - Dans les mêmes conditions
  - Augmenter la charge
    - Applications web
      - augmenter le nombre d'utilisateurs simulés
  - Conditions de mesure à maîtriser
    - Serveur dédié
    - Installation minimale



# Développement logiciel

- Utilisation de référentiel de code (ex : git)
  - Système original sur une branche
  - Système complété/optimisé sur une autre  
=> comparaison de consommation de ressources
  - Avant la mise en production (CI)





# Développement logiciel

- Utilisation d'outils de build (ex : maven)
  - Compile, teste, empaquette, exécute le système et lance les tests d'intégration, ...
- Utilisation d'intégration continue (ex : jenkins)
  - Lance l'outil de build à chaque mise à jour
    - Sur chaque branche
    - Comparaison des métriques remontées
      - Ex : couverture de code (surveillance de la qualité)
- Idée : intégrer la mesure de consommation de ressources dans le build
  - Permet l'intégration dans l'intégration continue

# Automatisation

- Extension JUnit 5
  - Définition
    - Du scénario (selenium webdriver fourni)
    - Des conteneurs à mesurer
    - Des répétitions et de la montée en charge
  - Lancement/arrêt automatique des outils
    - Connexion à un démon Docker
      - Distant et dédié de préférence (perturber le moins possible la mesure)
    - cAdvisor, Prometheus, Selenium Server, Gatling Recorder, Gatling (en local)
  - Récupération des métriques
    - Stockées en CSV
    - Agrégations HTML et prom

# Exemple : gestion du navigateur

```
@EcoDockerContainer(id = "anti-demo-proxy-1", port = 8082)
@EcoMonitor(containerId = "anti-demo-app-1")
@EcoMonitor(containerId = "anti-demo-db-1")
@EcoWebDriver(remote = true)
@EcoEnergyCounter(type = EcoEnergyCounterType.POWERSPY,
                  name = "Main", endPoint = "$POWERSPY_HOST$")
@EcoGatling(userCount = 50, rampDuration = 10)
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @RepeatedTest(5)
    public void papillonnage(WebDriver wb){
        JavascriptExecutor js = (JavascriptExecutor) wb;
        wb.get("/");
        ...
    }
}
```

Port d'entrée du système

Injecté dans Docker

Injecté dans le test

Hôte et port « deviné » des paramètres précédents

# Exemple : Wattmètre

```
@EcoDockerContainer(id = "anti-demo-proxy-1", port = 8082)
@EcoMonitor(containerId = "anti-demo-app-1")
@EcoMonitor(containerId = "anti-demo-db-1")
@EcoWebDriver(remote = true)
@EcoEnergyCounter(type = EcoEnergyCounterType.POWERSPY,
                  name = "Main", endPoint = "$POWERSPY_HOST$")
@EcoGatling(userCount = 50, rampDuration = 10)
@ExtendWith(EcoExtension.class)
public class PremierTest {

    @RepeatedTest(5)
    public void papillonnage(WebDriver wb){
        JavascriptExecutor js = (JavascriptExecutor) wb;
        wb.get("/");

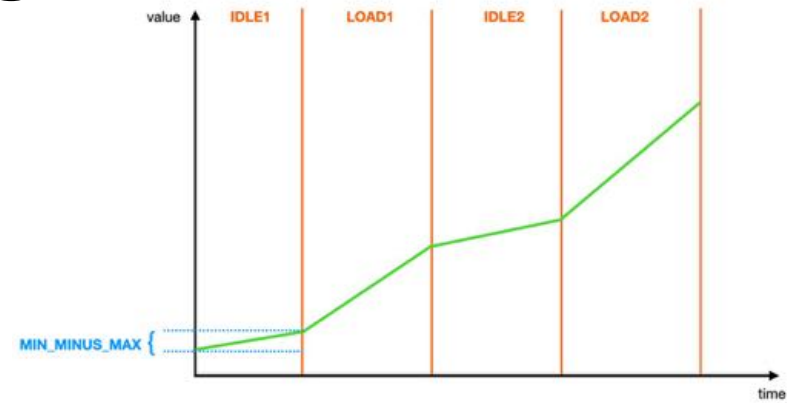
        ...
    }
}
```

Variable d'environnement

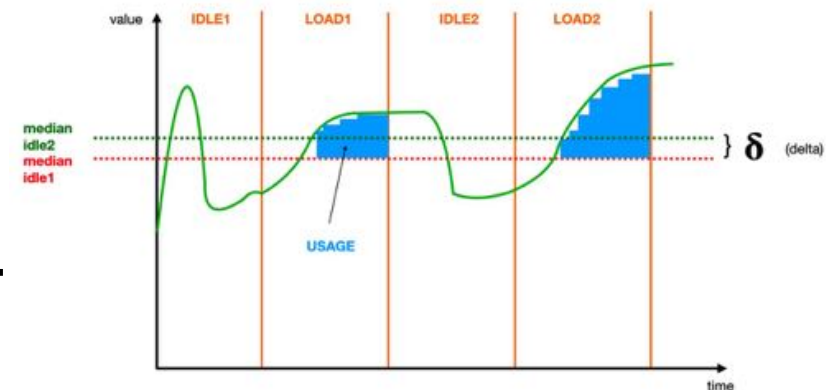
Export HTTP au format Prometheus

# Exploitation des résultats

- Métriques « compteurs »
  - Monotones croissantes
    - Usage de la CPU, Énergie, ...



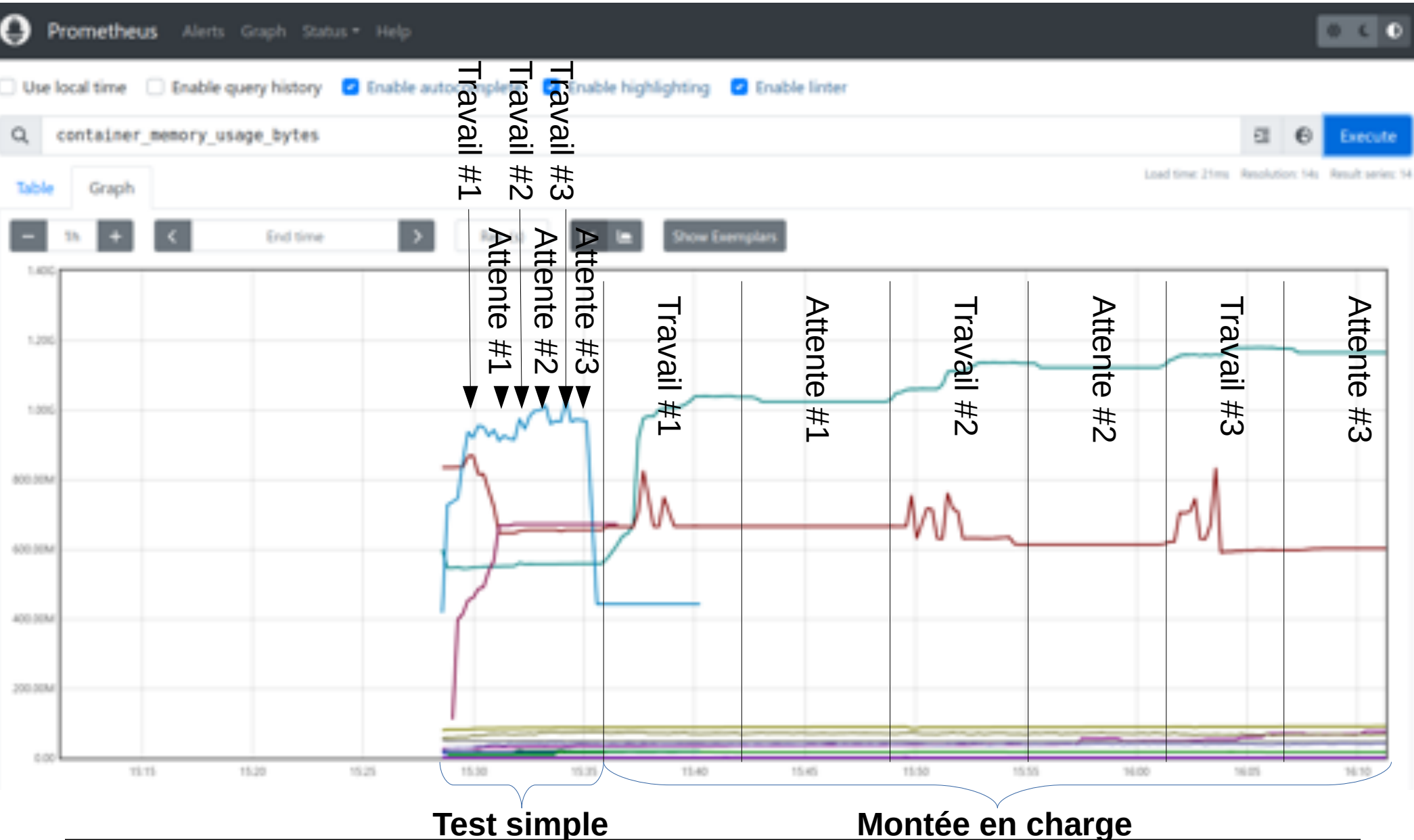
- Métriques « jauges »
  - Libres
    - Mémoire, Puissance appelée, ...



| Type     | Agrégation dans une phase | Agrégation attente / charge | Agrégation entre répétitions |
|----------|---------------------------|-----------------------------|------------------------------|
| Compteur | Dernière - Première       | Différence                  | Moyenne + Écart-type         |
| Jauge    | Médiane                   |                             |                              |

**Mesure simple** → **Navigateur**  
**Montée en charge** → **Backend**

# Utilisation Mémoire



# Intégration au CI/CD Gitlab

image: `maven:3-eclipse-temurin-17`

variables:

`DOCKER_HOST: "tcp://10.0.10.10:2375"`

start\_containers:

image: `docker`

script:

- `cd anti-demo`
- `docker compose up -d --build`

resources\_test:

needs:

- `start_containers`

variables:

`GATLING_VERSION: "3.9.5"`

script:

- `cd testbench`
- `. downloadGatling.sh`
- `export GATLING_HOME=/usr/share/gatling`
- `mvn -U verify`

artifacts:

paths:

- `**/target/ecoconception/**`

reports:

junit:

- `**/target/surefire-reports/TEST-*.xml`

metrics:

- `**/target/ecoconception/*/metrics.prom`

# Repetition - 1 (IDLE)

| Metric label                        | Container name    | Aggregation Type | Value      |
|-------------------------------------|-------------------|------------------|------------|
| container_cpu_usage_seconds_total   | anti-demo-app-1   | MIN_MINUS_MAX    | 0.71859    |
| container_cpu_usage_seconds_total   | anti-demo-db-1    | MIN_MINUS_MAX    | 1.31937    |
| container_cpu_usage_seconds_total   | anti-demo-proxy-1 | MIN_MINUS_MAX    | 0          |
| container_memory_usage_bytes        | anti-demo-app-1   | MEDIAN           | 1690439680 |
| container_memory_usage_bytes        | anti-demo-db-1    | MEDIAN           | 822784000  |
| container_memory_usage_bytes        | anti-demo-proxy-1 | MEDIAN           | 10133504   |
| container_network_total_bytes_total | anti-demo-app-1   | MIN_MINUS_MAX    | 0          |
| container_network_total_bytes_total | anti-demo-db-1    | MIN_MINUS_MAX    | 0          |
| container_network_total_bytes_total | anti-demo-proxy-1 | MIN_MINUS_MAX    | 0          |
| energy_joules                       | Main              | MIN_MINUS_MAX    | 1770       |
| power_rms_milliwatts                | Main              | MEDIAN           | 21323      |



# Repetition - 1 (LOAD)

| Metric label                        | Container name    | Aggregation Type | Value       |
|-------------------------------------|-------------------|------------------|-------------|
| container_cpu_usage_seconds_total   | anti-demo-app-1   | MIN_MINUS_MAX    | 105.16595   |
| container_cpu_usage_seconds_total   | anti-demo-db-1    | MIN_MINUS_MAX    | 37.67394    |
| container_cpu_usage_seconds_total   | anti-demo-proxy-1 | MIN_MINUS_MAX    | 19.83412    |
| container_memory_usage_bytes        | anti-demo-app-1   | MEDIAN           | 1690441728  |
| container_memory_usage_bytes        | anti-demo-db-1    | MEDIAN           | 819955712   |
| container_memory_usage_bytes        | anti-demo-proxy-1 | MEDIAN           | 40091648    |
| container_network_total_bytes_total | anti-demo-app-1   | MIN_MINUS_MAX    | 14630332753 |
| container_network_total_bytes_total | anti-demo-db-1    | MIN_MINUS_MAX    | 7548485242  |
| container_network_total_bytes_total | anti-demo-proxy-1 | MIN_MINUS_MAX    | 14597159665 |
| energy_joules                       | Main              | MIN_MINUS_MAX    | 4249        |
| power_rms_milliwatts                | Main              | MEDIAN           | 43822       |

# Repetition - 1

| Metric label                        | Container name    | Aggregation Type | Value        |
|-------------------------------------|-------------------|------------------|--------------|
| container_cpu_usage_seconds_total   | anti-demo-app-1   | DELTA            | 104.44736    |
| container_cpu_usage_seconds_total   | anti-demo-db-1    | DELTA            | 36.35457     |
| container_cpu_usage_seconds_total   | anti-demo-proxy-1 | DELTA            | 19.83412     |
| container_memory_usage_bytes        | anti-demo-app-1   | USAGE            | -18820407296 |
| container_memory_usage_bytes        | anti-demo-db-1    | USAGE            | -515846144   |
| container_memory_usage_bytes        | anti-demo-proxy-1 | USAGE            | 3548381184   |
| container_network_total_bytes_total | anti-demo-app-1   | DELTA            | 14630332753  |
| container_network_total_bytes_total | anti-demo-db-1    | DELTA            | 7548485242   |
| container_network_total_bytes_total | anti-demo-proxy-1 | DELTA            | 14597159665  |
| energy_joules                       | Main              | DELTA            | 2479         |
| power_rms_milliwatts                | Main              | USAGE            | 22499        |

| Metric label                        | Container name    | Aggregation Type | Value            |
|-------------------------------------|-------------------|------------------|------------------|
| container_cpu_usage_seconds_total   | anti-demo-app-1   | MEAN             | 88.77252         |
|                                     |                   | STD              | 8.80199          |
| container_cpu_usage_seconds_total   | anti-demo-db-1    | MEAN             | 36.0822          |
|                                     |                   | STD              | 0.23901          |
| container_cpu_usage_seconds_total   | anti-demo-proxy-1 | MEAN             | 19.31916         |
|                                     |                   | STD              | 0.36036          |
| container_memory_usage_bytes        | anti-demo-app-1   | MEAN             | -4971575705.6    |
|                                     |                   | STD              | 7795747831.91428 |
| container_memory_usage_bytes        | anti-demo-db-1    | MEAN             | -261454233.6     |
|                                     |                   | STD              | 670973196.12415  |
| container_memory_usage_bytes        | anti-demo-proxy-1 | MEAN             | 4506088243.2     |
|                                     |                   | STD              | 687037309.19779  |
| container_network_total_bytes_total | anti-demo-app-1   | MEAN             | 14634053746      |
|                                     |                   | STD              | 2363144.90063    |
| container_network_total_bytes_total | anti-demo-db-1    | MEAN             | 7552170520.6     |
|                                     |                   | STD              | 2314715.86648    |
| container_network_total_bytes_total | anti-demo-proxy-1 | MEAN             | 14608522678.2    |
|                                     |                   | STD              | 6854095.41758    |
| energy_joules                       | Main              | MEAN             | 2124.8           |
|                                     |                   | STD              | 198.04595        |
| power_rms_milliwatts                | Main              | MEAN             | 25276.6          |
|                                     |                   | STD              | 1555.78495       |

# Perspectives plus complètes

- Plus de métriques (RAPL, stockage, I/Os, ..)
- Meilleurs rapports
  - Séparer les rapports démarrage à froid / à chaud
  - Agrégation des jauges (mémoire)
  - Comparaison entre différentes branches
  - Agrégation des métriques en une seule par script (ex : eqCO<sub>2</sub>)
    - Base de faits
- Détermination automatique des paramètres (répétitions, utilisateurs)
- Notion de protocole lors de la montée en charge
- Prise en charge de systèmes plus complexes
  - Tâches en arrière plan
- Prise en considération de systèmes physiques contrôlés
  - Jumeaux numériques