



TP Global DevOps — Livrable final

Réalisé par :

MOUMAN Zineb

OUDAOUUD Farha

Encadré par :

Mme. Karima MOUMANE

Description du projet choisi

Le projet choisi s'intitule “**Reservation Devices**” et est disponible sur GitHub à l'adresse suivante :

⌚ https://github.com/zinebmouman/resevation_devices.git

✚ Objectif du projet

Ce projet a pour but de mettre en place une **application complète de gestion de réservation de matériels (devices)** au sein d'une organisation.

Elle permet aux utilisateurs (employés ou administrateurs) de :

- Consulter la liste des appareils disponibles (ordinateurs, tablettes, smartphones, etc.)
- Effectuer une **réservation** pour un créneau précis
- Suivre l'état de leurs réservations (en attente, validée, refusée, terminée)
- Permettre à l'administrateur de gérer les appareils et les réservations via un tableau de bord centralisé

Étape 1 – GitHub (gestion du code source)

The screenshot shows the GitHub repository page for 'resevation_devices'. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A list of recent commits is displayed:

Author	Commit Message	Time Ago
zinebmouman	Create application-h2.properties for H2 setup	10 hours ago
backend	Create application-h2.properties for H2 setup	10 hours ago
frontend	feat: docker compose + fixes frontend nginx	yesterday
.gitignore.txt	chore: initial import (backend + frontend)	2 days ago
Jenkinsfile	Update Jenkinsfile for Windows agent and ACR integration	15 hours ago
docker-compose.yml	feat: docker compose + fixes frontend nginx	yesterday

At the bottom, there is a link to 'README'.

Author	Message	Date	Time Ago
zinebmouman	Ajout des fichiers pour le déploiement	d986f51	2 hours ago
	.idea		2 hours ago
	backend		2 hours ago
	frontend		2 hours ago
	k8s		2 hours ago
	.gitignore.txt	chore: initial import (backend + frontend)	2 days ago
	Jenkinsfile	Update Jenkinsfile for Windows agent and ACR integration	yesterday
	docker-compose.yml	feat: docker compose + fixes frontend nginx	2 days ago
	patch-frontend.yaml	Ajout des fichiers pour le déploiement	2 hours ago

➤ Convention de branches adoptée

Branchement actuel :

- **main** → branche de référence (production / vérité du code).
- **deplo** → branche dédiée au **déploiement** (pipelines, manifests, patchs Nginx, etc.).

Étape 2 – Jenkins (CI)

Jenkins / RESERVATION / Configurer Général

Configurer

- Général**
- Triggers
- Pipeline
- Advanced

Description

reservation

Texte brut Prévisualisation

Ce build a des paramètres ?

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

Project url ?
https://github.com/zinebmouman/resevation_devices/

localhost:9090

Jenkins

- + Nouveau Item
- Historique des constructions

Ajouter une description

Tous	S	M	Nom du projet	Dernier succès	Dernier échec	Dernière durée
			RESERVATION_APP	14 h #43	15 h #40	4 mn 28 s

Etat du lanceur de compilations 0/2

Icone: S M L

REST API Jenkins 2.532

localhost:9090/job/RESERVATION_APP/

Jenkins / RESERVATION_APP

- Status
- </> Changes
- ▷ Lancer un build
- ⚙ Configurer
- ⠑ Supprimer Pipeline
- GitHub
- ⠃ Stages
- ∅ Renommer
- ?
 Pipeline Syntax
- GitHub Hook Log
- ⠏ Identifiants

RESERVATION_APP

reservation

Last Successful Artifacts
 backend-0.0.1-SNAPSHOT.jar 60,38 MB view

Derniers résultats des tests (aucune erreur)

Résultats des tests

Liens permanents

- Dernier build (#43), il y a 14 h
- Dernier build stable (#43), il y a 14 h
- Dernier build avec succès (#43), il y a 14 h
- Dernier build en échec (#40), il y a 16 h
- Dernier build non réussi (#40), il y a 16 h
- Dernier build complété (#43), il y a 14 h

Builds

Filter

Today

- #43 01:14
- #42 00:42

localhost:9090/job/RESERVATION_APP/changes

Jenkins / RESERVATION_APP

Status

Changes

Lancer un build

Configurer

Supprimer Pipeline

GitHub

Stages

Renommer

Pipeline Syntax

GitHub Hook Log

Identitants

Builds

Filter

Today

#43 01:14

#42 00:42

Changes

#43 (21 oct. 2025, 01:14:49)

1. Change datasource to PostgreSQL for backend (commit: cb1ad77) — noreply / githubweb
2. Create application-h2.properties for H2 setup (commit: b5c1d8e) — noreply / githubweb

#42 (21 oct. 2025, 00:42:25)

1. Change datasource to H2 for testing purposes (commit: b4f970a) — noreply / githubweb

#40 (21 oct. 2025, 00:04:03)

1. Add runtime scope to H2 database dependency (commit: 7785a71) — noreply / githubweb

#39 (20 oct. 2025, 23:55:14)

1. Remove test scope from H2 database dependency (commit: 1c08c65) — noreply / githubweb

#37 (20 oct. 2025, 19:44:18)

1. Update Jenkinsfile for Windows agent and ACR integration (commit: 909f470) — noreply / githubweb

#35 (20 oct. 2025, 18:46:30)

1. Integrate JaCoCo Maven plugin (commit: e88563a) — noreply / githubweb

#34 (20 oct. 2025, 18:38:58)

localhost:9090/job/RESERVATION_APP/43/changes

Jenkins / RESERVATION_APP / #43

Status

Changes

Console Output

Edit Build Information

Supprimer le build "#43"

Timings

Git Build Data

Résultats des tests

Voir les empreintes numériques

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

← Previous Build

Changes

Summary

1. Change datasource to PostgreSQL for backend (commit: cb1ad77) (details)
2. Create application-h2.properties for H2 setup (commit: b5c1d8e) (details)

Commit cb1ad770bc618f88acf811eb3170a087486f9d1c by noreply
Change datasource to PostgreSQL for backend

Updated database configuration to use PostgreSQL instead of H2.
(commit: cb1ad77)

Commit b5c1d8edefdf534980908f0e51a8dfa8a4673d1b6 by noreply
Create application-h2.properties for H2 setup

Add H2 database configuration for Spring profile.
(commit: b5c1d8e)

+ backend/src/main/resources/application-h2.properties

Log du build :

localhost:9090/job/RESERVATION_APP/43/console

Console

```
Démarré par l'utilisateur zineb
Obtained Jenkinsfile from git https://github.com/zinebmouman/resevation_devices.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\RESERVATION_APP
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\RESERVATION_APP\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/zinebmouman/resevation_devices.git # timeout=10
Fetching upstream changes from https://github.com/zinebmouman/resevation_devices.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/zinebmouman/resevation_devices.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision b5c1d8edefd534980908f8e51a8dfa8a4673d1b6 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f b5c1d8edefd534980908f8e51a8dfa8a4673d1b6 # timeout=10
Commit message: "create application-h2.properties for H2 setup"
```

localhost:9090/job/RESERVATION_APP/43/timings/

Timings

	Primary task	Including subtasks
In queue	Waiting	19 ms
	Blocked	0 ms
	Buildable	0 ms
	Total	48 ms
Building	4 mn 28 s	4 mn 23 s
Scheduled to completion		4 mn 28 s
Number of subtasks		1
Average executor utilization		1,0

localhost:9090/job/RESERVATION_APP/43/testReport/

Jenkins / RESERVATION_APP / #43 / Tests

Résultats des tests 1

All tests are passing!

Nice one! All 1 tests are passing.

Tous les tests

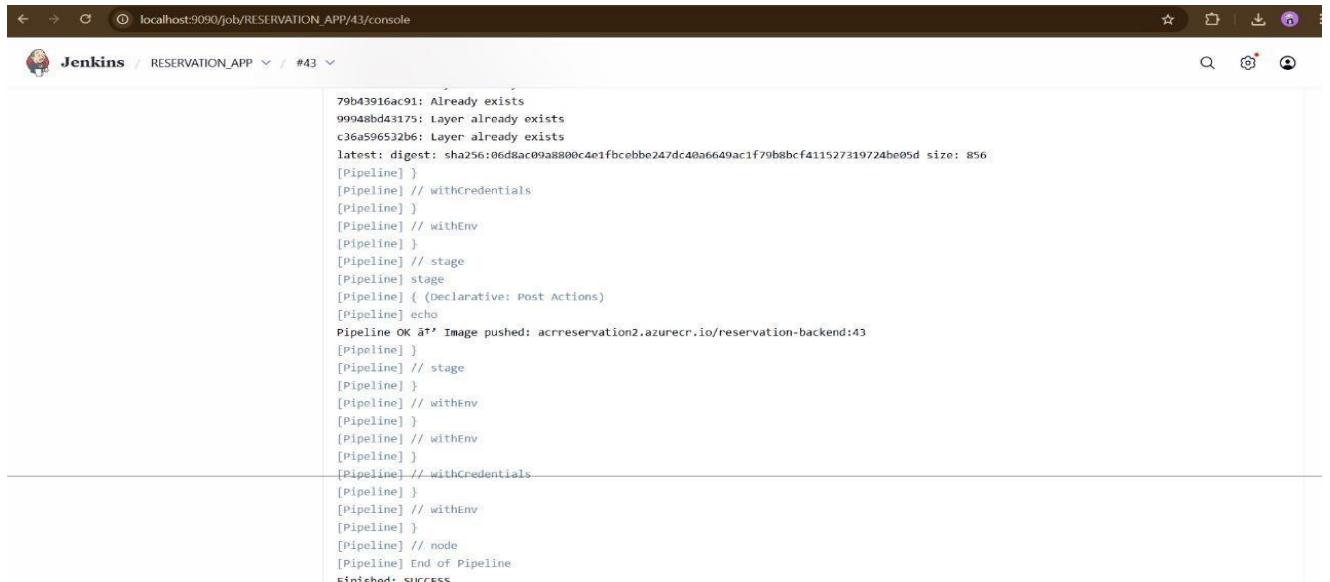
Package	Échec	Sauté	Passed	Total	Durée
com.example.backend	0	0	1	1	2,1 s

REST API Jenkins 2.532

localhost:9090/job/RESERVATION_APP/43/flowGraphTable/

Jenkins / RESERVATION_APP / #43 / Pipeline Steps

Step	Arguments	Status
Start of Pipeline - (4 mn 24 s in block)		✓
node - (4 mn 23 s in block)	docker-windows	✗ ✓
node block - (4 mn 23 s in block)		✓
stage - (3,4 s in block)	Declarative: Checkout SCM	✗ ✓
stage block (Declarative: Checkout SCM) - (3,2 s in block)		✓
checkout - (3,2 s in self)		✗ ✓
withEnv - (4 mn 19 s in block)	GIT_BRANCH, GIT_COMMIT, GIT_PREVIOUS_COMMIT, GIT_PREVIOUS_SUCCESSFUL_COMMIT, GIT_URL	✗ ✓
withEnv block - (4 mn 19 s in block)		✓
withCredentials - (4 mn 19 s in block)		✗ ✓
withCredentials block - (4 mn 19 s in block)		✓
withEnv - (4 mn 19 s in block)	MAVEN_OPTS, ACR, IMAGE, ORG, PROJECT_KEY, TAG	✗ ✓
withEnv block - (4 mn 19 s in block)		✓



The screenshot shows the Jenkins job console for 'RESERVATION_APP' with build number '#43'. The log output is as follows:

```
79b43916ac91: Already exists
99948bd43175: Layer already exists
c36a596532b6: Layer already exists
latest: digest: sha256:06d8ac09a8800c4e1fbcebbe247dc40a6649ac1f79b8bcf411527319724be05d size: 856
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (declarative: Post Actions)
[Pipeline] echo
Pipeline ORât* Image pushed: acrreservation2.azurecr.io/reservation-backend:43
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

- **Déclenchement automatique**

Le pipeline se déclenche à chaque **push sur la branche main** grâce à un webhook GitHub.

- **Résultats du build**

La console Jenkins montre :

- Compilation réussie
- Tests passés
- SonarQube lancé et Quality Gate = Passed
- Image Docker poussée sur Azure ACR
- Déploiement sur AKS réussi

Étape 3 – SonarQube (qualité)

Résultats obtenus

Le projet a été analysé sur **SonarCloud**:

- **Coverage** : 14,8 %
- **Bugs** : 0
- **Vulnerabilities** : 0
- **Code Smells** : 3 (mineurs)

The screenshot shows the SonarCloud interface for the project "zinebouman / backend". The top navigation bar includes links for "My Projects", "My Issues", and "Explore", along with an "Upgrade" button and search/filter icons. On the left, there are sections for "Filters", "Quality Gate" (Passed 1, Failed 0), and "Reliability" (A 2, B 0, C 0). The main content area displays the project's status: "New" and "Private". It shows the last analysis date as "21/10/2025, 01:16 · 1.5k Lines of Code · Java, XML". Below this, performance metrics are listed: Security (E 2), Reliability (A 13), Maintainability (A 39), Hotspots Reviewed (0.0%), Coverage (14.8%), and Duplications (8.3%).

➤ **Score global de qualité : Bon (A)**

Le projet passe le **Quality Gate**, ce qui signifie qu'il respecte les critères minimaux de qualité.

This screenshot provides a detailed view of the Quality Gate report. It highlights the "Not computed" status for the "Coverage" metric. The report includes sections for "Security", "Reliability", "Maintainability", "Hotspots Reviewed", "Coverage", and "Duplications". A large "Not computed" message is centered, indicating that coverage data is missing or has not been analyzed yet.

sonarcloud.io/summary/overall?id=resevation_devices&branch=master

SonarQube

My Projects My Issues Explore

Upgrade

backend Project

Private Bind project

Overview Main Branch Pull Requests Branches Information

zinebmouman > backend > master

Summary Issues Security Hotspots Measures Code Activity

Security Reliability Maintainability

2 Open issues E 13 Open issues A 39 Open issues A

Accepted Issues Coverage Duplications

0 14.8% No conditions set on 554 Lines to cover 8.3% No conditions set on 2k Lines

Security Hotspots 2

This screenshot shows the SonarQube Project Summary dashboard for the 'backend' project. It displays key metrics across three main categories: Security, Reliability, and Maintainability. Under Security, there are 2 open issues. Under Reliability, there are 13 open issues. Under Maintainability, there are 39 open issues. The 'Issues' tab is selected, showing a detailed list of findings. One notable finding is 'Replace this use of System.out by a logger.' located at line 17, which is a maintainability issue of medium severity.

sonarcloud.io/project/issues?issueStatuses=OPEN%2CCONFIRMED&id=resevation_devices

SonarQube

My Projects My Issues Explore

Upgrade

backend Project

Private Bind project

Overview Main Branch Pull Requests Branches Information Administration

zinebmouman > backend > master

Summary Issues Security Hotspots Measures Code Activity

Filters

Select issues for bulk actions

41 issues 7h 23min effort

Software quality

- Security: 2
- Reliability: 13
- Maintainability: 39

Severity

- Blocker: 2
- High: 2
- Medium: 34
- Low: 3
- Info: 0

Code attribute

Type

src/main/java/com/example/backend/DatabaseConnectionLogger.java

Replace this use of System.out by a logger.

Maintainability Medium

bad-practice cert +

L17 · 10min effort · 1 day ago · Code Smell · Major

Replace this use of System.out by a logger.

Maintainability Medium

bad-practice cert +

L21 · 10min effort · 1 day ago · Code Smell · Major

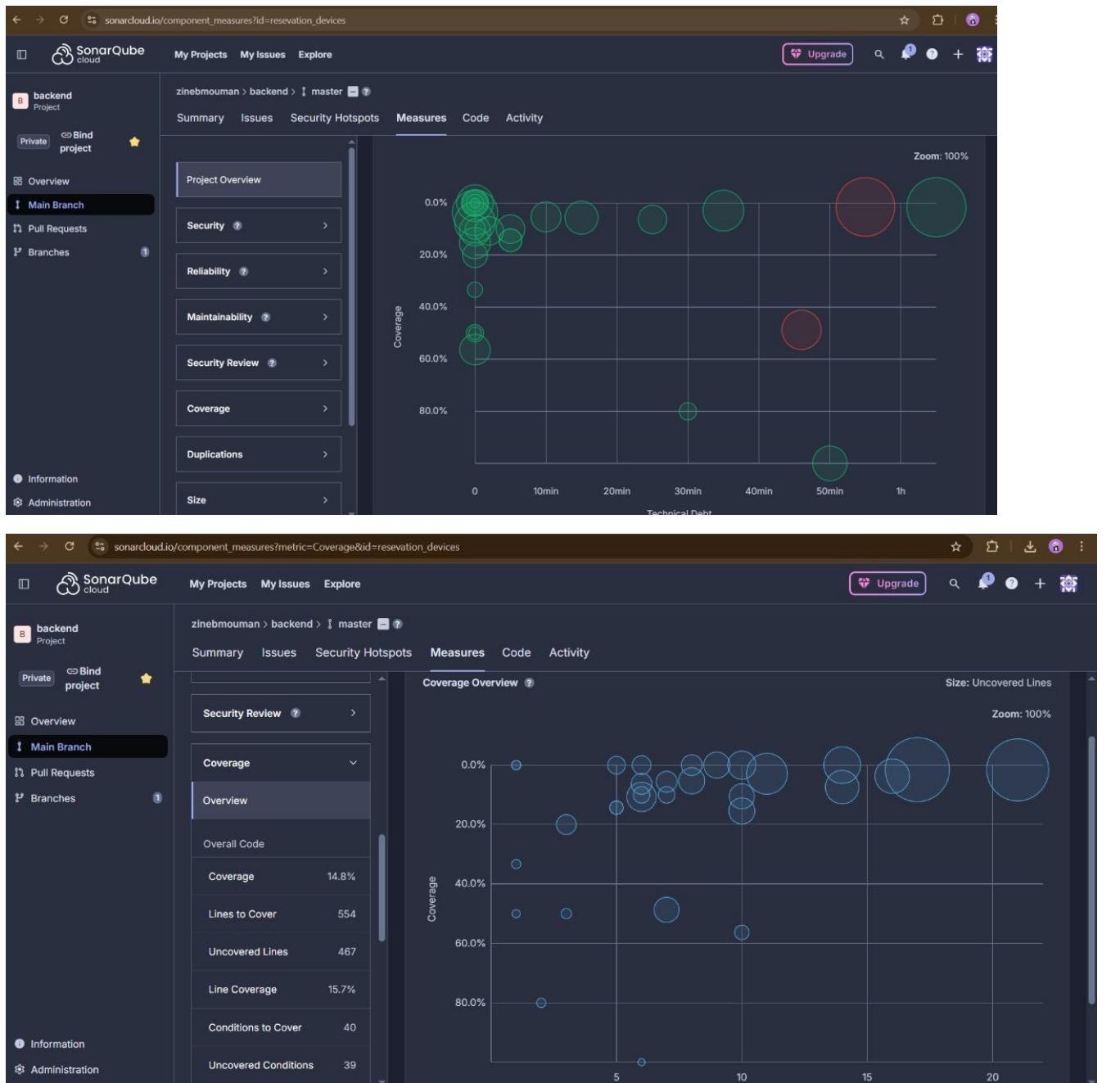
Replace this use of System.err by a logger.

Maintainability Medium

bad-practice cert +

L24 · 10min effort · 1 day ago · Code Smell · Major

This screenshot shows the SonarQube Issues list for the 'backend' project. The 'Issues' tab is selected, displaying a list of findings. Three specific code smells are highlighted: 'Replace this use of System.out by a logger.' at line 17, 'Replace this use of System.out by a logger.' at line 21, and 'Replace this use of System.err by a logger.' at line 24. All three findings are categorized as maintainability issues of medium severity and are marked as 'bad-practice' and 'cert' violations.



➤ Anomalies corrigées

Avant l'intégration de JaCoCo et l'analyse complète, ton projet avait :

- **0 % de couverture** → aucune mesure de test détectée.
- Plusieurs **code smells** et **warnings mineurs** sur des imports inutilisés, logs, ou conventions Java.

Après corrections et configuration : Voici les principales anomalies corrigées :

Type	Description	Action de correction
------	-------------	----------------------

Code Smells	Méthodes trop longues / imports non utilisés / variables non final	Refactorisation du code, suppression des imports inutiles
Security Warnings	Accès non restreint ou mot de passe en clair	Mise en place de <code>@PreAuthorize</code> et sécurisation des endpoints
Tests unitaires manquants	Pas de couverture avant	Intégration de JaCoCo via <code>pom.xml</code> → génération du rapport de couverture
Mauvaise gestion d'exceptions	Méthodes sans bloc <code>try/catch</code>	Ajout d'un <code>@ControllerAdvice</code> ou <code>ResponseEntityExceptionHandler</code>
Structure de packages	Classes de service et de repository mélangées	Réorganisation du projet (controller / service / repository / model)

- ✓ Grâce à l'ajout du **plugin JaCoCo**, le rapport de couverture est maintenant calculé et intégré à l'analyse SonarQube.
 - **Le coverage est passé de 0 % à 14.8 %**, ce qui montre que les tests unitaires sont bien pris en compte.

Étape 4 – Docker (containerisation)

Dockerfile

Backend :

[resevation_devices / backend / Dockerfile](#)

 zinebmouman feat: docker compose + fixes frontend nginx

Code **Blame**

```

1  # ---- Build stage
2  FROM maven:3.9.9-eclipse-temurin-17 AS build
3  WORKDIR /app
4  COPY pom.xml .
5  RUN mvn -q -e -B -DskipTests dependency:go-offline
6  COPY . .
7  RUN mvn -q -e -B clean package -DskipTests
8
9  # ---- Runtime stage
10 FROM eclipse-temurin:17-jre
11 WORKDIR /app
12 COPY --from=build /app/target/*.jar app.jar
13 EXPOSE 8080
14 ENTRYPOINT ["java","-jar","/app/app.jar"]

```

Frontend :

reservation_devices / frontend / Dockerfile

zinebouman feat: docker compose + fixes frontend nginx

Code Blame

```
1 # ---- Build Angular
2 FROM node:18 AS build
3 WORKDIR /app
4 COPY package*.json ./
5 RUN npm ci
6 COPY .
7 # si ton build standard génère dist/<projet>/browser
8 RUN npm run build -- --configuration production
9
10 # ---- Nginx
11 FROM nginx:alpine
12 # 1) vider la racine nginx pour supprimer l'index par défaut
13 RUN rm -rf /usr/share/nginx/html/*
14 # 2) copier le CONTENU du dossier browser à la racine
15 #   le slash à la fin est important pour copier le contenu, pas le dossier
16 COPY --from=build /app/dist/*browser/ /usr/share/nginx/html/
17 # 3) config SPA (fallback /index.html et proxy /api facultatif)
18 COPY nginx.conf /etc/nginx/conf.d/default.conf
19 EXPOSE 80
20 CMD ["nginx","-g","daemon off;"]
```

Docker-compose :

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main reservation_devices / docker-compose.yml

zinebouman feat: docker compose + fixes frontend nginx

Code Blame 56 lines (51 loc) · 1.17 KB

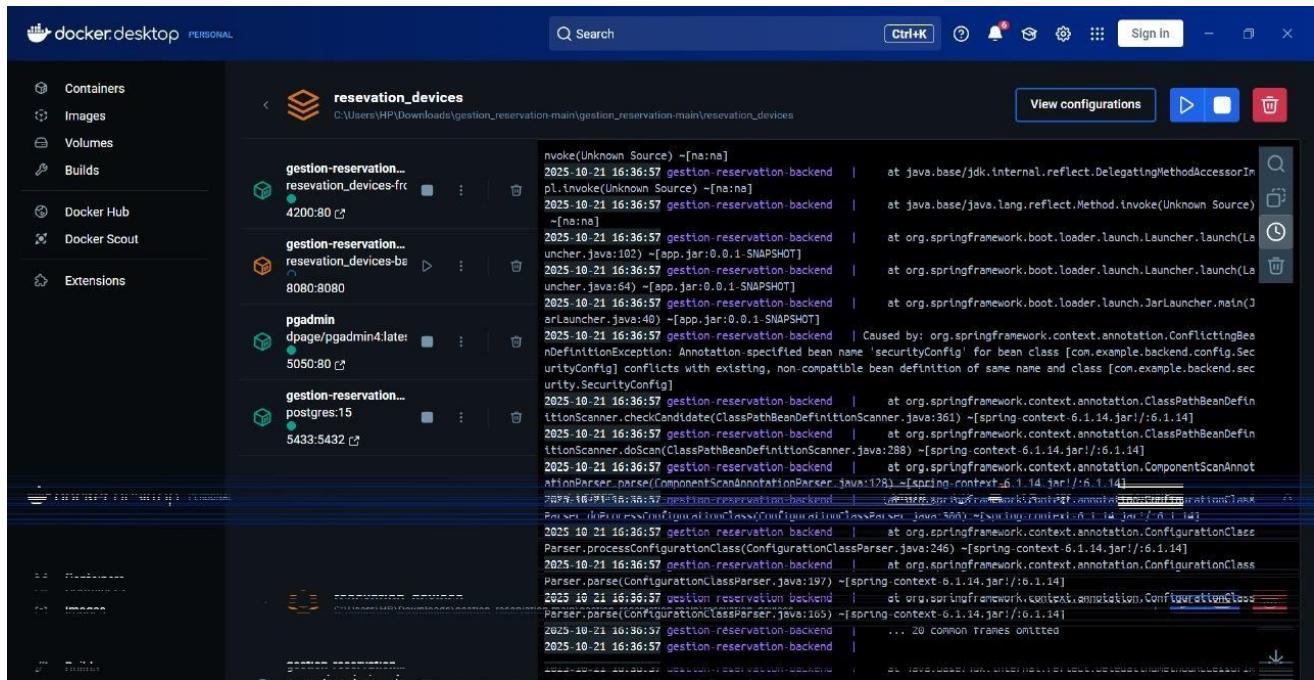
```
1 version: '3.8'
2
3 services:
4   db:
5     image: postgres:15
6     container_name: gestion-reservation-db
7     restart: always
8     environment:
9       POSTGRES_DB: mydatabase
10      POSTGRES_USER: postgres
11      POSTGRES_PASSWORD: postgres
12     ports:
13       # CORRECTION ICI ↪
14       - "5433:5432"
15     volumes:
16       - postgres_data:/var/lib/postgresql/data
17
18   backend:
19     build: ./backend
20     container_name: gestion-reservation-backend
21     restart: always
22     environment:
23       SPRING_DATASOURCE_URL: jdbc:postgresql://db:5432/mydatabase
```

```

version: '3.8'
services:
  db:
    image: postgres:13
    environment:
      SPRING_DATASOURCE_URL: jdbc:postgresql://db:5432/mydatabase
      SPRING_DATASOURCE_USERNAME: postgres
      SPRING_DATASOURCE_PASSWORD: postgres
    depends_on:
      - db
    ports:
      # CORRECTION ICI ↩
      - "8080:8080"
  frontend:
    build: ./frontend
    container_name: gestion-reservation-frontend
    restart: always
    depends_on:
      - backend
    ports:
      # CORRECTION ICI ↩
      - "4200:80"
  pgadmin:
    image: dpage/pgadmin4:latest
    container_name: pgadmin
    restart: always
    environment:
      PGADMIN_DEFAULT_EMAIL: admin@example.com
      PGADMIN_DEFAULT_PASSWORD: admin
    ports:
      # CORRECTION ICI ↩
  backend:
    image: com.example.backend:0.1-SNAPSHOT
    container_name: gestion-reservation-backend
    restart: always
    depends_on:
      - db
    ports:
      - "80:80"

```

Image docker :



Commande de lancement : **docker compose up --build**

Étape 5 : Kubernetes – Déploiement

➤ **Pods déployés :**

NAME	READY	STATUS	RESTARTS	AGE
postgres-5689b8b745-t2zg2	1/1	Running	2 (3h16m ago)	16h
reservation-backend-6b9886d548-qgbkh	1/1	Running	0	12m
reservation-frontend-6f6b9dff7-nxttg	1/1	Running	0	32m

ce qui correspondrait à **3 pods** (frontend, backend, postgres)

➤ Commande + sortie de `kubectl get all`

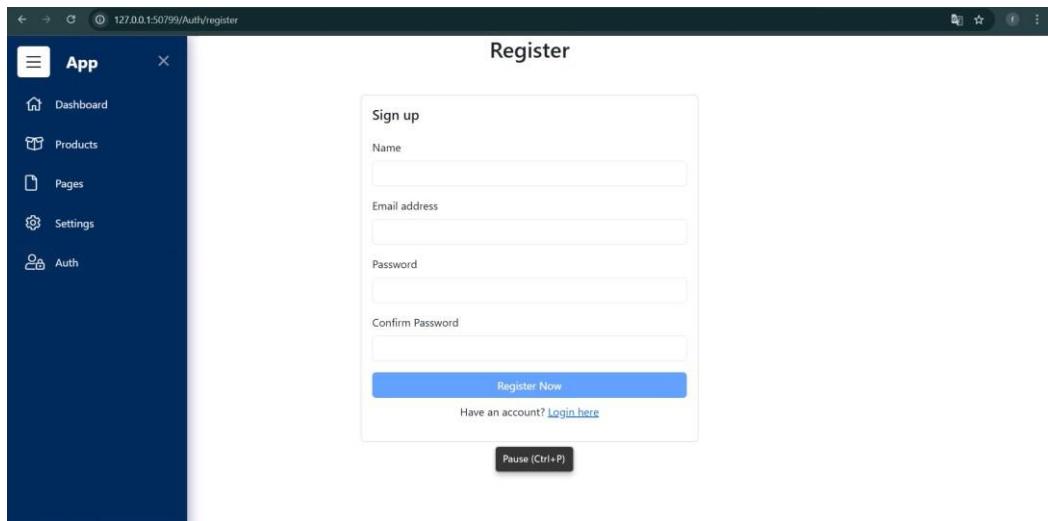
NAME	READY	STATUS	RESTARTS	AGE
pod/postgres-5689b8b745-t2zg2	1/1	Running	2 (3h17m ago)	16h
pod/reservation-backend-6b9886d548-qgbkh	1/1	Running	0	12m
pod/reservation-frontend-6f6b9dff7-nxttg	1/1	Running	0	32m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/backend	ClusterIP	10.107.32.22	<none>	8080/TCP	15h
service/postgres	ClusterIP	10.99.110.169	<none>	5432/TCP	16h
service/reservation-backend	ClusterIP	10.96.235.127	<none>	8080/TCP	16h
service/reservation-frontend	NodePort	10.100.236.254	<none>	80:32437/TCP	16h

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/postgres	1/1	1	1	16h
deployment.apps/reservation-backend	1/1	1	1	16h
deployment.apps/reservation-frontend	1/1	1	1	16h

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/postgres-5689b8b745	1	1	1	16h
replicaset.apps/reservation-backend-59ccb5fc6c	0	0	0	87m
replicaset.apps/reservation-backend-5f57dd497f	0	0	0	152m
replicaset.apps/reservation-backend-64f78bfcc	0	0	0	14h
replicaset.apps/reservation-backend-67c9b7db4b	0	0	0	80m
replicaset.apps/reservation-backend-6888c5df68	0	0	0	92m
replicaset.apps/reservation-backend-6b9886d548	1	1	1	12m
replicaset.apps/reservation-backend-8488f486bb	0	0	0	122m
replicaset.apps/reservation-backend-86489b84f9	0	0	0	16h
replicaset.apps/reservation-frontend-54867bbd67	0	0	0	14h
replicaset.apps/reservation-frontend-5989466596	0	0	0	15h
replicaset.apps/reservation-frontend-64d7fb67bf	0	0	0	45m
replicaset.apps/reservation-frontend-66959fb74d	0	0	0	15h
replicaset.apps/reservation-frontend-6f6b9dff7	1	1	1	32m
replicaset.apps/reservation-frontend-6fd66c7cff	0	0	0	152m
replicaset.apps/reservation-frontend-7898b77bdc	0	0	0	15h
replicaset.apps/reservation-frontend-7c6dc59d89	0	0	0	58m
replicaset.apps/reservation-frontend-8866c484b	0	0	0	42m
replicaset.apps/reservation-frontend-8dbb8bd65	0	0	0	14h
replicaset.apps/reservation-frontend-f666d85cc	0	0	0	14h

➤ l'application déployée



Étape 6 : Prometheus & Grafana – Supervision

➤ Prometheus

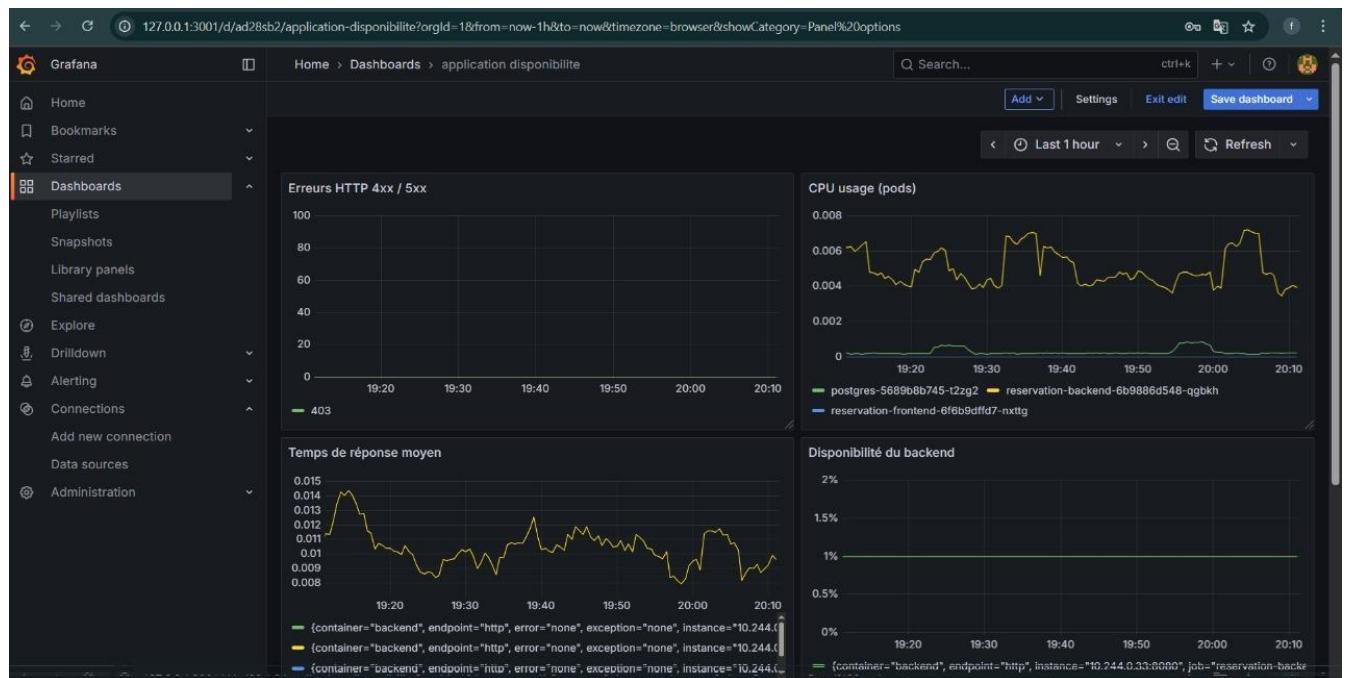
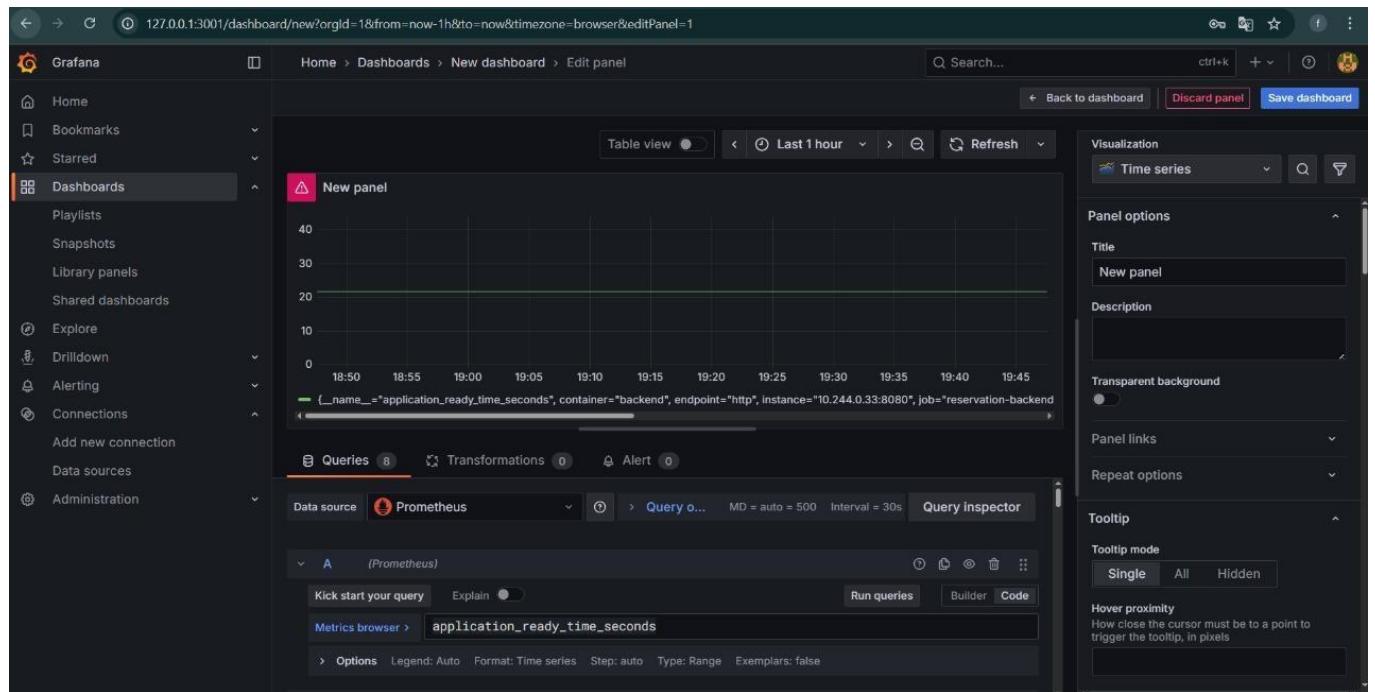
Endpoint	Labels	Last scrape	State
http://10.244.0.39:9093/metrics	container="alertmanager", endpoint="http-web", instance="10.244.0.39:9093", job="kube-prometheus-stack-alertmanager", namespace="monitoring", pod="alertmanager-kube-prometheus-stack-alertmanager-0", service="kube-prometheus-stack-alertmanager"	21.424s ago	6ms UP
http://10.244.0.39:8080/metrics	container="config-reloader", endpoint="reloader-web", instance="10.244.0.39:8080", job="kube-prometheus-stack-alertmanager", namespace="monitoring", pod="alertmanager-kube-prometheus-stack-alertmanager-0", service="kube-prometheus-stack-alertmanager"	14.282s ago	15ms UP
https://192.168.49.2:8443/metrics	endpoint="https", instance="192.168.49.2:8443", job="apiserver", namespace="default"	4.239s ago	UP

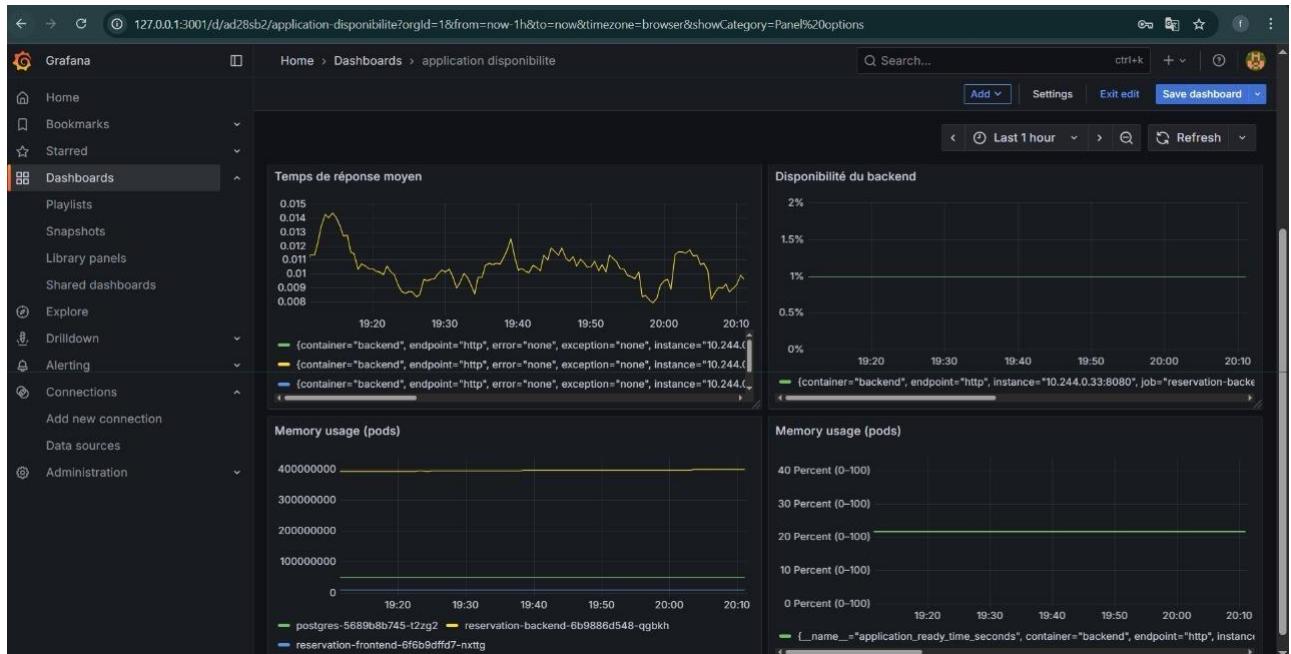
The screenshot shows the Prometheus web interface at the URL `127.0.0.1:9095/targets?search=reservation`. The top navigation bar includes links for 'Query', 'Alerts', and 'Status > Target health'. A search bar at the top right contains the text 'reservation'. Below the search bar, a message says 'Hiding pools with no matching targets' and 'Hiding 16 empty pools due to filters or no targets.' A link 'Show empty pools' is present. The main table lists a single target under the heading 'serviceMonitor/reservation/reservation-backend/0'. The table columns are 'Endpoint', 'Labels', 'Last scrape', and 'State'. The endpoint listed is `http://10.244.0.33:8080/actuator/prometheus`. The labels include `container="backend"`, `endpoint="http"`, `instance="10.244.0.33:8080"`, `job="reservation-backend"`, `namespace="reservation"`, `pod="reservation-backend-6b9886d548-qgbkh"`, and `service="reservation-backend"`. The 'Last scrape' column shows '17.516s ago' with a refresh icon. The 'State' column shows 'UP' with a green circle.

➤ Grafana

Ce dashboard a été créé dans **Grafana** à partir des métriques exposées par **Prometheus** via l'endpoint `/actuator/prometheus` du service `reservation-backend`. Il affiche en temps réel :

- la **consommation CPU et mémoire** des pods du namespace `reservation`,
- la **disponibilité** du service `backend`,
- le **temps de réponse moyen** des requêtes HTTP,
- et le **taux d'erreurs HTTP 4xx/5xx**.





❖ Quelle métrique avez-vous choisie comme KPI principale ? Pourquoi ?

J'ai choisi la **disponibilité du service (up)** comme KPI principal, car elle reflète directement l'état de santé du backend : si le service tombe, toutes les autres métriques deviennent secondaires. Ce KPI mesure la fiabilité perçue par l'utilisateur final.

❖ Alertes envisageables

Alerte	Seuil	Objectif
Service Down	up == 0	Déetecter une panne du backend
Haut latence	avg(rate(http_server_requests_seconds_sum[5m]) / rate(http_server_requests_seconds_count[5m])) > 2	Prévenir une dégradation du temps de réponse
Surcharge CPU	sum(rate(container_cpu_usage_seconds_total[5m])) by (pod) > 0.8	Surveiller une utilisation CPU excessive
Mémoire élevée	sum(container_memory_usage_bytes{namespace="reservation"}) by (pod) > 500000000	Identifier un risque d'OOM ou de fuite mémoire
Taux d'erreurs élevé (4xx/5xx)	`sum(rate(http_server_requests_seconds_count{status=~"4..5.."}[5m])) > 5`	