



REPUBLIQUE TUNISIENNE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Carthage
Institut National des Sciences Appliquées et de Technologie



Projet Fin d'Année

Renforcement de la sécurité et de la detection d'anomalies dans un cluster kubernetes par l'IA

Réalisé Par: Aloui Ghofrane – Jarboui Nada – Bel haj fraj Nour ElHouda

Encadré Par:
Prof. Noureddine Hamdi

Examiné Par:
Prof. Abderrazek JEMAI

Plan:

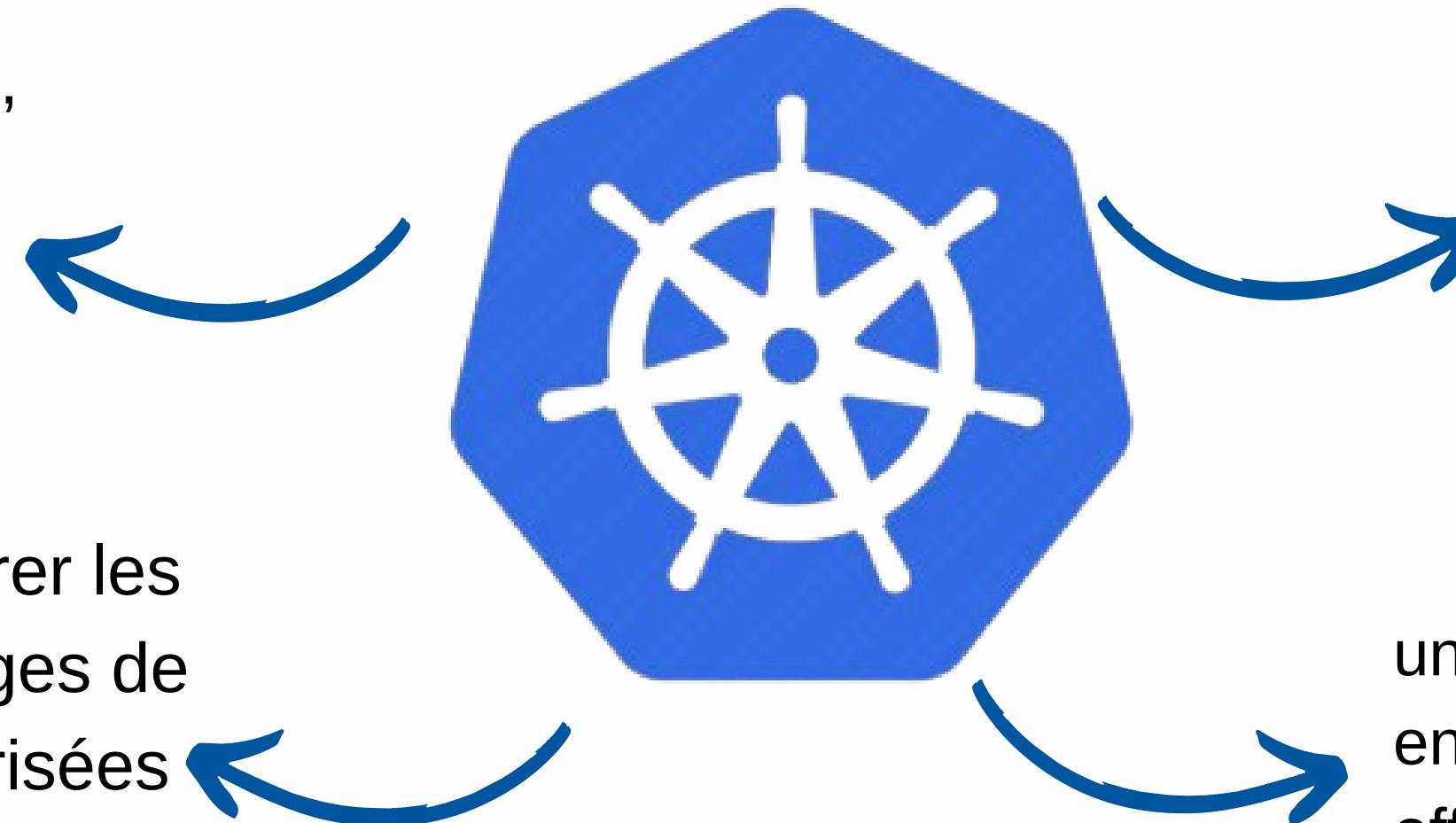
- ▶ Contexte et État de l'Art 01
- ▶ Architecture du système 02
- ▶ Collecte de données et scénarios de test 03
- ▶ Entrainement et développement du modèle AI 04
- ▶ Conclusion et Perspectives 05

01

Contexte et État de l'Art

KUBERNETES (K8S)

Plateforme open source puissante, portable et extensible



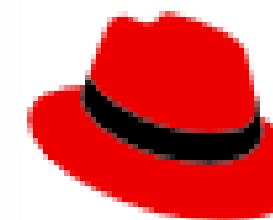
Conçue pour gérer les services et charges de travail conteneurisées

facilite l'automatisation, la configuration déclarative et la gestion des applications à grande échelle

un écosystème riche et en constante évolution, offrant une large gamme de services, d'outils et de support.

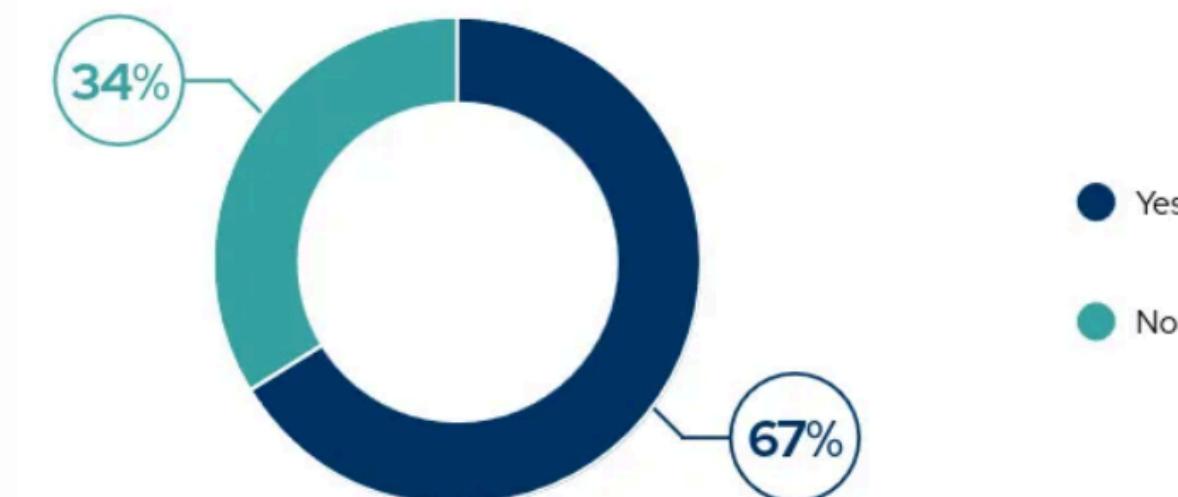
Kubernetes et la Transformation Numérique

- Kubernetes, leader des orchestrateurs cloud-native, automatise le déploiement et la gestion des applications conteneurisées.
- Utilisé par startups et infrastructures critiques, il est au cœur de la transformation numérique.
- Sa complexité et nature dynamique exposent les clusters à des failles de sécurité : 67 % des entreprises utilisant Kubernetes ont signalé des failles en 2024 (Red Hat).



Red Hat

Have you ever delayed or slowed down application deployment
into production due to container or Kubernetes security concerns?



Les approches Existantes: Systèmes de détection basés sur des politiques prédéfinies ou des seuils statiques



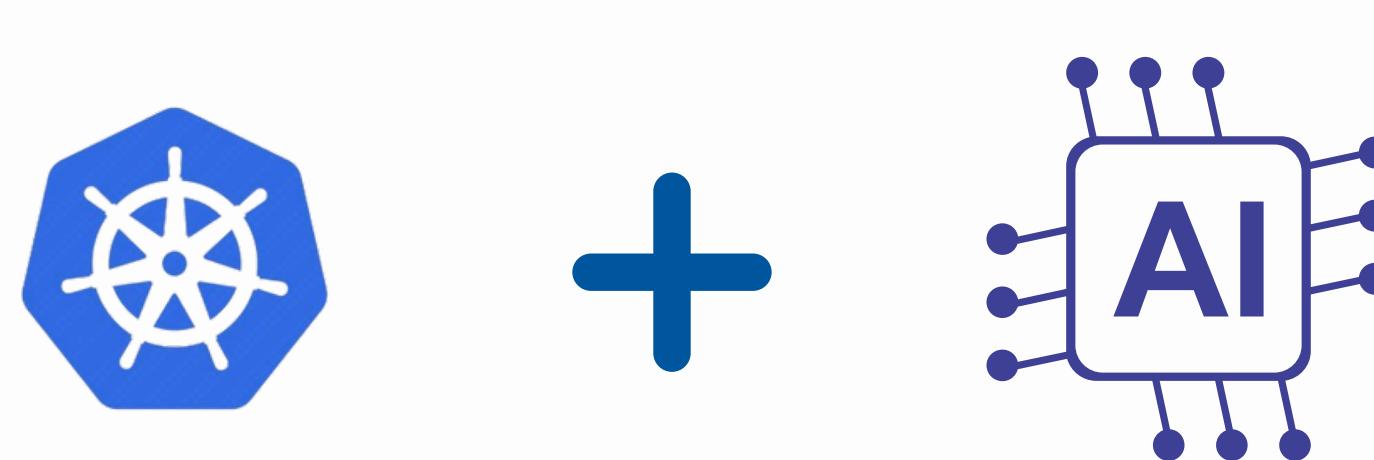
Prometheus



Ces approches traditionnelles, bien que robustes pour des scénarios **prévisibles**, échouent souvent à répondre aux défis des environnements Kubernetes **dynamiques**, où les conteneurs sont éphémères et les menaces évoluent rapidement.

Problématique et Objectif:

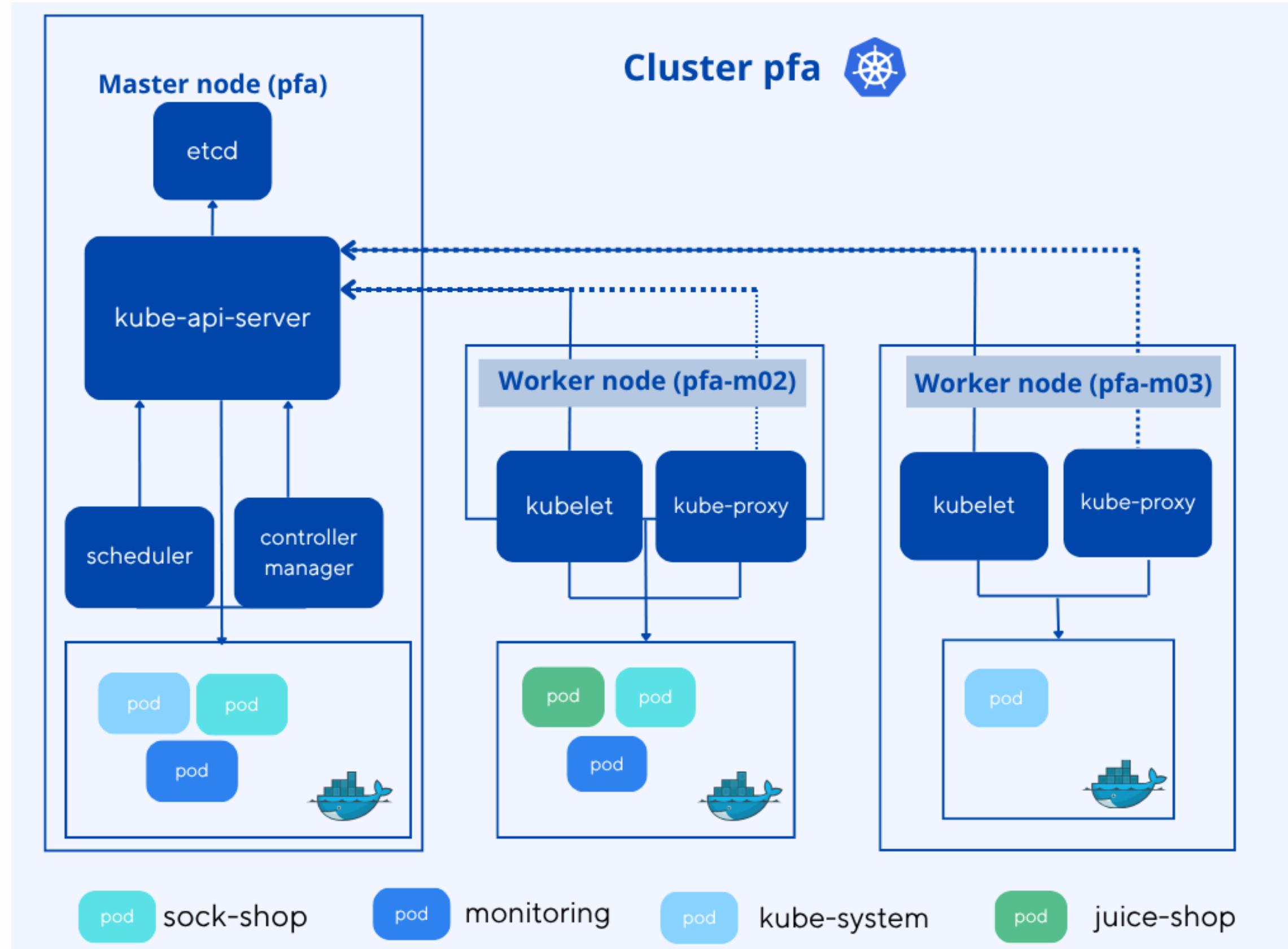
- **Problématique:** Les approches courantes échouent à détecter les menaces évolutives dans Kubernetes.
- **Objectif :** développer une solution proactive, basée sur l'IA, pour détecter les comportements anormaux en temps réel et renforcer la résilience des clusters Kubernetes



02

Architechture Système

Architecture du système

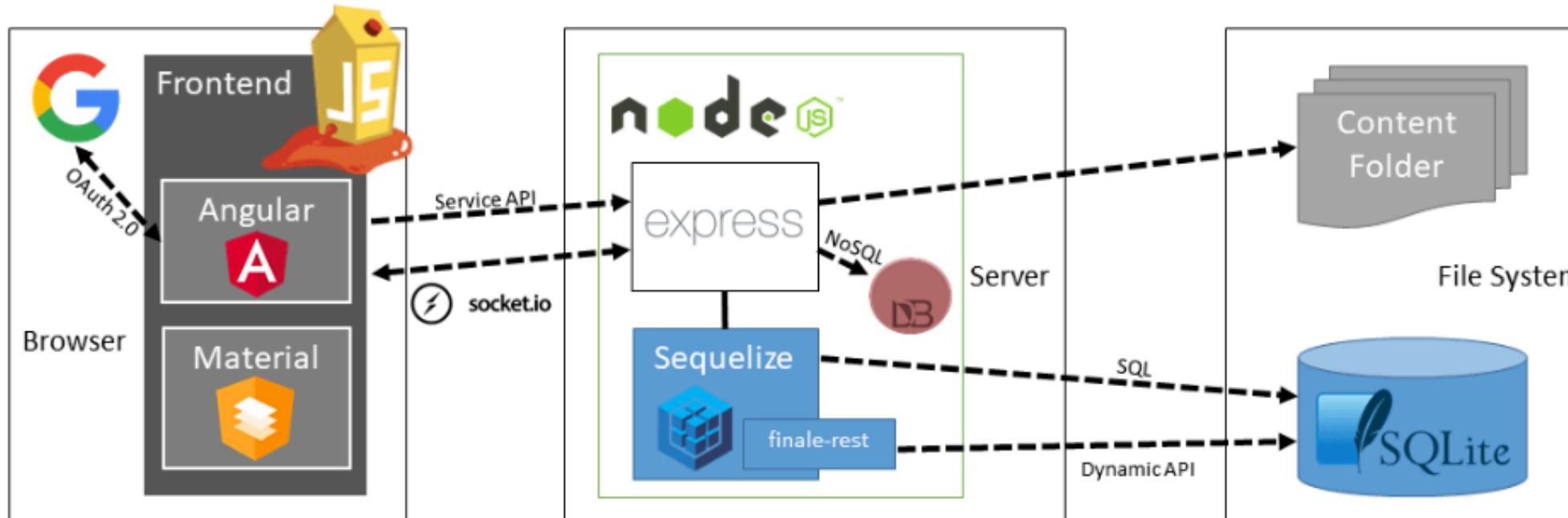


Déploiement des applications

◆ Juice Shop (Monolithique) :

namespace: juice-shop

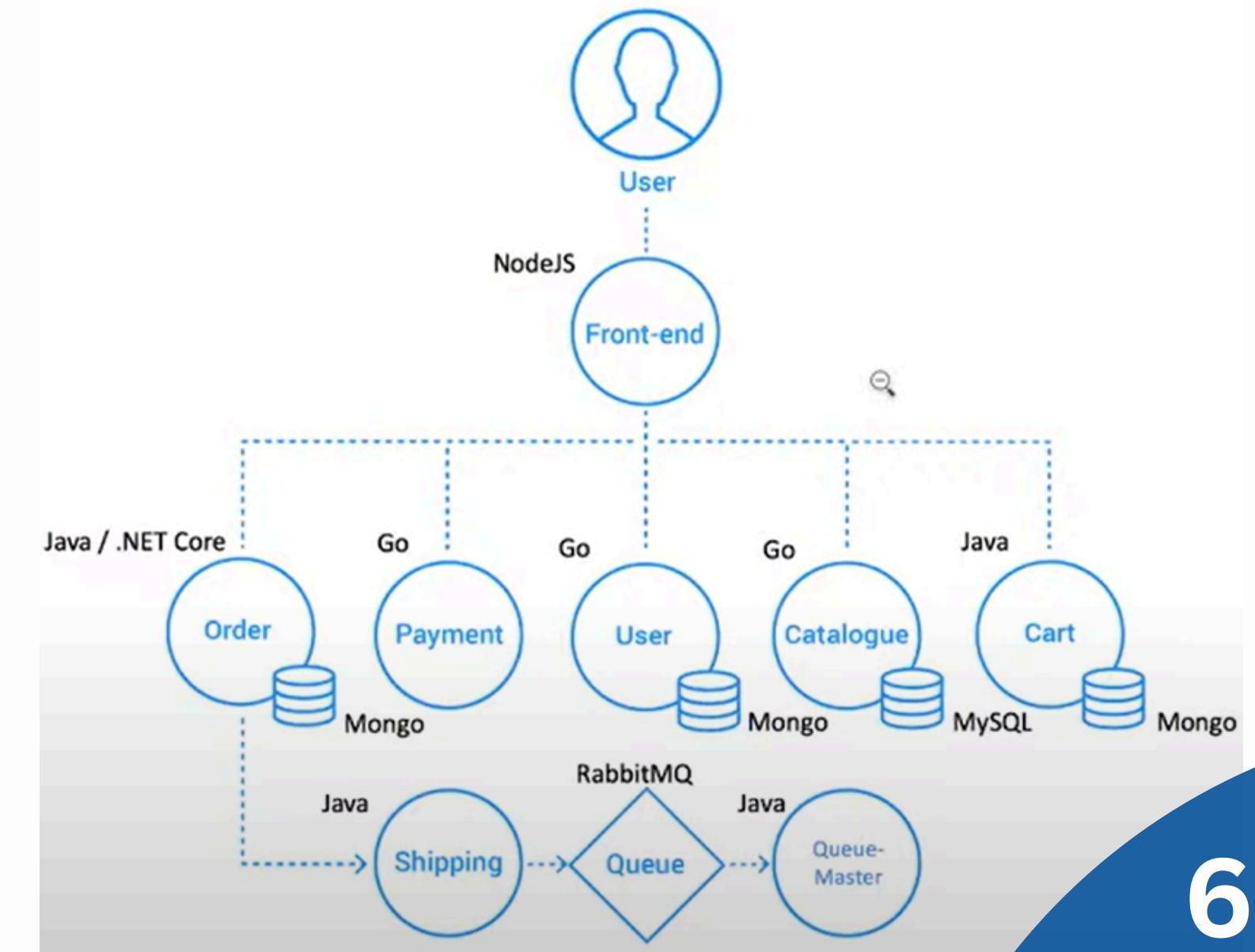
backend + frontend dans un seul conteneur



◆ Sock Shop (Microservices) :

namespace: sock-shop

plusieurs petits services indépendants



Intégration de Prometheus



Prometheus

namespace : monitoring

- Il interroge automatiquement les endpoints exposés par les services (kubelet, kube-apiserver, Juice Shop, Sock Shop) à l'aide de ServiceMonitor
- Toutes les métriques collectées sont stockées en base temporelle (time series) pour analyse

03

SIMULATION ET COLLECTE DES DONNÉES

Typologie des anomalies et métriques

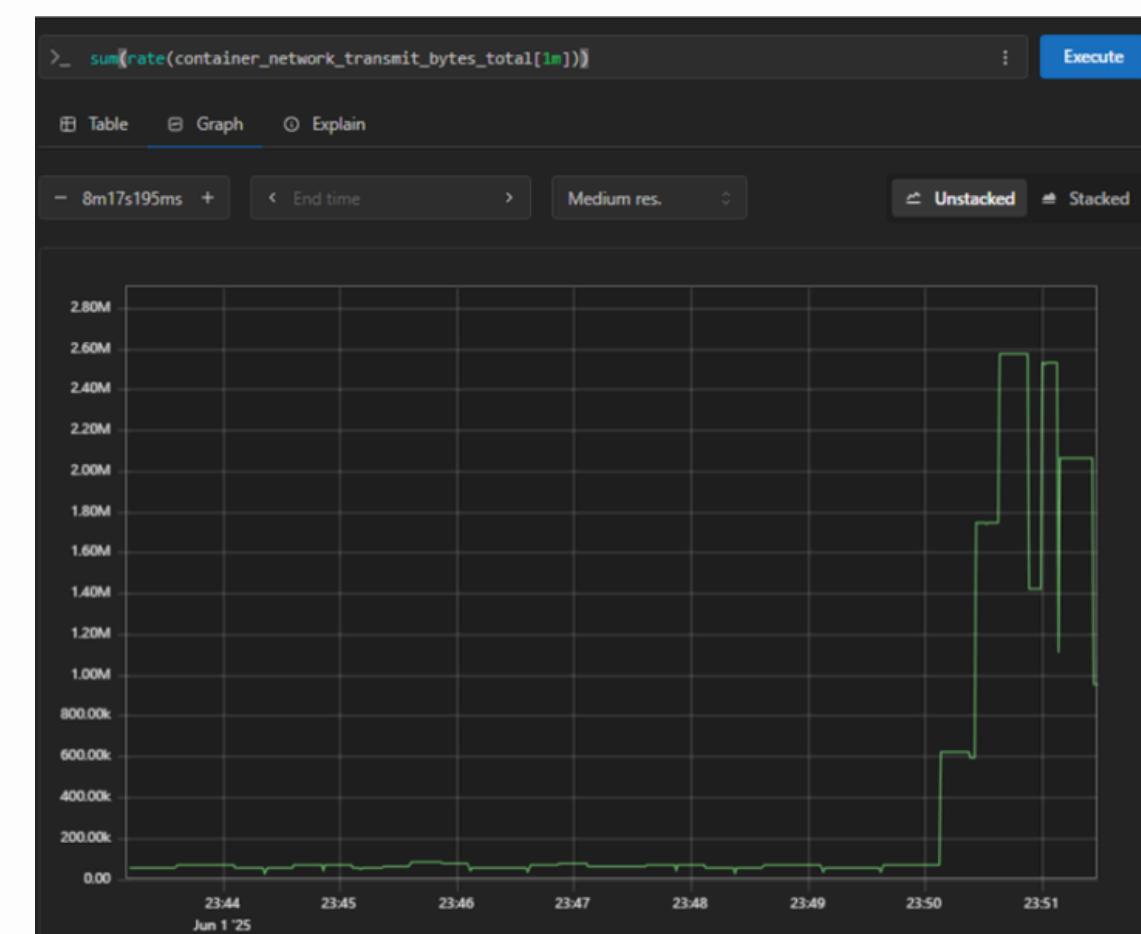
Catégorie d'anomalie	Comportement ciblé	Métriques sélectionnées
Anomalies de ressources	CPU, mémoire	<ul style="list-style-type: none">• Container_cpu_usage_seconds_total• Container_memory_usage_bytes
Anomalie réseau	Trafic entrant/sortant	<ul style="list-style-type: none">• Container_network_receive_bytes_total• Container_network_transmit_bytes_total
Anomalie d'état	Redémarrages des pods	<ul style="list-style-type: none">• kube_pod_container_status_restarts_total
Anomalie de sécurité	Trafic suspect	<ul style="list-style-type: none">• Container_network_transmit_bytes_total• Container_network_packets_total

Scénarios de test

- Minage de crypto : sollicite CPU et mémoire.
- Scan réseau et DoS : augmente le trafic réseau.
- CrashLoopBackOff : redémarrages en boucle.
- Exfiltration de données : transferts vers l'extérieur.

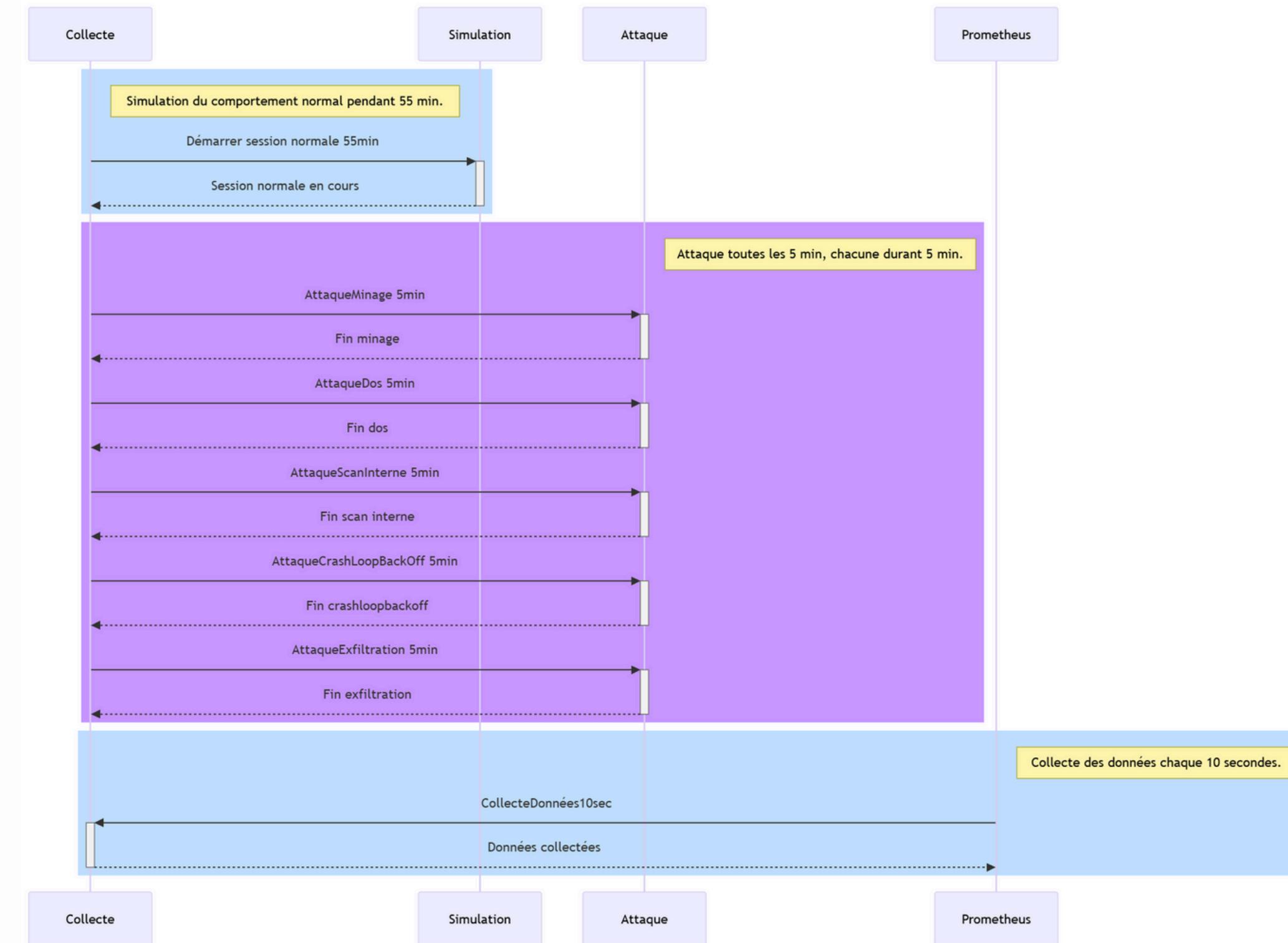


La consommation des ressources lors de l'attaque de minage de crypto



Évolution du trafic réseau lors d'une attaque DoS

Simulation et collecte des données



Conclusion

En simulant à la fois des comportements normaux et des attaques, nous avons constitué un jeu de données réaliste et complet. Cela nous permettra d'entraîner nos modèles de détection d'anomalies et ainsi renforcer la sécurité de notre cluster Kubernetes.

04

ENTRAINEMENT ET DEVELOPPEMENT DU MODELE AI

Format et Qualité des Données

Contenu du dataset :

- Colonnes : **timestamp, cpu, memory, net_tx, net_rx, restarts, label**
- 662 lignes de données propres, sans valeurs manquantes

Observation :

- cpu, memory, net_tx, net_rx → grande variabilité utile
- Forte corrélation entre CPU ↔ trafic réseau

----> ces observations guident le choix des algorithmes

	timestamp	restarts	cpu	memory	net_tx	net_rx	label
0	2025-05-31T13:29:15.909462	21	2.718602	1.475078e+09	56543.379375	75897.256662	normal
1	2025-05-31T13:29:25.983756	21	2.923115	1.072476e+09	73571.786432	82516.245167	normal
2	2025-05-31T13:31:22.742969	21	3.060250	1.192731e+09	77178.765515	66844.052103	normal
3	2025-05-31T13:31:32.807852	21	3.121569	1.291049e+09	92443.204075	72657.879052	normal
4	2025-05-31T13:31:42.868335	21	2.739631	1.248582e+09	129747.178167	77503.513799	normal

Prétraitement des Données

- **Conversion du timestamp**
- **Normalisation des valeurs numériques**
- **Encodage des labels** : normal = 0, anomalie = 1
- **Division du dataset** : 70 % entraînement et 30 % test

	timestamp	restarts	cpu	memory	net_tx	net_rx	label
0	2025-05-31 13:29:15.909462	21	-0.795973	-0.108141	-4.595319	-4.435137	0
1	2025-05-31 13:29:25.983756	21	-0.689387	-0.765725	-4.470362	-4.346536	0
2	2025-05-31 13:31:22.742969	21	-0.617916	-0.569308	-4.443893	-4.556323	0
3	2025-05-31 13:31:32.807852	21	-0.585958	-0.408722	-4.331880	-4.478499	0
4	2025-05-31 13:31:42.868335	21	-0.785013	-0.478085	-4.058136	-4.413636	0

Algorithmes Testés

6 modèles sélectionnés :

Type	Modèle	Raison du choix
Non supervisé	Isolation Forest	Rapide, efficace, pas besoin d'étiquettes
Non supervisé	One-Class SVM	Bien adapté aux données normales
Non supervisé	LOF	Déetecte les anomalies locales
Non supervisé	KNN	Simple, intuitif
Non supervisé	PCA	Détection via réduction de dimensions
Supervisé	Random Forest	Très précis avec données simulées

Métriques de performances:

$$Exactitude = \frac{VP + VN}{VP + VN + FP + FN}$$

$$Précision = \frac{VP}{VP + FP}$$

$$F1-score = \frac{2 \cdot Précision \cdot Recall}{Précision + Recall}$$

$$Recall = \frac{VP}{VP + FN}$$

AUC = Area Under the Curve: mesure la capacité d'un modèle à distinguer les classes: compromis entre rappel et spécificité.

Performances Atteintes

Modèle	Accuracy	F1-score (Anomaly)	Recall (Anomaly)	Precision (Anomaly)	AUC
Isolation Forest	0.66	0.32	0.20	0.89	0.59
One-Class SVM	0.65	0.29	0.17	0.88	0.58
Local Outlier Factor	0.61	0.15	0.09	0.70	0.53
KNN	0.65	0.29	0.17	0.82	0.57
PCA	0.66	0.31	0.19	0.88	0.58
Random Forest (Sup.)	0.80	0.73	0.65	0.82	0.78

05

CONCLUSION ET PERSPECTIVES

Conclusion - Succès et Apports du Projet

- Mise en place d'un cluster Kubernetes avec une stack de monitoring (Prometheus, Grafana).
- Collecte de métriques et simulation de scénarios normaux/attaques pour entraîner un modèle ML.
- Démonstration de l'efficacité de l'IA pour une détection proactive des anomalies.
- Contribution : Solution adaptative surmontant les limites des approches traditionnelles.

Défis - Contraintes du Projet

- Ressources limitées : Tests sur seulement trois machines, réduisant la scalabilité.
- Manque de références pour configurer un cluster Kubernetes local.
- Difficultés d'exposition des métriques via Prometheus (configuration, intégration).
- Sélection complexe des scénarios pertinents pour l'entraînement du modèle.

Perspectives - Améliorations Futures

- Développer un modèle IA plus performant (réseaux de neurones, apprentissage par renforcement).
- Déployer le modèle dans le cluster pour une détection en temps réel.
- Implémenter un système de notification automatisé (Slack, email, Grafana).
- Objectif : Renforcer la réactivité et la résilience des clusters Kubernetes.

**MERCI POUR
VOTRE
ATTENTION !**