

# DOCUMENTATION TECHNIQUE

## SLA - MPL\_10

**VERSION**  
1.0

**AUTEURS**  
MPL\_10

Date de dernière mise à jour : mardi 6 juin 2023

# Table des matières

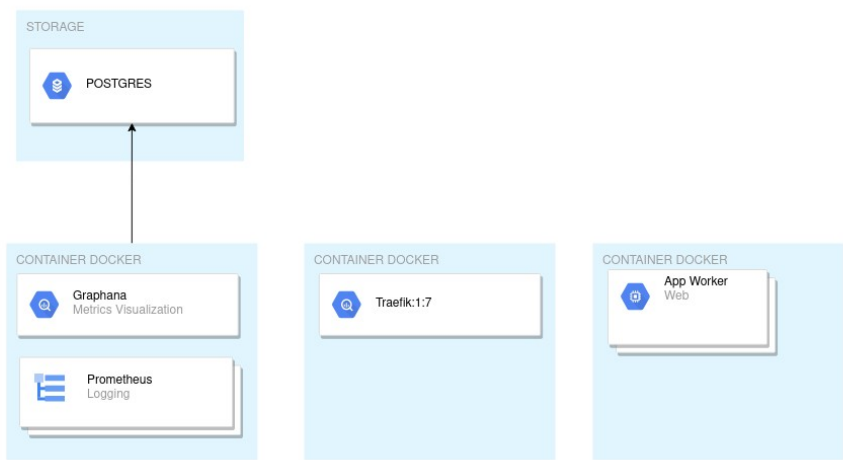
---

- 1 - Historique du document.....3
- 2 - Schema de fonctionnement.....4
- 3 - Monitoring Tools and Dashboards.....5
  - 3.1 - Grafana.....5
  - 3.2 - Prometheus.....8
  - 3.3 - Zabbix.....9
  - 3.4 - Traefik.....11
- 4 - Stress Tools.....13
  - 4.1 - Stress-ng.....13

# 1 - Historique du document

Date	Auteur	Version	Sujet de la modification
30/12/1899		0.0	Création
30/06/2023	MPL_10	0.5	Première modification
30/06/2023	MPL_10	0.6	Ajout de screenshots
05/06/2023	MPL_10	0.8	Ajout Zabbix
06/06/2023	MPL_10	1.0	Dernière modif et Rendu

## 2 - Schema de fonctionnement



## 3 - Monitoring Tools and Dashboards

### 3.1 - Grafana

Grafana est une plateforme open-source de visualisation de données et de monitoring. Elle permet de collecter, analyser et afficher des métriques provenant de diverses sources, offrant une vue graphique et interactive de la santé, des performances et de l'état des composants d'infrastructure.

#### Principales fonctionnalités de la surveillance avec Grafana :

1. **Connexion à différentes sources de données** : Grafana prend en charge une variété de sources de données, telles que des bases de données, des services de monitoring tels que Prometheus, des systèmes de logs et des APIs. Cela permet de collecter des métriques provenant de différents systèmes et de les visualiser sur un même tableau de bord.
2. **Création de tableaux de bord personnalisés** : Grafana offre une interface conviviale permettant de créer des tableaux de bord personnalisés. Les utilisateurs peuvent ajouter des graphiques, des jauges, des tableaux et d'autres visualisations pour représenter les métriques de manière claire et intuitive. Les tableaux de bord peuvent être adaptés aux besoins spécifiques de l'infrastructure surveillée.
3. **Exploration et navigation des données** : Grafana permet aux utilisateurs d'explorer et de naviguer facilement dans les données surveillées. Ils peuvent zoomer, filtrer et effectuer des recherches pour obtenir des informations détaillées sur les métriques. Grafana facilite l'identification des tendances, des modèles et des anomalies pour une analyse approfondie des performances.
4. **Alertes et notifications** : Grafana permet de définir des seuils et des conditions d'alerte sur les métriques surveillées. Lorsqu'une condition est atteinte, des alertes peuvent être déclenchées pour informer les utilisateurs via des canaux tels que l'e-mail, les webhooks ou les intégrations avec des systèmes de messagerie. Les alertes aident à détecter rapidement les problèmes et à prendre des mesures appropriées.
5. **Partage des tableaux de bord** : Grafana offre la possibilité de partager les tableaux de bord avec d'autres utilisateurs ou de les publier publiquement. Cela permet une collaboration efficace et une transparence au sein de l'équipe. Les tableaux de bord partagés peuvent être utilisés pour la résolution collaborative des problèmes et la prise de décisions basées sur les données.

Grafana peut être utilisé pour visualiser les métriques collectées par les outils de surveillance tels que Prometheus. Il peut être connecté à différentes sources de données, notamment les serveurs d'application, les bases de données PostgreSQL et Redis, et les serveurs Docker, pour obtenir des métriques détaillées sur leurs performances et leur disponibilité.

Grafana facilite également la création d'alertes sur des seuils spécifiques. Par exemple, une alerte peut être déclenchée lorsque l'utilisation du CPU dépasse un certain seuil ou lorsque la latence de l'application dépasse une valeur définie. Ces alertes aident à détecter les problèmes critiques et à prendre des mesures proactives pour minimiser les interruptions de service.

**Lien de connexion :**

<http://default-group11-docker.francecentral.cloudapp.azure.com:3000/>  
user : admin  
mdp : uUG@PWxx3eYgZL



## 3.2 - Prometheus

Prometheus est un système open-source de surveillance et d'alerte basé sur une architecture de type "pull". Il est conçu pour collecter, stocker et analyser des métriques provenant de diverses sources, offrant ainsi une visibilité en temps réel sur la santé, les performances et l'état des composants d'infrastructure.

### Principales fonctionnalités de la surveillance avec Prometheus :

1. **Collecte de métriques** : Prometheus collecte les métriques à l'aide d'un modèle de collecte appelé "scraping". Il peut interroger les endpoints HTTP exposant des métriques, interroger des services de découverte de services tels que Consul ou Kubernetes, ou recevoir des métriques poussées par des agents ou des applications spécifiques. Cette approche permet une collecte flexible et adaptée à différentes architectures.
2. **Stockage et rétention des données** : Prometheus stocke les métriques collectées dans une base de données de séries temporelles. Il offre une rétention configurable des données, permettant de conserver des données historiques sur une période spécifiée. Cela permet d'effectuer des analyses rétrospectives, d'identifier les tendances et de comprendre l'évolution des performances sur le long terme.
3. **Requêtes et alertes** : Prometheus propose un langage de requête puissant appelé PromQL, qui permet d'interroger et d'agréger les métriques stockées. Il permet de créer des expressions complexes pour filtrer, agréger et calculer des statistiques sur les données. Prometheus intègre également un système d'alerte flexible, permettant de définir des règles d'alerte basées sur des seuils ou des conditions spécifiques.
4. **Intégrations et écosystème** : Prometheus offre une intégration étroite avec d'autres outils et services populaires tels que Grafana, Alertmanager, et différents fournisseurs cloud. Cela permet d'étendre ses fonctionnalités et de créer des workflows complets pour la surveillance, l'alerte et la visualisation des métriques. Il bénéficie également d'une communauté active et d'une large gamme de bibliothèques et d'exporteurs prêts à l'emploi.
5. **Scalabilité et haute disponibilité** : Prometheus est conçu pour être hautement scalable et tolérant aux pannes. Il prend en charge le déploiement de multiples instances pour la collecte de métriques, permettant ainsi de gérer des environnements à grande échelle. De plus, il supporte la fédération de plusieurs instances Prometheus, permettant d'agréger et de centraliser les données de différents clusters ou zones géographiques.

#### Lien de connexion :

<http://default-group11-docker.francecentral.cloudapp.azure.com:9000/>



```
1 version: '3'
2
3 volumes:
4   prometheus-data:
5     driver: local
6   grafana-data:
7     driver: local
8
9 services:
10  prometheus:
11    image: prom/prometheus:latest
12    container_name: prometheus
13    ports:
14      - "9000:9090"
15    volumes:
16      - ./prometheus:/etc/prometheus
17      - prometheus-data:/prometheus
18    restart: unless-stopped
19    command: --web.enable-lifecycle --config.file=/etc/prometheus/prometheus.yml
20
21  zabbix_db:
22    image: postgres
23    restart: always
24    environment:
25      POSTGRES_PASSWORD
26      POSTGRES_USER
27      POSTGRES_DB
28
29  grafana:
30    image: grafana/grafana:latest
31    container_name: grafana
32    ports:
33      - "9091:9091"
34    volumes:
35      - grafana-data:/var/lib/grafana
36    restart: unless-stopped
```

## 3.3 - Zabbix

---

Zabbix est une solution de surveillance open-source qui permet aux organisations de surveiller la santé, les performances et la disponibilité de leurs composants d'infrastructure informatique. Elle offre une plateforme centralisée pour collecter, analyser et visualiser les données provenant de différentes sources, permettant aux administrateurs d'obtenir des informations précieuses sur le comportement du système et de prendre des décisions éclairées.

### Principales fonctionnalités de la surveillance avec Zabbix :

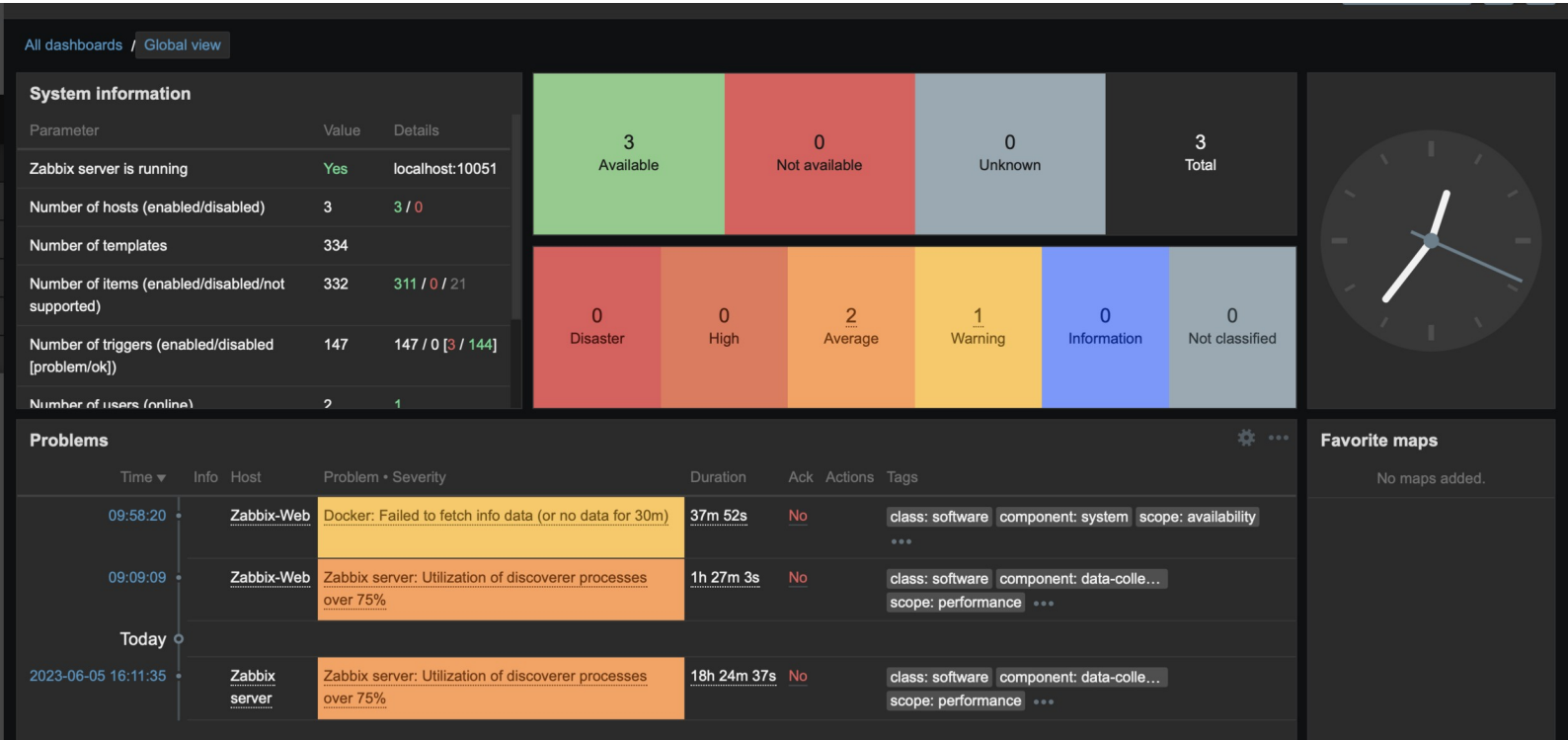
1. **Surveillance des métriques** : Zabbix permet de surveiller une large gamme de métriques, notamment la santé des serveurs, des équipements réseau, des bases de données, des applications et des ressources cloud. Il prend en charge la surveillance active et passive, permettant la détection proactive des problèmes et des réponses réactives aux événements.
2. **Alertes et notifications** : Zabbix offre des capacités d'alerte flexibles, permettant aux utilisateurs de définir des déclencheurs basés sur des seuils ou des conditions spécifiques. Lorsqu'un incident est détecté, il peut envoyer des notifications via différents canaux tels que l'e-mail, les SMS, Slack ou des scripts personnalisés, assurant ainsi une notification rapide du personnel concerné.
3. **Collecte et stockage des données** : Zabbix collecte les données à l'aide d'agents installés sur les périphériques surveillés ou en utilisant des protocoles tels que SNMP, ICMP, JMX et IPMI. Les données collectées sont stockées dans une base de données backend, permettant une analyse historique, une surveillance des tendances et une planification des capacités.
4. **Visualisation et reporting** : Zabbix offre des tableaux de bord et des rapports personnalisables pour visualiser les données surveillées. Les utilisateurs peuvent créer des graphiques, des diagrammes et des cartes pour visualiser les performances et la disponibilité des composants de leur infrastructure. Il propose également des modèles prédéfinis et des widgets pour un déploiement rapide et une personnalisation aisée.
5. **Scalabilité et surveillance distribuée** : Zabbix est conçu pour gérer des déploiements à grande échelle, ce qui le rend adapté aux organisations disposant d'infrastructures complexes et distribuées. Il prend en charge la surveillance distribuée, permettant la configuration de plusieurs serveurs et proxies Zabbix pour répartir la charge de surveillance et assurer la redondance.
6. **Automatisation et remédiation** : Avec Zabbix, il est possible d'automatiser des actions basées sur des déclencheurs prédéfinis. Cela inclut l'exécution de scripts,

le redémarrage de services ou le déclenchement de workflows automatisés pour remédier aux incidents détectés.

**Note aux développeurs :**  
Ceci est une note pour les développeurs !  
*Exemples :*

- Prévoir en flexibilité : la liste pourra changer au cours du temps et être 10 fois plus longue.
- Le système AX est incompatible avec le produit AY : Utilisez la méthode BY.

```
36 //Conf Zabbix :
37
38 LogFileSize=0
39
40 PidFile=/run/zabbix/zabbix_server.pid
41
42 SocketDir=/run/zabbix
43
44 DBHost=127.0.0.1
45
46 #sudo zcat /usr/share/doc/zabbix-sql-scripts/mysql/create.sql.gz | mysql -
47 uzabbix -p zabbix
48
49 #/etc/zabbix/zabbix_server.conf conf file
50
51 #DBPassword=password
```



## 3.4 - Traefik

---

Traefik est un serveur proxy inversé et un routeur de trafic open-source conçu pour les environnements de conteneurisation. Il agit en tant que point d'entrée pour les requêtes HTTP et gère la répartition du trafic vers les différentes applications ou services en fonction de règles de routage définies.

### **Principales caractéristiques de la surveillance avec Traefik :**

1. **Gestion dynamique du trafic :** Traefik permet une gestion dynamique du trafic en se connectant directement à l'orchestrateur de conteneurs (comme Docker, Kubernetes, etc.) pour découvrir les services disponibles et leurs instances. Il met à jour automatiquement la configuration de routage pour prendre en compte les changements dans l'environnement des conteneurs.
2. **Tableaux de bord et métriques intégrées :** Traefik offre des fonctionnalités intégrées pour la surveillance et la collecte de métriques. Il expose des endpoints de métriques qui peuvent être utilisés par des outils de surveillance tels que Prometheus pour collecter et stocker les métriques de performance de Traefik. Il propose également des tableaux de bord prêts à l'emploi pour visualiser les statistiques et les informations sur le trafic en temps réel.
3. **Alertes et notifications :** Traefik peut être configuré pour envoyer des alertes et des notifications en cas de comportement anormal ou de problèmes de disponibilité. Cela permet de détecter rapidement les incidents, tels que les erreurs de routage, les défaillances des services ou les pics de trafic, et de prendre des mesures appropriées pour résoudre les problèmes.
4. **Gestion des certificats SSL :** Traefik prend en charge la gestion automatique des certificats SSL grâce à l'intégration avec des fournisseurs tels que Let's Encrypt. Il peut générer, renouveler et gérer les certificats SSL pour les domaines configurés, ce qui garantit la sécurité des connexions et la disponibilité des services avec une configuration simplifiée.
5. **Logs et journalisation :** Traefik fournit des fonctionnalités de journalisation qui enregistrent les événements et les actions effectuées par le serveur proxy. Les logs peuvent être consultés pour suivre l'activité du trafic, diagnostiquer les problèmes et analyser les performances de l'infrastructure.

Traefik est utilisé comme serveur proxy pour la gestion du trafic des applications hébergées dans des conteneurs. En surveillant Traefik, les administrateurs peuvent collecter des métriques sur les performances du serveur proxy, le trafic entrant, les erreurs de routage, les temps de réponse, etc.

De plus, les fonctionnalités de gestion des certificats SSL de Traefik garantissent la disponibilité sécurisée des connexions, tandis que les capacités de journalisation facilitent la surveillance et l'analyse des événements et des actions effectués par Traefik.

## 4 - Stress Tools

### 4.1 - Stress-ng

Stress-ng est un outil open-source utilisé pour effectuer des tests de stress et de charge sur les systèmes informatiques. Il permet de simuler des scénarios de charge élevée afin d'évaluer la robustesse, les performances et la stabilité de l'infrastructure.

#### Principales caractéristiques du test de stress avec stress-ng :

1. **Diverses méthodes de stress** : stress-ng propose une gamme de méthodes de stress pour simuler différents types de charges sur le système. Cela inclut des tests de charge CPU, de mémoire, de disque, de réseau et d'autres composants critiques. Chaque méthode de stress est conçue pour exercer intensément les ressources spécifiques du système afin d'évaluer leur comportement en situation de charge élevée.
2. **Personnalisation des paramètres de stress** : stress-ng offre la possibilité de personnaliser les paramètres des tests de stress en fonction des besoins spécifiques. Les utilisateurs peuvent ajuster la charge exercée sur chaque composant, la durée des tests, les intervalles de rapport et d'autres paramètres pour répondre aux exigences du scénario de test.
3. **Surveillance des performances** : Pendant les tests de stress, stress-ng fournit des informations en temps réel sur les performances et les métriques du système. Cela permet de suivre les variations de la charge, les temps de réponse, l'utilisation des ressources, les erreurs et d'autres indicateurs clés. La surveillance des performances aide à identifier les points faibles, les goulots d'étranglement et les problèmes potentiels lors de situations de stress.
4. **Analyse des résultats** : Une fois les tests de stress terminés, stress-ng génère des rapports détaillés sur les résultats obtenus. Ces rapports fournissent des statistiques sur les performances du système, les temps de réponse, les erreurs rencontrées et d'autres métriques pertinentes. L'analyse des résultats permet d'évaluer la capacité de l'infrastructure à résister à des charges élevées et à identifier les améliorations potentielles.
5. **Répétabilité des tests** : stress-ng permet d'exécuter des tests de stress de manière répétable. Cela signifie que les tests peuvent être exécutés plusieurs fois avec les mêmes paramètres pour valider les résultats, mesurer les écarts de performances et identifier les variations. La répétabilité des tests est essentielle pour obtenir des résultats cohérents et fiables.

Une fois les tests de stress terminés, les rapports générés par stress-ng fournissent une analyse détaillée des résultats obtenus, permettant d'évaluer la capacité de l'infrastructure à résister à des charges élevées et d'identifier les domaines d'amélioration potentiels. Ces informations peuvent servir de base pour prendre des décisions d'optimisation et de dimensionnement de l'infrastructure afin de garantir une haute disponibilité et des performances optimales.