Département du Système d'Information

CONTEXTE • **ELK via Docker**

SUJET • **Mise en service**

référence	*001 - dossier d'exploitation.docx*
version	1.1
statut	
créé le	*12/11/2020 11:03:00*
par	Lukas Taboga
mis à jour le	*03/06/2024 17:21:00*
par	Lukas Taboga
validé le	*03/06/2024 17:21:00*
par	En attende validation
diffusé le	
à	

Péremption, archivage et restriction de diffusion

Nature de la restriction : confidentiel, diffusion restreinte, diffusion interne, restriction annulée

Table des mises à jour du document

version	date	objet de la mise à jour
01	09/11/2020	Version initiale

Table des matières

1. Document d'exploitation (ELK via Docker)

1.1. Supervision

1.1.1. Supervision système

Statut des Conteneurs : Vérifiez que tous les conteneurs Docker (Elasticsearch, Logstash, Kibana) sont en cours d'exécution.shCopier le code

docker ps

Relance Automatique : Configurez les politiques de redémarrage pour assurer que les conteneurs redémarrent automatiquement en cas de panne.shCopier le code

docker update --restart unless-stopped <container_id>

Utilisation des Ressources (CPU, Mémoire)

Surveillance en Temps Réel : Utilisez docker stats pour surveiller la consommation de ressources des conteneurs.shCopier le code

docker stats

Espace Disque

Espace Utilisé : Vérifiez régulièrement l'espace disque pour éviter les saturations, surtout avec Elasticsearch qui peut consommer beaucoup d'espace.shCopier le code

df -h

docker system df

Logs

Accès aux Logs : Consultez les logs des conteneurs pour détecter des erreurs ou avertissements.shCopier le code

docker logs <container_id>

Centralisation des Logs : Utilisez la stack ELK elle-même pour centraliser et analyser les logs des conteneurs et des applications.

Performances Réseau

Ports et Latence : Assurez-vous que les ports nécessaires sont accessibles et surveillez la latence réseau.shCopier le code

netstat -tuln

Surveillance Réseau : Utilisez des outils comme ss ou des plugins spécifiques pour ELK pour une surveillance détaillée.

Configuration de l'Hôte

Versions Requises

Docker Engine version 18.06.0 ou plus récent

Docker Compose version 1.28.0 ou plus récent (y compris Compose V2)

Minimum de 1.5 Go de RAM

Permissions

Sur Linux, assurez-vous que votre utilisateur a les permissions nécessaires pour interagir avec le daemon Docker.shCopier le code

sudo usermod -aG docker $USER

Ports Exposés par Défaut:

• 5044 : Entrée Logstash Beats

• 50000 : Entrée Logstash TCP

• 9600 : API de monitoring Logstash

• 9200 : HTTP Elasticsearch

• 9300 : Transport TCP Elasticsearch

• 5601 : Kibana

Références

Docker-ELK GitHub Repository

Monitoring Docker Containers

Linux Process Monitoring

Disk Usage Monitoring

Network Performance Monitoring

1.1.2. Supervision applicative

Méthode 1 : Utiliser l'API Kibana

1. Exécuter le conteneur Kibana:

Bash
```
docker run -d -p 5601:5601 --name kibana \
 --network elk_net \
 elastic/kibana:7.10.2
```

2. Vérifier l'état du conteneur Kibana:

Bash
```
docker ps -a
```

3. Accéder à l'interface Web de Kibana:

Ouvrez votre navigateur Web et accédez à l'URL suivante:

http://<adresse_IP_ou_nom_de_domaine>:5601
Remplacez <adresse_IP_ou_nom_de_domaine> par l'adresse IP ou le nom de domaine de votre serveur Docker.

4. Procédez comme décrit dans la section "Méthodes" de la réponse précédente.

Méthode 2 : Utiliser l'outil curl

1. Exécuter le conteneur Kibana:

Bash
```
docker run -d -p 5601:5601 --name kibana \
 --network elk_net \
 elastic/kibana:7.10.2
```

2. Vérifier l'état du conteneur Kibana:

Bash
docker ps -a

3. Exécuter la commande curl:

Bash
curl -v http://<adresse_IP_ou_nom_de_domaine>:5601/

Remplacez <adresse_IP_ou_nom_de_domaine> par l'adresse IP ou le nom de domaine de votre serveur Docker.

Méthode 3 : Utiliser l'outil wget

1. Exécuter le conteneur Kibana:

Bash
docker run -d -p 5601:5601 --name kibana \
 --network elk_net \
 elastic/kibana:7.10.2

2. Vérifier l'état du conteneur Kibana:

Bash
docker ps -a

3. Exécuter la commande wget:

Bash

wget -O - http://<adresse_IP_ou_nom_de_domaine>:5601/

Remplacez <adresse_IP_ou_nom_de_domaine> par l'adresse IP ou le nom de domaine de votre serveur Docker.

Remarques:

- Assurez-vous de remplacer <adresse_IP_ou_nom_de_domaine> par l'adresse IP ou le nom de domaine de votre serveur Docker dans toutes les commandes.
- Vous pouvez utiliser les commandes docker logs et docker exec pour inspecter les logs des conteneurs Kibana et Elasticsearch pour obtenir plus d'informations sur l'état de votre stack ELK.

1.2. Sauvegardes

1.2.1. Stratégie appliquée

1. Configuration des Sauvegardes

Sauvegarde Totale Hebdomadaire

Créer un Script de Sauvegarde Totale :

Créez un fichier backup_full.sh pour sauvegarder toutes les données des volumes Docker utilisés par Elasticsearch :

bashCopier le code

#!/bin/bash

Arrêter les conteneurs ELK

docker-compose down

Créer une sauvegarde complète

tar -czvf /path/to/backup/full_backup_$(date +%F).tar.gz /path/to/elasticsearch/data

Redémarrer les conteneurs ELK

docker-compose up -d

Assurez-vous de remplacer /path/to/backup/ par le chemin où vous souhaitez stocker vos sauvegardes et /path/to/elasticsearch/data par le chemin du volume de données d'Elasticsearch.

Planifier la Sauvegarde Hebdomadaire :

Utilisez cron pour planifier l'exécution du script chaque semaine. Ouvrez le fichier crontab pour l'utilisateur qui exécute Docker :

bashCopier le code

crontab -e

Ajoutez la ligne suivante pour exécuter le script chaque dimanche à 2 heures du matin :

bashCopier le code

0 2 * * 0 /path/to/backup_full.sh

Sauvegarde Différentielle Quotidienne

Créer un Script de Sauvegarde Différentielle :

Créez un fichier backup_diff.sh pour sauvegarder uniquement les changements depuis la dernière sauvegarde complète :

bashCopier le code

#!/bin/bash

Arrêter les conteneurs ELK

docker-compose down

Créer une sauvegarde différentielle

tar -czvf /path/to/backup/diff_backup_$(date +%F).tar.gz --newer-mtime="$(date -d '1 week ago' +%F)" /path/to/elasticsearch/data

Redémarrer les conteneurs ELK

docker-compose up -d

Remplacez /path/to/backup/ et /path/to/elasticsearch/data par les chemins appropriés.

Planifier la Sauvegarde Quotidienne :

Ajoutez la tâche cron pour exécuter le script chaque jour à 2 heures du matin, sauf le dimanche :

bashCopier le code

crontab -e

Ajoutez la ligne suivante :

bashCopier le code

0 2 * * 1-6 /path/to/backup_diff.sh

1.3. Restauration

1.3.1. Restauration du système

Prérequis:

Vous devez avoir une sauvegarde de votre stack ELK. Vous pouvez créer une sauvegarde en utilisant des outils tels que docker commit, docker export ou des solutions de sauvegarde tierces. Vous devez connaître l'emplacement de votre sauvegarde. Méthodes de restauration:

Restauration à partir d'une image Docker:

Chargez l'image de sauvegarde:

docker load -i <chemin_vers_image_sauvegarde>

Remplacez <chemin_vers_image_sauvegarde> par le chemin d'accès à votre image de sauvegarde Docker.

Exécutez les conteneurs à partir de l'image restaurée:

```
docker run -d -p 5601:5601 --name kibana \
 --network elk_net \
 <nom_image_sauvegarde>
```

Remplacez <nom_image_sauvegarde> par le nom de votre image de sauvegarde Docker.

Restauration à partir d'une sauvegarde de données Elasticsearch:

Restaurez les données Elasticsearch: Utilisez l'outil elasticsearch-snapshot pour restaurer les données Elasticsearch à partir de votre sauvegarde. Consultez la documentation Elasticsearch pour plus d'informations sur la restauration des données.

Redémarrez les conteneurs Elasticsearch et Kibana:

bash

```
docker restart elasticsearch kibana
```

Use code with caution.

Restauration à partir d'une sauvegarde complète de la stack:

Arrêtez tous les conteneurs de la stack ELK:

bash

```
docker stop $(docker ps -q --filter network=elk_net)
```

Supprimez les conteneurs et les volumes de la stack ELK:

bash

```
docker rm $(docker ps -q --filter network=elk_net)
```

docker volume rm $(docker volume ls --filter network=elk_net)

Restaurez les données et les configurations à partir de votre sauvegarde complète.

Démarrez les conteneurs de la stack ELK:

Exécutez les commandes docker run pour démarrer les conteneurs Elasticsearch, Kibana, Logstash et Beats.

Remarques:

Assurez-vous de bien comprendre les implications de chaque méthode de restauration avant de l'utiliser. Il est recommandé de tester régulièrement votre procédure de restauration pour vous assurer qu'elle fonctionne correctement. Vous pouvez utiliser des outils d'automatisation comme Ansible ou Chef pour simplifier le processus de restauration.

1.3.2. Restauration des applicatifs

Décrire comment réaliser la restauration applicative.

1.3.3. Restauration des données

Chargez l'image de sauvegarde :

bash

docker load -i <chemin_vers_image_sauvegarde>

Remplacez <chemin_vers_image_sauvegarde> par le chemin d'accès à votre image de sauvegarde Docker.

Exécutez le conteneur à partir de l'image restaurée :

docker run -d <nom_image_sauvegarde>

Remplacez <nom_image_sauvegarde> par le nom de votre image de sauvegarde Docker.

Restauration à partir d'une sauvegarde de données :

Remarque: Cette méthode suppose que vous avez sauvegardé les données de votre application séparément de l'image Docker.

Restaurez les données de l'application. La méthode de restauration des données dépendra du type de données et du système de stockage que vous utilisez. Consultez la documentation de votre application ou de votre système de stockage pour plus d'informations.

Exécutez le conteneur Docker. Une fois les données restaurées, vous pouvez exécuter le conteneur Docker de votre application. La commande docker run spécifique dépendra de votre application.

Restauration à partir d'un environnement de staging :

Remarque: Cette méthode suppose que vous avez un environnement de staging pour votre application.

Arrêtez le conteneur de production de l’application.

bash

docker stop <nom_conteneur_production>

Remplacez <nom_conteneur_production> par le nom de votre conteneur de production Docker.

Copiez les données de l'environnement de staging vers l'environnement de production. La méthode de copie des données dépendra du système de stockage que vous utilisez. Vous pouvez utiliser des commandes Docker telles que docker cp ou docker exec pour copier les données.

Démarrez le conteneur de production de l’application.

bash

docker start <nom_conteneur_production>

Remplacez <nom_conteneur_production> par le nom de votre conteneur de production Docker.

Remarques :

Assurez-vous de bien comprendre les implications de chaque méthode de restauration avant de l'utiliser.

Il est important de tester régulièrement votre procédure de restauration pour vous assurer qu'elle fonctionne correctement.

Vous pouvez utiliser des outils d'automatisation comme Ansible ou Chef pour simplifier le processus de restauration applicative.

1.4. Procédure d’arrêt

1.4.1. Ordonnancement et séquencement

Pour ordonnancer l’arrêt de l’application ou de l’infrastructure ELK, il est recommandé de suivre l'ordre suivant :

Kibana : Arrêter l'interface utilisateur en premier.

Logstash : Arrêter le pipeline de traitement des logs.

Elasticsearch : Arrêter le moteur de recherche et d’analyse.

1.4.2. Arrêt global et validation

Pour réaliser l’arrêt global de l'infrastructure ELK, exécutez les commandes Docker suivantes dans l'ordre :

```
docker-compose down
```

1.4.3. Arrêt spécifique d’une application ou d'un service spécifique

Pour arrêter un service spécifique, utilisez la commande suivante en remplaçant service_name par le nom du service (kibana, logstash, elasticsearch) :

```
docker-compose stop service_name
```

1.5. Procédure de démarrage

1.5.1. Ordonnancement et dépendance

Pour ordonnancer le démarrage de l’infrastructure ELK, suivez cet ordre :

Elasticsearch : Démarrer en premier car Kibana et Logstash en dépendent.

Logstash : Démarrer ensuite pour qu'il puisse envoyer les données à Elasticsearch.

Kibana : Démarrer en dernier pour qu'il puisse se connecter à Elasticsearch.

1.5.2. Relance du serveur et des applications

Pour démarrer l'infrastructure ELK :

docker-compose up -d

Cela démarre tous les services en arrière-plan.

1.5.3. Relance d’une application ou d'un service spécifique

Décrire comment réaliser le démarrage spécifique de l’application ou d’un service.

1.6. Tests de bon fonctionnement

1.6.1. Contrôle quotidien des applications

Assurez-vous que tous les conteneurs sont en cours d'exécution et en bonne santé :

docker ps

Vérifiez les journaux des conteneurs pour détecter les erreurs :

docker-compose logs service_name

1.6.2. Plan de reboot régulier des serveurs ou composants

Planifiez des reboots réguliers en utilisant des tâches cron sur le serveur hôte pour redémarrer les conteneurs ELK :

0 3 * * * docker-compose restart

1.7. Pilotage des environnements

1.7.1. Logs

Les logs de chaque service peuvent être consultés via Docker :

docker-compose logs elasticsearch

docker-compose logs logstash

docker-compose logs kibana

1.7.2. Seuils et purges

Configurez les seuils et les purges dans Elasticsearch pour gérer la rétention des données. Utilisez les API d'Elasticsearch pour mettre en place des politiques de gestion du cycle de vie des index (ILM).

1.7.3. Traitements et batchs

Définissez et gérez les traitements batchs dans Logstash en configurant les pipelines dans le répertoire logstash/pipeline/.

1.7.4. Gestion des droits applicatifs

Configurez les utilisateurs et les rôles dans Kibana et Elasticsearch via les API de sécurité d'Elasticsearch.

1.8. Maintenance et support

1.8.1. Plage de maintenance

Précisez une plage de maintenance hebdomadaire, par exemple tous les dimanches de 02:00 à 04:00.

1.8.2. Mises à jour

Les mises à jour du Docker-ELK sont disponibles sur le dépôt GitHub docker-elk. Pour mettre à jour votre installation Docker-ELK, suivez les étapes suivantes :

Vérifiez les changements sur le dépôt :

Consultez les releases pour identifier les nouvelles versions et leurs notes de publication.

Téléchargez les nouvelles versions :

git pull origin master

Recréez les conteneurs :

docker-compose down docker-compose up -d

Vérifiez que tous les services sont opérationnels :

1. docker-compose ps

1.8.3. Contrats

Les contrats d'application couvrent les aspects suivants :

Horaires de support :

Niveau 1 : 24/7

Niveau 2 : Lundi à Vendredi, 08:00 - 18:00

Niveau 3 : Lundi à Vendredi, 09:00 - 17:00

Moyens de contact :

Niveau 1 : Numéro de téléphone, Email de support général

Niveau 2 : Numéro de téléphone spécifique, Email dédié

Niveau 3 : Email de contact direct des développeurs, Système de tickets

1.8.4. Licences

Les licences pour les composants ELK (Elasticsearch, Logstash, Kibana) sont disponibles selon les termes suivants :

Type de licences :

Elasticsearch : Elastic License 2.0 (ELv2)

Logstash : Apache License 2.0

Kibana : Elastic License 2.0 (ELv2)

Emplacement des licences :

Les termes des licences sont disponibles sur les pages officielles des composants respectifs.

Implémentation des licences :

Lors de l'utilisation des images Docker fournies par Elastic, les licences sont automatiquement intégrées dans les conteneurs. Aucune action supplémentaire n'est nécessaire pour leur implémentation, sauf si vous utilisez des fonctionnalités nécessitant une licence spécifique (par exemple, certaines fonctionnalités avancées de la suite Elastic).

1.9. Niveaux de support

1.9.1. Niveau 1

1.9.1.1. PLAGE HORAIRE

Plage horaire : 24/7

1.9.1.2. ACTEURS

Acteurs : Opérateurs IT

1.9.1.3. ACTIONS RÉALISÉES

Actions réalisées : Surveillance, redémarrage des services, escalade au niveau 2

1.9.2. Niveau 2

1.9.2.1. PLAGE HORAIRE

Plage horaire : Lundi à Vendredi, 08:00 - 18:00

1.9.2.2. ACTEURS

Acteurs : Administrateurs système

1.9.2.3. ACTIONS RÉALISÉES

Actions réalisées : Résolution des incidents, maintenance préventive, escalade au niveau 3.

1.9.3. Niveau 3

1.9.3.1. PLAGE HORAIRE

Plage horaire : Lundi à Vendredi, 09:00 - 17:00

1.9.3.2. ACTEURS

Acteurs : Développeurs ELK

1.9.3.3. ACTIONS RÉALISÉES

Actions réalisées : Débogage avancé, correctifs, retour au niveau 1 et 2

1.11.Sécurité

1.11.1. Conformité RGPD

Pour garantir la conformité RGPD (Règlement Général sur la Protection des Données), les mesures suivantes doivent être mises en œuvre :

Anonymisation des données :

Utilisez des techniques telles que le hachage, la pseudonymisation ou l'anonymisation pour protéger les données personnelles stockées dans Elasticsearch.

Configurez Logstash pour anonymiser les données avant leur ingestion.

Sécurité des accès :

Utilisez les fonctionnalités de sécurité d'Elasticsearch pour contrôler l'accès aux données, telles que le contrôle des accès basé sur les rôles (RBAC).

Activez l'authentification et le chiffrement des communications via TLS/SSL pour Elasticsearch, Logstash et Kibana.

Limitez l'accès aux données sensibles en configurant des permissions spécifiques pour les utilisateurs et les rôles.

Droits des utilisateurs :

Implémentez des mécanismes pour permettre aux utilisateurs d'exercer leurs droits (accès, rectification, suppression) conformément aux exigences du RGPD.

Conformité NIS

1.11.2. Conformité NIS

Pour garantir la conformité NIS (Network and Information Systems Directive), suivez ces directives :

Anonymisation des données :

Similaire à la conformité RGPD, utilisez des techniques pour anonymiser les données personnelles.

Sécurité des accès :

Configurez des pare-feu et des règles de sécurité pour restreindre l'accès aux services Elasticsearch, Logstash et Kibana.

Mettez en place des systèmes de détection et de prévention des intrusions (IDS/IPS) pour surveiller les activités suspectes.

1.11.3. Tests d'intrusion

Effectuez des tests d'intrusion réguliers pour identifier et corriger les vulnérabilités potentielles dans l'infrastructure ELK. Suivez les directives de l'ISO27001 pour garantir la sécurité de l’infrastructure. Les tests doivent inclure :

Tests de pénétration externes : Simulez des attaques extérieures pour tester la résistance de l'infrastructure contre les menaces externes.

Tests de pénétration internes : Simulez des attaques internes pour évaluer la sécurité des accès et des données sensibles.

Rapports et analyses : Documentez les résultats des tests et mettez en œuvre des actions correctives pour remédier aux vulnérabilités identifiées.

1.11.4. Homologation ISO27001

Pour obtenir et maintenir l'homologation ISO27001 :

Suivez les directives de l'ISO27001 : Implémentez un système de gestion de la sécurité de l'information (ISMS) conforme aux exigences de l'ISO27001.

Effectuez des audits réguliers : Réalisez des audits internes et externes pour vérifier la conformité aux normes ISO27001.

Mettez en œuvre les mesures correctives : Adressez les non-conformités et les vulnérabilités identifiées lors des audits et des tests d'intrusion.

1.12.Performances

1.12.1. Connexions concurrentes

Le nombre de connexions concurrentes que doit supporter l'infrastructure ELK dépend de la configuration matérielle et des ressources allouées. Pour une configuration typique, l'infrastructure doit pouvoir gérer plusieurs centaines à milliers de connexions concurrentes.

1.12.2. Temps de réponse attendus

Les temps de réponse que doit supporter l'application ou le service sont les suivants :

Elasticsearch : < 100 ms pour les requêtes de recherche simples.

Kibana : < 200 ms pour le chargement des dashboards.

Logstash : < 1 seconde pour le traitement des logs et l'envoi à Elasticsearch.

1.12.3. Test de charge

Pour réaliser des tests de charge :

Simulez des connexions concurrentes : Utilisez des outils comme Apache JMeter ou Gatling pour générer des charges de travail et mesurer les performances.

Mesurez les temps de réponse : Enregistrez les temps de réponse pour différentes charges et identifiez les points de contention.

Résultats des tests : Documentez les résultats obtenus. Par exemple :

500 connexions concurrentes : Temps de réponse moyen de 90 ms pour Elasticsearch.

1000 connexions concurrentes : Temps de réponse moyen de 150 ms pour Elasticsearch.

1.13.Support de formation

Mettez à disposition les supports de formation selon les profils des utilisateurs de l'infrastructure ELK :

Guides de démarrage rapide : Fournissez des documents de démarrage rapide pour les administrateurs système et les utilisateurs finaux.

Tutoriels vidéo : Créez des tutoriels vidéo couvrant les configurations de base, les tâches courantes et les meilleures pratiques.

Sessions de formation : Organisez des sessions de formation régulières pour les nouveaux utilisateurs et les administrateurs.

Documentation détaillée : Fournissez une documentation détaillée et à jour sur la configuration et l'utilisation de l'infrastructure ELK.

Ces mesures garantiront que votre infrastructure ELK est sécurisée, performante et conforme aux exigences réglementaires.