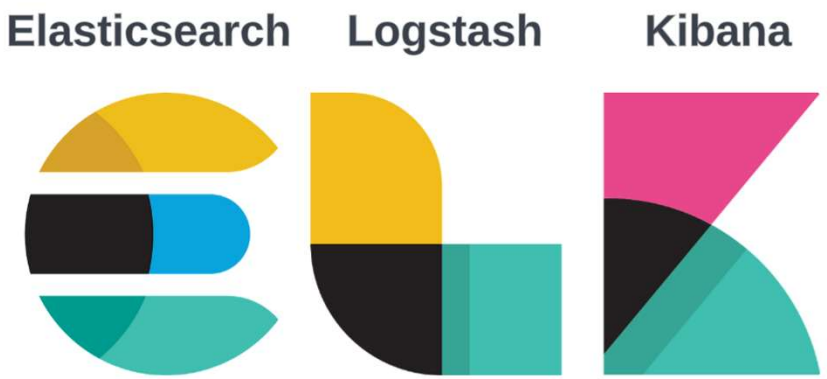
ELK



Le contexte

Contexte et besoin

- Un développeur de l'entreprise a soumis une demande via le système de ticketing.
- Il souhaitait disposer d'un outil lui permettant de **centraliser et visualiser les logs** de ses environnements de travail afin de faciliter ses missions quotidiennes.
- Mon rôle : mettre en place et livrer ce service de manière opérationnelle et documentée.

La démarche

-Pour le SSH, nous désactivons l'accès root, imposons l'utilisation du protocole 2 et limitons l'accès au seul utilisateur « developpeur » crée. Enfin, nous redémarrons le service afin d'appliquer l'ensemble des modifications.

Commande pour la configuration du SSH :

- PermitRootLogin no
- Protocol 2
- PasswordAuthentication yes
- AllowUsers developpeur

-Concernant la gestion des comptes, nous verrouillons les comptes inutilisés et mettons en place une politique de mot de passe renforcée. Pour cela, nous utilisons le module « libpam-pwquality ». Parmi les comptes désactivés : games, nobody, daemon, man, news, uucp, Debian-exim, irc.

Commande pour verrouiller les comptes:

- passwd -l games
- passwd -l nobody
- passwd -l daemon

Commande pour politique de mot de passe :

- apt install libpam-pwquality -y
- nano /etc/security/pwquality.conf
- minlen = 12 | dcredit = -1 | ucredit = -1 | ocredit = -1 | lcredit = -1 | retry = 3 | enforce_for_root

-Les développeurs disposent d'un compte dédié nommé "developpeur", dont le mot de passe est conservé dans le KeePass du SI d'Enfrasys. Leur élévation de privilèges est strictement limitée à l'utilisation de « sudo », et ils ne peuvent pas se connecter directement au compte root. Les administrateurs, quant à eux, disposent uniquement d'un accès « root », mais doivent impérativement se connecter localement à la machine virtuelle et non via SSH.

Commande pour installer Docker :

- apt remove docker.io docker-doc docker-compose podman-docker containerd runc
- apt install -y ca-certificates curl
- sudo install -m 0755 -d /etc/apt/keyrings
  sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
  sudo chmod a+r /etc/apt/keyrings/docker.asc
- echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian trixie stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list
- sudo apt update
- sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
- sudo usermod -aG docker $USER

Installation ELK avec Docker

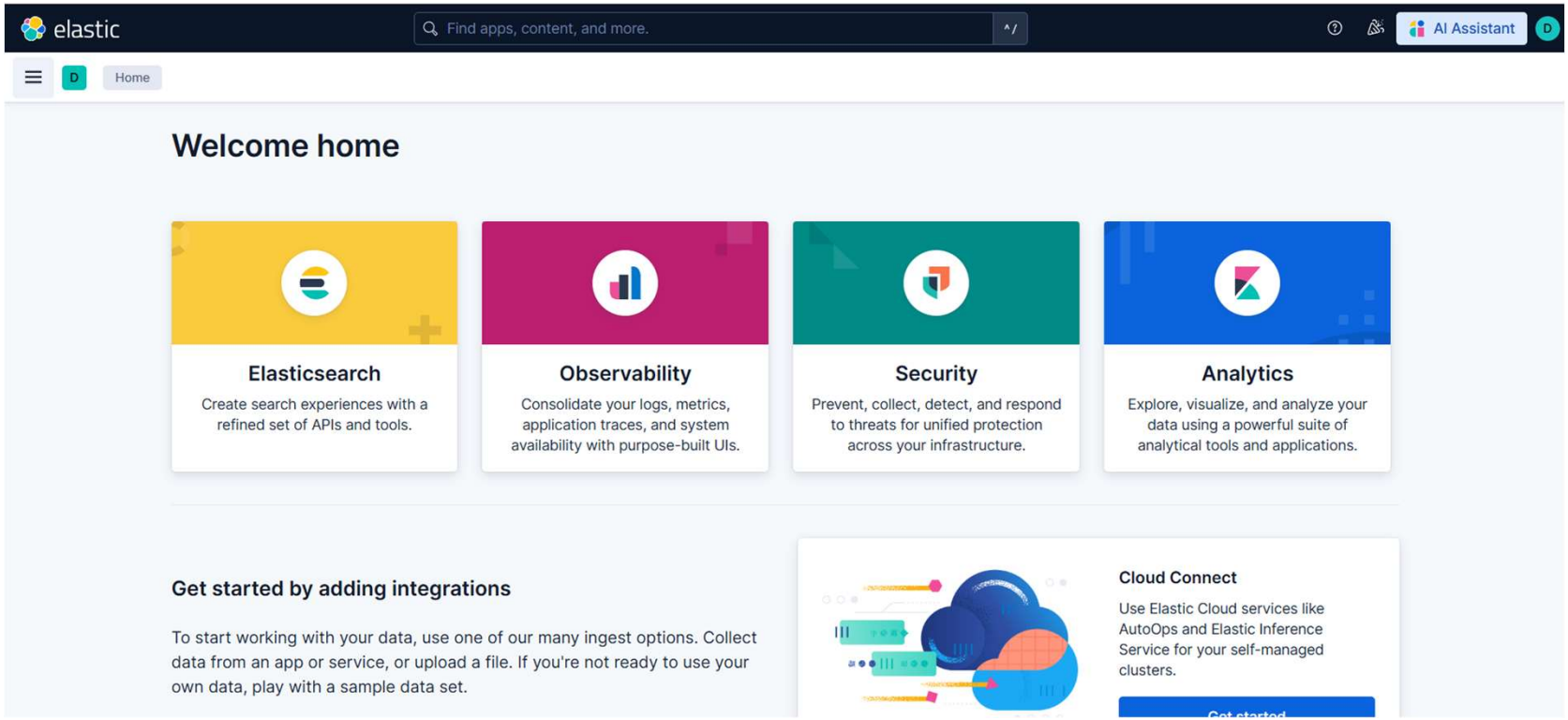
Maintenant, nous allons procéder à l'installation de l'ELK (Elasticsearch + LogStach + Kiban
Le but étant qu'Elasticsearch soit le moteur de recherche et stockage, Logstash sera le pipe
de traitement des logs et Kibana pour la visualisation :

Commande pour installer l'ELK :

- mkdir elk-stack
- cd elk-stack
- nano /home/developpeur/elk-stack
-

```
  container_name: elasticsearch
  environment:
    - discovery.type=single-node
    - xpack.security.enabled=true
    - ES_JAVA_OPTS=-Xms1g -Xmx1g
  ports:
    - "9200:9200"
  volumes:
    - esdata:/usr/share/elasticsearch/data
  ulimits:
    memlock:
      soft: -1
      hard: -1

kibana:
  image: docker.elastic.co/kibana/kibana:9.3.0
  restart: unless-stopped
  container_name: kibana
  environment:
    - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
    - ELASTICSEARCH_USERNAME=kibana_system
    - ELASTICSEARCH_PASSWORD=${KIBANA_SYSTEM_PASSWORD}

    - XPACK_ENCRYPTEDSAVEDOBJECTS_ENCRYPTIONKEY=${KIBANA_ENCRYPTEDSAVEDOBJECTS_KEY}
    - XPACK_SECURITY_ENCRYPTIONKEY=${KIBANA_SECURITY_KEY}
    - XPACK_REPORTING_ENCRYPTIONKEY=${KIBANA_REPORTING_KEY}

  ports:
    - "5601:5601"
  depends_on:
    - elasticsearch

logstash:
  image: docker.elastic.co/logstash/logstash:9.3.0
  restart: unless-stopped
  container_name: logstash
  ports:
    - "5044:5044"        # Filebeat input
    - "5000:5000/tcp"    # TCP input
    - "5000:5000/udp"    # UDP input
    - "9600:9600"        # Logstash API
  volumes:
```

La mise en place du service



L’accompagnement des utilisateurs

Pour voir les dashboard, il faut aller dans Elasticsearch > Home > Dashboards. Il recensera actuellement tous les dashboards crées.

Pour rechercher des logs, il faut aller dans Observability > Logs. Il recensera tous les logs crées.

Si vous souhaitez modifier le mot de passe, il suffira de cliquer sur le profile en haut a droite, puis cliquer sur Edit profile et ainsi cliquer sur Change password.

Les prérequis techniques sont d’avoir une machine virtuelle qui disposent déjà d’une VM durci avec Debian dernière version pour correspondre aux normes ISO 27001. Il faudra donc installer docker, puis récupérer les dépendances des images de docker compose permettant de pouvoir exécuter les conteneurs Dockers. L’ELK va donc être basé sur docker avec comme modules l’Elasticsearch, Logstash, et Kibana. Il faudra par ailleurs ouvrir le port sur le firewall de ElasticSearch qui est le 5601.

La procédure d’installation est dans les premières pages.

On valide la mise en production à partir du moment où l’applicatif est mise en ligne sur le réseau local de l’entreprise et qui correspond bien aux attentes de ce que demande l’utilisateur pour la mise en place d’un service informatique. Une fois que la VM est bien sécurisé et correspond aux normes de l’ISO 27001, on peut donc passer en production.

Pour voir si tout fonctionne bien, nous allons pouvoir créer un Dashboard.

On peut faire glisser des visualisations pour tester si cela marche. On peut voir que tout est parfait :