Projet Appscore - Logs

Le plus simple dans Kubernetes est d’utiliser la stack **EFK** (Elasticsearch + Fluentd + Kibana) qui est plus légère et facile à déployer dans un cluster local type Docker Desktop.

J'ai choisi la stack **EFK** (Elasticsearch, Fluentd, Kibana). **Fluentd** tourne en DaemonSet pour collecter les flux `stdout` de tous les conteneurs, les envoie à **Elasticsearch** pour l'indexation, et j'utilise **Kibana** pour la visualisation.

Le Plan d'action (La Stack EFK)

- **Elasticsearch (Le Cerveau/Stockage) :** C'est une grosse base de données qui va indexer tous tes logs pour qu'on puisse faire des recherches rapides (ex: "trouve moi les erreurs 500").
- **Fluentd (Le Facteur) :** C'est lui qui va lire les fichiers de logs cachés dans ton cluster et les envoyer à Elasticsearch.
- **Kibana (L'Interface) :** C'est la page web où tu vas créer ton beau Dashboard avec des graphiques.

```
kubectl create namespace logging
```

Clique sur **"Explore on my own"**. Maintenant, on va remplir la consigne de ton M2 (le Dashboard et le Filtre).

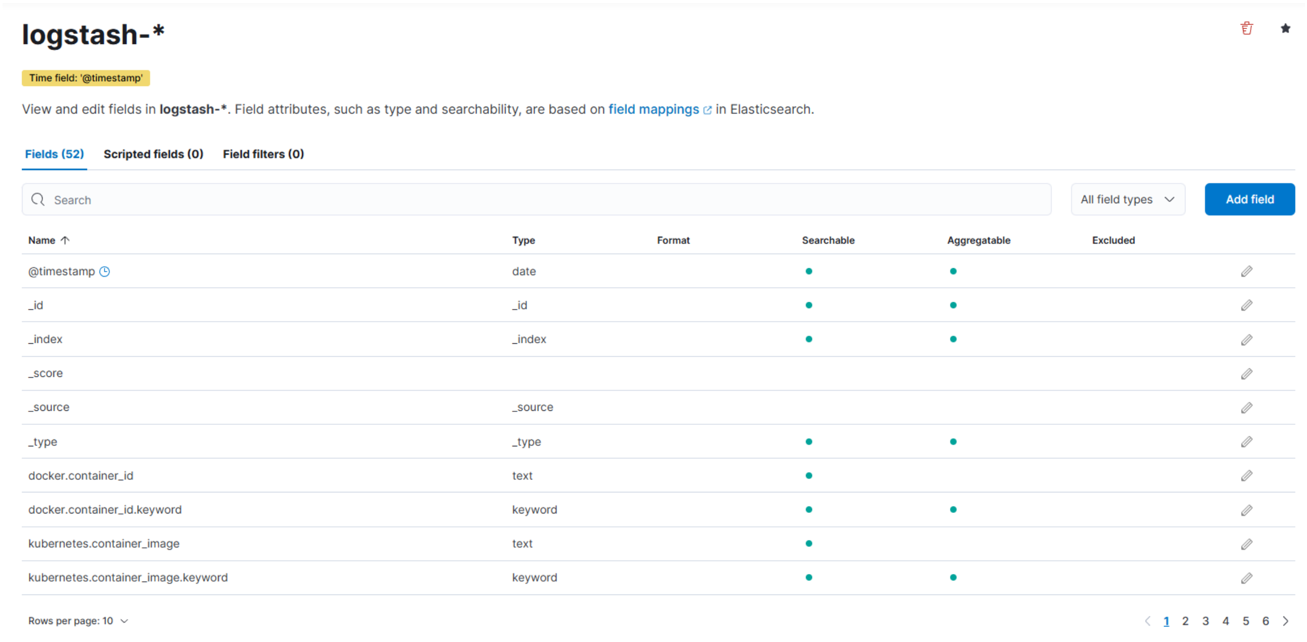
Voici la marche à suivre exacte :

Connecter les données (L'Index Pattern)

Kibana est vide par défaut, il faut lui dire d'aller lire ce que Fluentd envoie dans Elasticsearch.

1. Ouvre le menu (les 3 barres horizontales en haut à gauche).
2. Descends tout en bas jusqu'à **Stack Management**.
3. Dans le menu de gauche, clique sur **Index Patterns**.
4. Clique sur le bouton bleu **Create index pattern**.
5. Dans le champ "Name", tape : `logstash-*` (Fluentd utilise ce nom par défaut).
6. Clique sur **Next step**.
7. Dans "Time field", choisis **@timestamp**.

8. Clique sur **Create index pattern**.



logstash-*

Time field: '@timestamp'

View and edit fields in **logstash-***. Field attributes, such as type and searchability, are based on field mappings in Elasticsearch.

Fields (52) | Scripted fields (0) | Field filters (0)

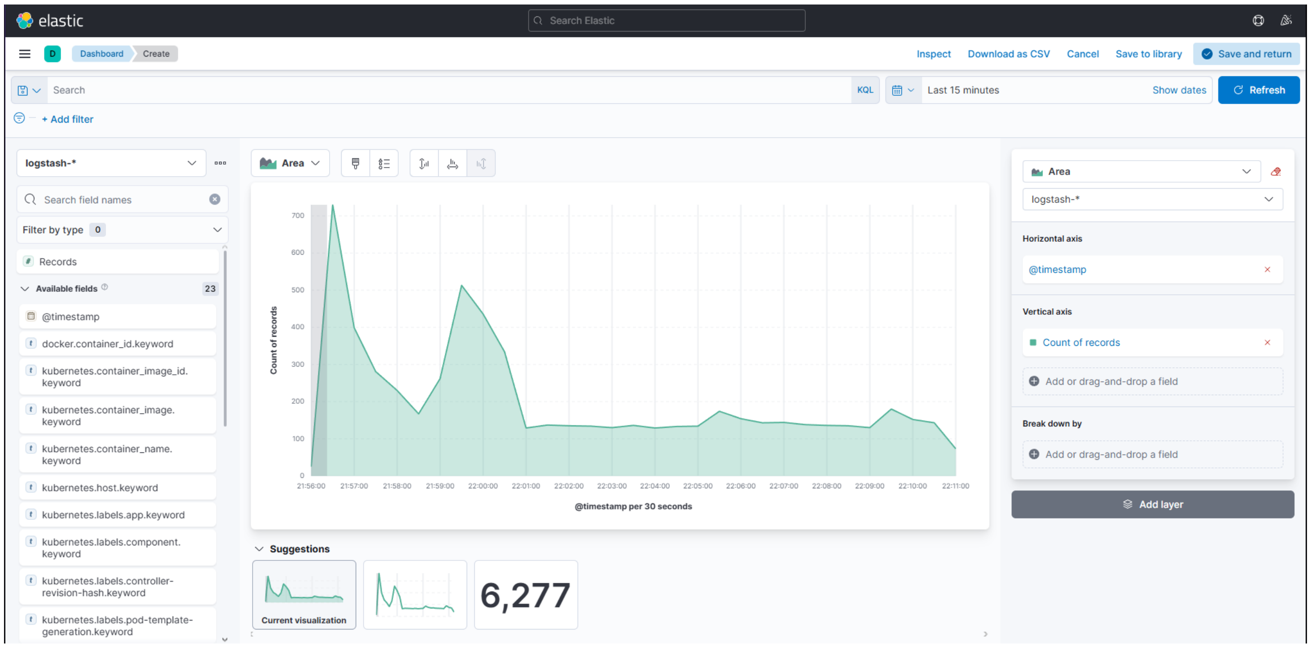
Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	
_id	_id		●	●	
_index	_index		●	●	
_score					
_source	_source				
_type	_type		●	●	
docker.container_id	text		●		
docker.container_id.keyword	keyword		●	●	
kubernetes.container_image	text		●		
kubernetes.container_image.keyword	keyword		●	●	

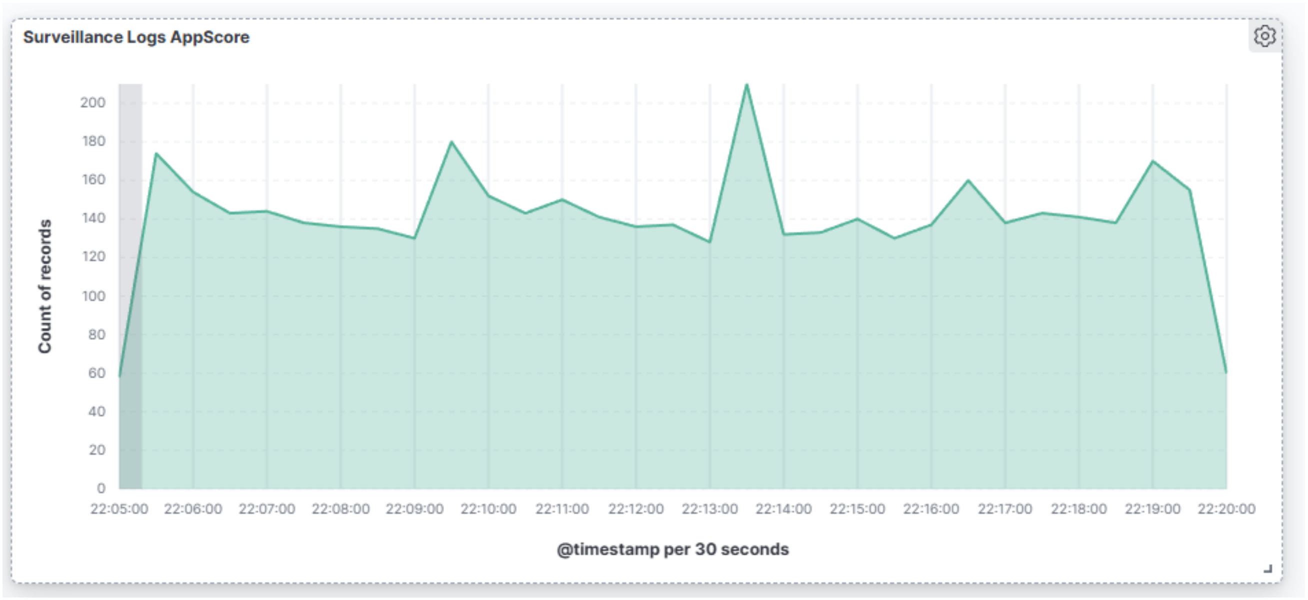
Rows per page: 10 — 1 2 3 4 5 6

Créer le Dashboard

Pour la partie "Visualisation" de ton bilan de santé :

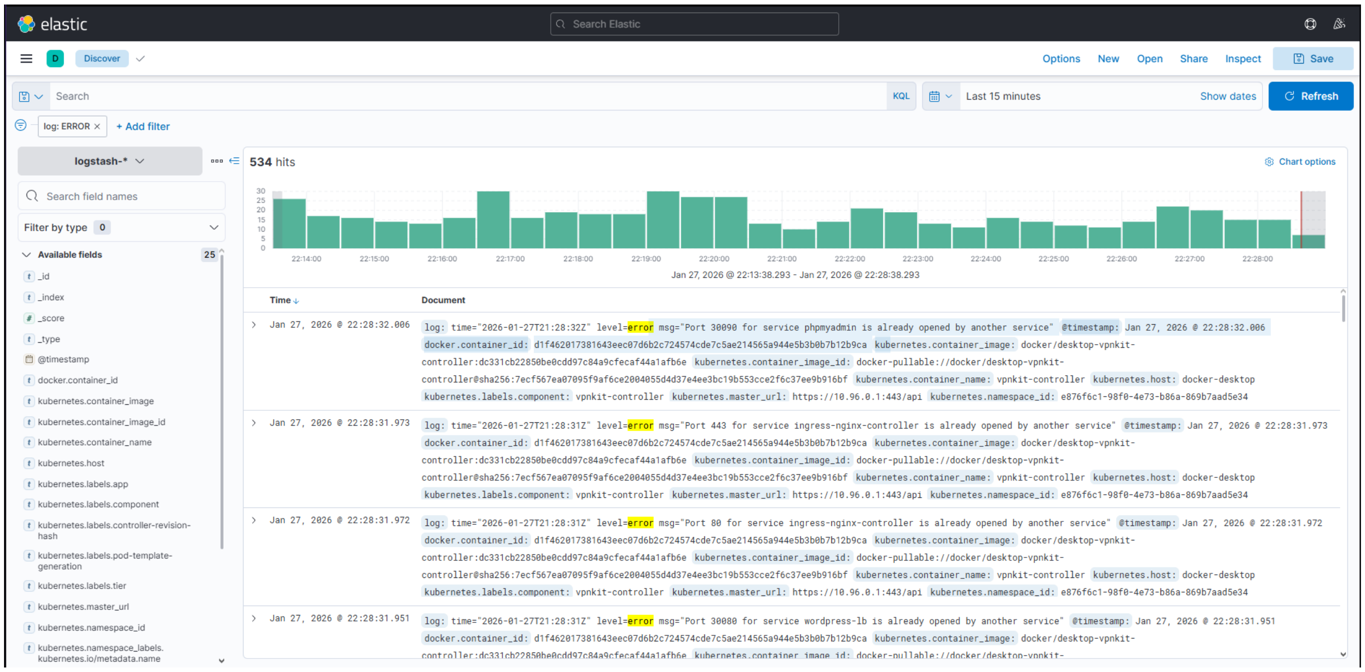
1. Va dans le menu > **Dashboard**.
2. Clique sur **Create new dashboard** puis **Create visualization**.
3. Choisis le type **Area** ou **Bar** (le plus simple).
4. Glisse le champ **@timestamp** sur l'axe horizontal (X) et **Count** sur l'axe vertical (Y).
5. Tu verras un graphique qui montre le volume de logs en temps réel.





Ici, j'ai mon dashboard de logs. Si je vois un pic rouge, c'est que mon application génère des erreurs. Je peux filtrer par service pour savoir exactement qui pose problème.

Page discover :



```
Jan 27, 2026 @ 22:28:32.006  log: time="2026-01-27T21:28:32Z" level=error msg="Port 30090 for service phpmyadmin is already opened by another service" @timestamp: Jan 27, 2026 @ 22:28:32.006
docker.container_id: d1f462017381643eec07d6b2c724574cde7c5ae214565a944e5b3b0b7b12b9ca kubernetes.container_image: docker/desktop-vpnkit-
controller:dc331cb22850be0cdd97c84a9cfecaf44a1afb6e kubernetes.container_image_id: docker-pullable://docker/desktop-vpnkit-
controller@sha256:7ecf567ea07095f9af6ce2004055d4d37e4ee3bc19b553cce2f6c37ee9b916bf kubernetes.container_name: vpnkit-controller kubernetes.host: docker-desktop
kubernetes.labels.component: vpnkit-controller kubernetes.master_url: https://10.96.0.1:443/api kubernetes.namespace_id: e876f6c1-98f0-4e73-b86a-869b7aad5e34

Jan 27, 2026 @ 22:28:31.973  log: time="2026-01-27T21:28:31Z" level=error msg="Port 443 for service ingress-nginx-controller is already opened by another service" @timestamp: Jan 27, 2026 @ 22:28:31.973
docker.container_id: d1f462017381643eec07d6b2c724574cde7c5ae214565a944e5b3b0b7b12b9ca kubernetes.container_image: docker/desktop-vpnkit-
controller:dc331cb22850be0cdd97c84a9cfecaf44a1afb6e kubernetes.container_image_id: docker-pullable://docker/desktop-vpnkit-
controller@sha256:7ecf567ea07095f9af6ce2004055d4d37e4ee3bc19b553cce2f6c37ee9b916bf kubernetes.container_name: vpnkit-controller kubernetes.host: docker-desktop
kubernetes.labels.component: vpnkit-controller kubernetes.master_url: https://10.96.0.1:443/api kubernetes.namespace_id: e876f6c1-98f0-4e73-b86a-869b7aad5e34

Jan 27, 2026 @ 22:28:31.972  log: time="2026-01-27T21:28:31Z" level=error msg="Port 80 for service ingress-nginx-controller is already opened by another service" @timestamp: Jan 27, 2026 @ 22:28:31.972
docker.container_id: d1f462017381643eec07d6b2c724574cde7c5ae214565a944e5b3b0b7b12b9ca kubernetes.container_image: docker/desktop-vpnkit-
controller:dc331cb22850be0cdd97c84a9cfecaf44a1afb6e kubernetes.container_image_id: docker-pullable://docker/desktop-vpnkit-
controller@sha256:7ecf567ea07095f9af6ce2004055d4d37e4ee3bc19b553cce2f6c37ee9b916bf kubernetes.container_name: vpnkit-controller kubernetes.host: docker-desktop
kubernetes.labels.component: vpnkit-controller kubernetes.master_url: https://10.96.0.1:443/api kubernetes.namespace_id: e876f6c1-98f0-4e73-b86a-869b7aad5e34

Jan 27, 2026 @ 22:28:31.951  log: time="2026-01-27T21:28:31Z" level=error msg="Port 30080 for service wordpress-lb is already opened by another service" @timestamp: Jan 27, 2026 @ 22:28:31.951
docker.container_id: d1f462017381643eec07d6b2c724574cde7c5ae214565a944e5b3b0b7b12b9ca kubernetes.container_image: docker/desktop-vpnkit-
controller:dc331cb22850be0cdd97c84a9cfecaf44a1afb6e kubernetes.container_image_id: docker-pullable://docker/desktop-vpnkit-
```

Grâce à l'indexation, je peux isoler instantanément les erreurs critiques sans avoir à fouiller manuellement dans les logs de chaque Pod via la console.

"*Pourquoi EFK et pas ELK ?*"

Réponse : Fluentd est plus adapté à Kubernetes (Cloud Native) que Logstash. Il consomme moins de ressources et s'intègre nativement avec les métadonnées des Pods (Namespace, Pod Name, Labels), ce qui facilite grandement le filtrage que je viens de vous montrer.