

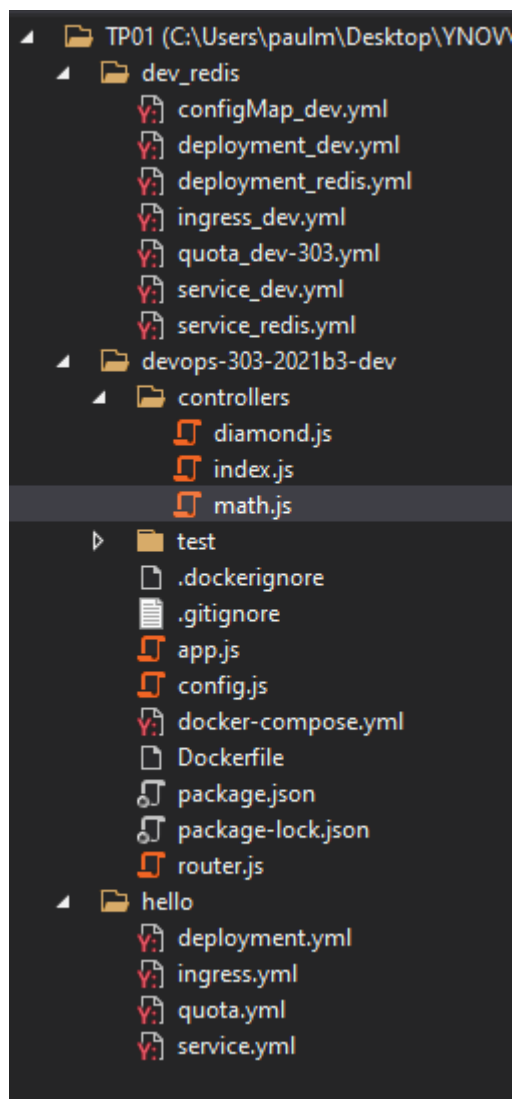
RENDU DEVOPS

Dans cette présentation, nous vous détaillerons les principales étapes pour mettre en place les différentes fonctionnalités.

Tout nous avons mis précédemment en place une série de .yaml et de Dockerfile afin d'afficher sur une page 3 hello-world.

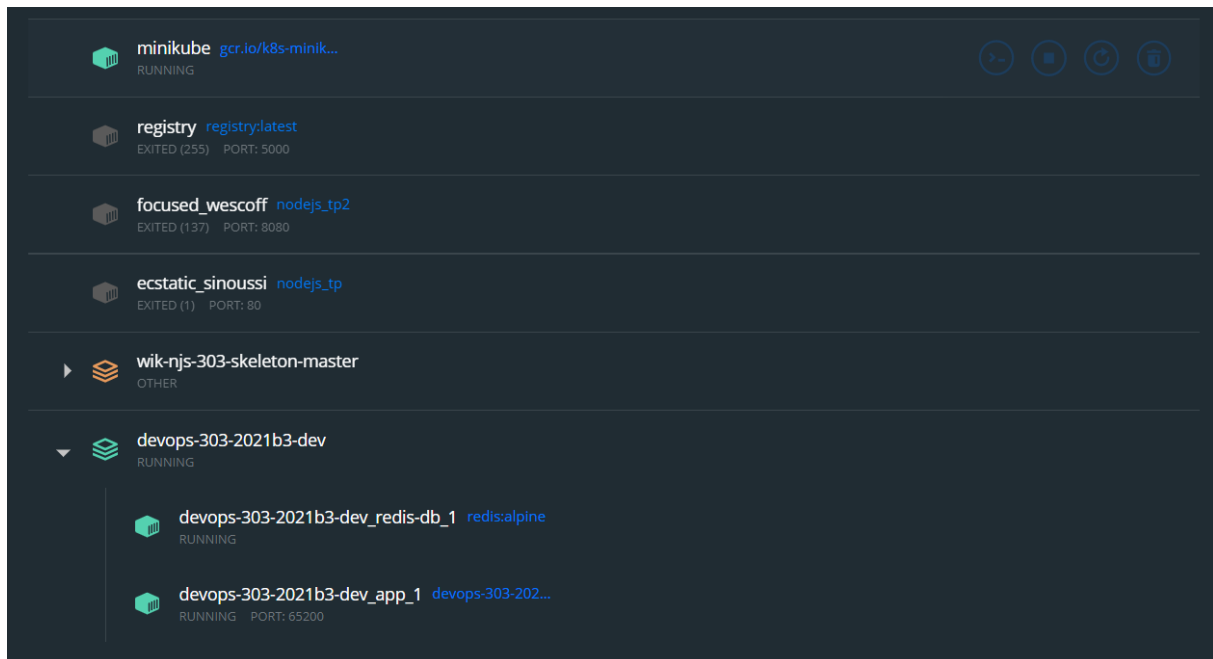
Vous pouvez retrouver les différentes configurations dans les .yaml ci-joint qui regroupe d'un cotés un dossier pour redis et un pour hello qui contiennent tous les 2 :

- Deployment
- Ingress
- Service



(pour redis les fichiers sont coupés en `_dev` pour la partie *développement devops-303* et `_redis` pour mettre en place la gestion de *Bdd « redis »*).

Nous avons ci-dessous ce que cela donne dans docker.



Voici les lignes de commande principale pour mettre en place cela.

1. `minikube start`
2. `minikube addons enable ingress`
3. `minikube image load devops-303-2021b3-dev_app`
4. `kubectl apply -f [...].yaml`
5. `minikube service -n ingress-nginx ingress-nginx-controller`

Ensuite nous devons mettre en place les `ressource quotas` et les `namespaces`.

Pour les `quotas` nous nous servons d'un `.yaml`

```
1  apiVersion: v1
2  kind: ResourceQuota
3  metadata:
4    name: mem-cpu-dev
5  spec:
6    hard:
7      requests.cpu: "1"
8      requests.memory: "1G"
9      limits.cpu: "2"
10     limits.memory: "2G"
```

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: mem-cpu-hello
spec:
  hard:
    requests.cpu: "300m"
    requests.memory: "384M"
    limits.cpu: "600m"
    limits.memory: "768M"
```

A gauche pour redis et a droite pour les hello

Ensuite nous créons les **namespace** dans le terminal de commande avec :

- `kubectl create namespace < name>`

```
C:\Users\paulm>kubectl get namespace
NAME                STATUS    AGE
default             Active    41d
dev-303             Active    13d
hello-303           Active    13d
ingress-nginx       Active    33d
kube-node-lease     Active    41d
kube-public         Active    41d
kube-system         Active    41d
```

Nous mettons ensuite les composants correspondant dans les **namespaces** que nous pouvons visualiser via describe :

```
C:\Users\paulm>kubectl describe namespace dev-303
Name:          dev-303
Labels:        kubernetes.io/metadata.name=dev-303
Annotations:   <none>
Status:        Active

Resource Quotas
  Name:          mem-cpu-dev
  Resource       Used   Hard
  -----
  limits.cpu     400m   2
  limits.memory  512Mi  2G
  requests.cpu   400m   1
  requests.memory 484Mi  1G
```

Nous pouvons donc par exemple voir les pods qui le compose :

```
C:\Users\paulm>kubectl get pods -n dev-303
NAME                                READY   STATUS    RESTARTS   AGE
devops3030-5bcf54d6fc-8tz85        1/1     Running   1 (13d ago) 13d
devops3030-5bcf54d6fc-c6d2n        1/1     Running   1 (13d ago) 13d
devops3030-5bcf54d6fc-nd2c7        1/1     Running   1 (13d ago) 13d
redis-leader-587d9df6ff-jbf2h      1/1     Running   1 (13d ago) 13d

C:\Users\paulm>kubectl get pods -n prometheus
NAME                                READY   STATUS    RESTARTS   AGE
alertmanager-prometheus-kube-prometheus-alertmanager-0  2/2     Running   0           3m29s
prometheus-grafana-74b9c448f6-nnxhj  2/2     Running   0           3m32s
prometheus-kube-prometheus-operator-577bb648c5-7tp5s    1/1     Running   0           3m32s
prometheus-kube-state-metrics-58c5cd6ddb-rx2vb          1/1     Running   0           3m32s
prometheus-prometheus-kube-prometheus-prometheus-0      2/2     Running   0           3m28s
prometheus-prometheus-node-exporter-4zlqd               1/1     Running   0           3m32s
```

Ensuite nous faisons des **configmap** :

```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: config-dev-303
5    namespace: dev-303
6  data:
7    PORT: "8884"
8    REDIS_PORT: "6379"
9    REDIS_HOST: "redis-leader"
```

Quant nous relançons cela dans le terminal

```
C:\Windows\system32>minikube service -n ingress-nginx ingress-nginx-controller
```

NAMESPACE	NAME	TARGET PORT	URL
ingress-nginx	ingress-nginx-controller	http/80	http://192.168.49.2:32667
		https/443	http://192.168.49.2:30752

* Tunnel de démarrage pour le service ingress-nginx-controller.

NAMESPACE	NAME	TARGET PORT	URL
ingress-nginx	ingress-nginx-controller		http://127.0.0.1:63803
			http://127.0.0.1:63804

Et que nous allons sur internet avec le bon name et le bon port nous obtenons cela :

```
← → ↻ devops3030.mourgues-paul:55883

Hello everybody !!
---
Hostname: devops3030-5bcf54d6fc-8tz85
Server IP: 172.17.0.2
---
Count in this container launch: 14 times
Count for this container: 14 times
Count for all container: 39 times
---
Container name      | Visit count
devops3030-5bcf54d6fc-8tz85 | 14
devops3030-5bcf54d6fc-c6d2n | 13
devops3030-5bcf54d6fc-nd2c7 | 12
```

Nous passons ensuite à la partie faite chez soi, à savoir **prometheus & Grafana**

Tout d'abord nous installons helm :

- **choco install kubernetes-helm**

Puis avec helm nous installons ensuite prometheus

```
C:\Users\paul>helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories

C:\Users\paul>helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. ☺Happy Helming!☺

C:\Users\paul>helm install prometheus prometheus-community/kube-prometheus-stack
W1128 18:34:06.482829 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:06.485992 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:06.489178 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:06.492371 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:06.495002 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:06.497627 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:06.500261 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:07.707063 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:08.141043 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.863714 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.911621 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.913217 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.913217 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.913217 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.913217 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.913217 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:12.913217 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:14.690461 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:15.121971 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W1128 18:34:19.173198 17024 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
NAME: prometheus
LAST DEPLOYED: Sun Nov 28 18:34:05 2021
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace default get pods -l "release=prometheus"

Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.

C:\Users\paul>kubectl create namespace prometheus
namespace/prometheus created
```

Nous regardons les events afin de voir quoi utiliser ensuite :

```
C:\Users\paul>kubectl get events -A
NAMESPACE LAST SEEN TYPE REASON OBJECT MESSAGE
default 12m Normal Scheduled pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Successfully assigned default/alertmanager-prometheus-kube-prometheus-alertmanag
er-0 to minikube
default 12m Normal Pulling pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Pulling image "quay.io/prometheus/alertmanager:v0.23.0"
default 12m Normal Pulled pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Successfully pulled image "quay.io/prometheus/alertmanager:v0.23.0" in 17.776144
ss
default 12m Normal Created pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Created container alertmanager
default 12m Normal Started pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Started container alertmanager
default 12m Normal Pulling pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Pulling image "quay.io/prometheus-operator/prometheus-config-reloader:v0.52.0"
default 12m Normal Pulled pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Successfully pulled image "quay.io/prometheus-operator/prometheus-config-reloade
r:v0.52.0" in 21.8777802s
default 12m Normal Created pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Created container config-reloader
default 12m Normal Started pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Started container config-reloader
default 6m17s Normal Killing pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Stopping container alertmanager
default 6m17s Normal Killing pod/alertmanager-prometheus-kube-prometheus-alertmanager-0 Stopping container config-reloader
```

Reinstallation des packages grafana (se referer a la ligne suivante) :

```
mourguesp@DESKTOP-H70AME7:~/kube-prometheus$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
OK
mourguesp@DESKTOP-H70AME7:~/kube-prometheus$ sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 https://packages.grafana.com/oss/deb stable InRelease [12.1 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:6 https://packages.grafana.com/oss/deb stable/main amd64 Packages [26.2 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1026 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [191 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [9072 B]
Get:11 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [526 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [75.4 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [528 B]
Get:14 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [663 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [111 kB]
Get:17 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [12.9 kB]
Get:18 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [21.9 kB]
Get:19 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [4948 B]
Get:20 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [540 B]
```

Ayant eu un probleme de connexion avec grafana, nous avons essayé de redefinir les MDP :

```

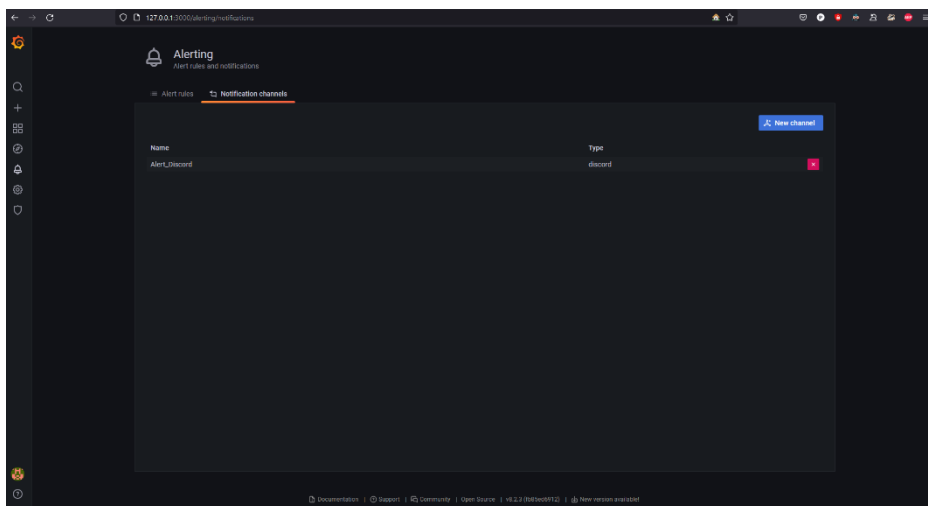
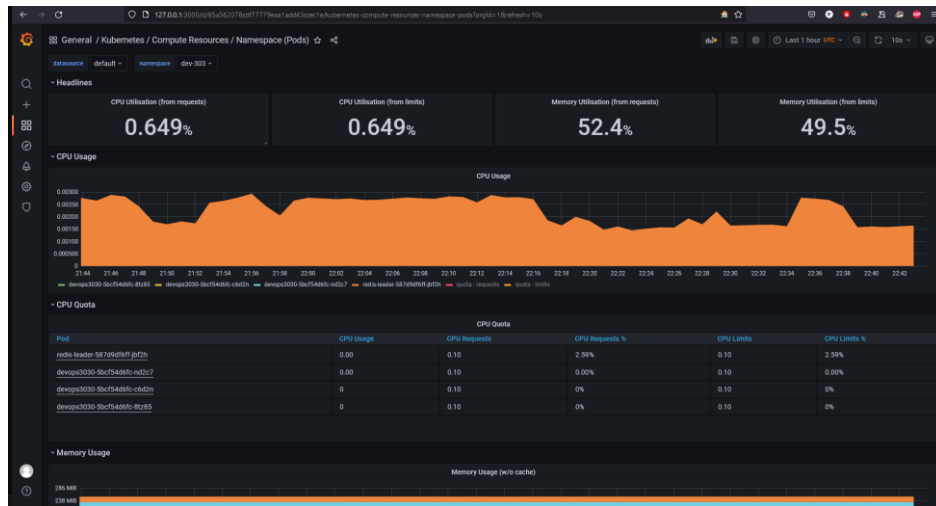
root@k8s-master:~# kubectl -n kube-prometheus$ sudo apt install grafana
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  grafana
0 upgraded, 1 newly installed, 0 to remove and 88 not upgraded.
Need to get 67.3 MB of archives.
After this operation, 225 MB of additional disk space will be used.
Get:1 https://packages.grafana.com/oss/deb/main amd64 grafana amd64 8.2.5 [67.3 MB]
Fetched 67.3 MB in 9s (7668 kB/s)
Selecting previously unselected package grafana.
(Reading database ... 32290 files and directories currently installed.)
Preparing to unpack .../grafana-8.2.5_amd64.deb ...
Unpacking grafana (8.2.5) ...
Setting up grafana (8.2.5) ...
Adding system user `grafana' (UID 112) ...
Adding new user `grafana' (UID 112) with group `grafana' ...
Not creating home directory `/usr/share/grafana'.
## NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable grafana-server
## You can start grafana-server by executing
sudo /bin/systemctl start grafana-server
Processing triggers for systemd (245.4-4ubuntu3.11) ...
root@k8s-master:~# kubectl -n kube-prometheus$ sudo apt install grafana-cli
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package grafana-cli
root@k8s-master:~# kubectl -n kube-prometheus$ sudo grafana-cli admin reset-admin-password admin
WARN[11-28|19:49:28] falling back to legacy setting of 'min_interval_seconds', please use the configuration option in the 'unified_alerting' section if Grafana 8 alerts are enabled. logger=settings
WARN[11-28|19:49:28] falling back to legacy setting of 'min_interval_seconds', please use the configuration option in the 'unified_alerting' section if Grafana 8 alerts are enabled. logger=settings
INFO[11-28|19:49:28] Config loaded from logger=settings file=/usr/share/grafana/conf.defaults.ini
INFO[11-28|19:49:28] Config loaded from logger=settings file=/etc/grafana/grafana.ini
INFO[11-28|19:49:28] Config overridden from command line logger=settings arg=default.paths.data=/var/lib/grafana
INFO[11-28|19:49:28] Config overridden from command line logger=settings arg=default.paths.logs=/var/lib/grafana
INFO[11-28|19:49:28] Config overridden from command line logger=settings arg=default.paths.plugins=/var/lib/grafana/plugins
INFO[11-28|19:49:28] Config overridden from command line logger=settings arg=default.paths.provisioning=/etc/grafana/provisioning
INFO[11-28|19:49:28] Path Home logger=settings path=/usr/share/grafana

```

Une fois que le problème a été résolu (le MDP était prom-operator à la place de admin) grafana se lance :

[illegible]

Nous pouvons accéder à l'interface afin de pouvoir voir différentes statistiques et mettre en place des alertes (ici un bot discord)



Voici (en meilleure qualité que sur les screen) la liste des commandes effectuées pour arriver au résultat:

```
kubectl get rs
```

```
kubectl get namespace
```

```
kubectl describe namespace dev-303
```

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

```
helm install prometheus prometheus-community/kube-prometheus-stack --namespace=prometheus
```

```
helm install prometheus prometheus-community/kube-prometheus-stack
```

```
kubectl get events -A
```

```
kubectl describe pods -n prometheus
```

```
kubectl get pods -n prometheus
```

```
git clone https://github.com/prometheus-operator/kube-prometheus.git
```

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

```
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
```

```
sudo apt install grafana
```

```
sudo grafana-cli admin reset-admin-password admin
```

```
kubectl port-forward -n prometheus prometheus-grafana-74b9c448f6-nnxhj 3000
```

Nous nous sommes retrouvé coincé à cette étape là avec RBAC donc c'est ici que ce termine notre travail.