



# Guide de maintenance

**XXII CORE REAL TIME  
V.2.0.0**

Version guide - 1.0

# Auteurs & historique des modifications.

## Contributeurs

Auteurs	Postes
Nathan LAFONTAINE	Technical Manager
Stanislas MAHIEUX	Directeur de la formation

## Historique du document

Auteurs	Date	Modifs
Nathan LAFONTAINE	5 Janvier 2023	Création du document
Stanislas MAHIEUX	10 Janvier 2023	Création du document



# Sommaire.

Auteurs & historique des modifications.	2
Contributeurs	2
Historique du document	2
1. Introduction.	6
2. Le processor de debug.	6
2.1. Qu'est ce que le processor de debug?	6
2.2. Activation du processor de debug.	6
3. Logiciel d'orchestration.	8
3.1. Connexion en ssh à la machine	8
3.1.1 Depuis un PC Windows	8
3.1.2 Depuis un PC Linux	9
3.1.3 Depuis un PC MAC	9
3.2. Utilisation de Grafana	9
3.2.1. Introduction	9
3.2.2 Accéder à Grafana	10
3.2.3. Liste des sources par défaut et celles de XXII	10
3.2.3.1. Prometheus	10
3.2.3.2. Dashboard	12
3.2.3.4. Alertes	13
3.2.3.5 Loki	14
3.2.4. Les tableaux	16
3.2.4.1. Nodes	16
3.2.4.2. Node Exporter Full	16
3.2.4.3. Kubernetes / Compute Resources / Namespace (Pods)	17
3.2.4.4. Kubernetes / Compute Resources / Pod	18
3.2.4.5. Prometheus Alerts	18
3.3. K9S	19
3.3.1. Introduction à K9S	19
3.3.2. Présentation de l'interface	19
3.3.3. Les différents modes	20

3.3.3.1. Le mode commande	20
3.3.3.2. Le mode filtre	22
3.3.3.3. Le panneau des raccourcis	23
3.3.3.4. Afficher les logs	24
3.3.3.5. Detruire un pod	25
3.3.3.6. Décrire / Editer une ressource	26
3.3.3.7. Vues speciales	26

# 1. Introduction.

Ce document à pour but de vous accompagner à la maintenance de la solution XXII CORE REAL TIME en version V.2.X.X.

Version compatible avec ce guide:

XXII CORE REAL TIME CITY	V.2.0.0	V.2.0.1	V.2.1.0
XXII CORE REAL TIME INFRA	V.2.0.0	V.2.0.1	V.2.1.0
XXII CORE REAL TIME QSR	V.2.0.0	V.2.0.1	V.2.1.0

# 2. Le processor de debug.

## 2.1. Qu'est ce que le processor de debug?

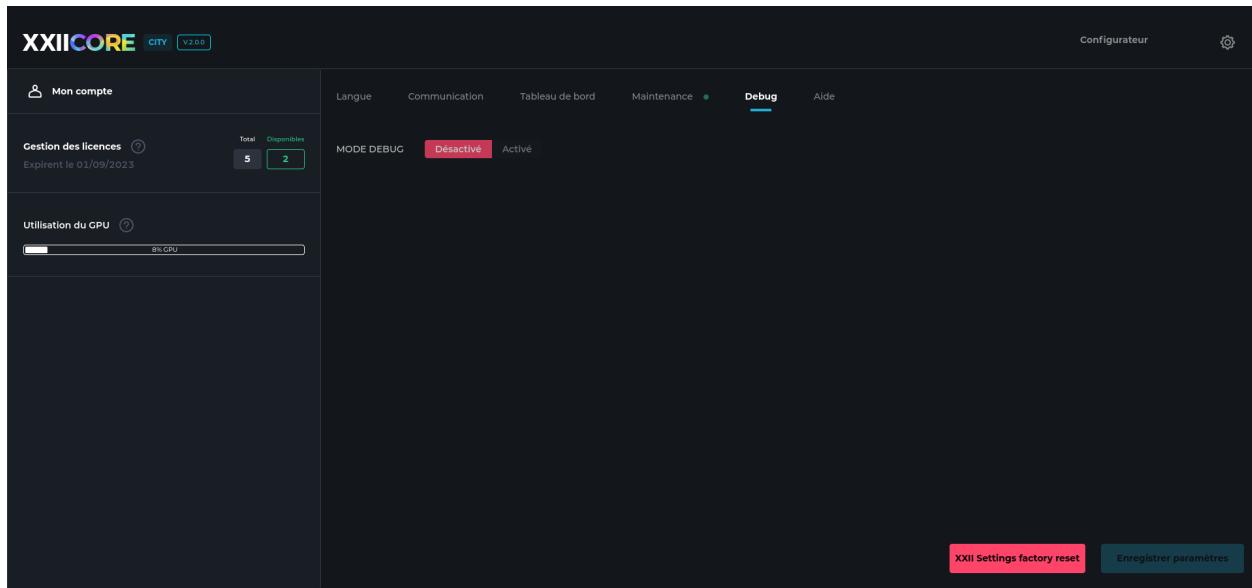
Le processor de debug est un composant interne à la plateforme permettant de les bounding box\* sur la / les vidéo(s) en live sur une page dédiée.

\* Elle permet de délimiter les objets reconnus par un rectangle sur l'image

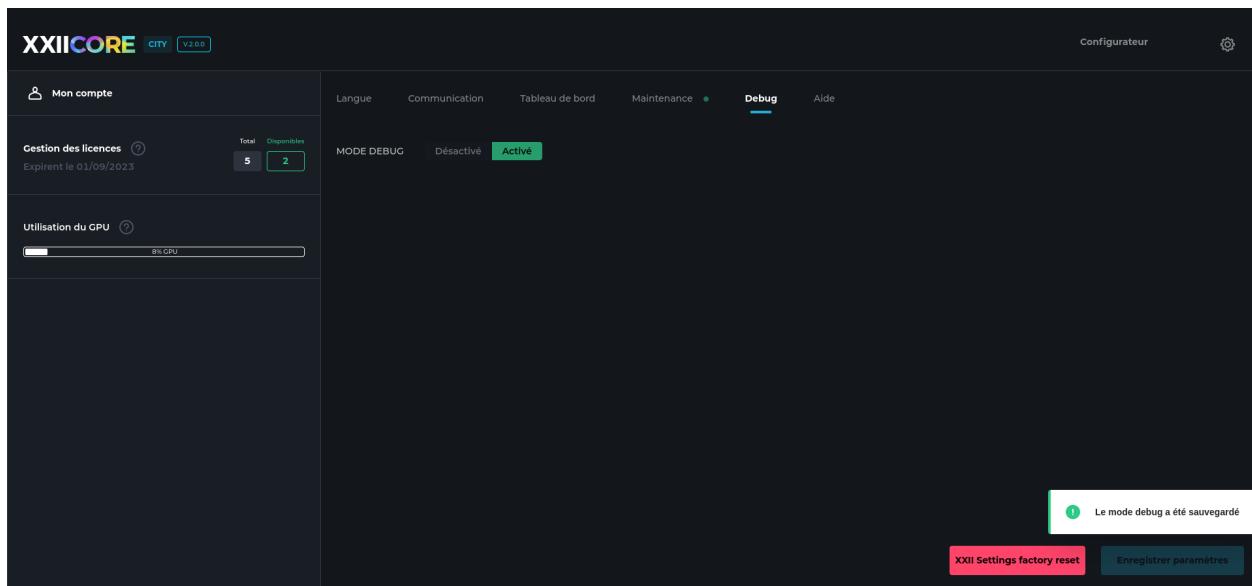
## 2.2. Activation du processor de debug.

Pour activer le processor de debug il faut dans un premier temps vous connecter à votre interface de configuration des compétences XXII en utilisant les liens suivants:

<b>SMART CITY</b>	<b>SMART INFRA</b>	<b>QSR</b>
<a href="http://smartcity.xxii-core.io/account/debug">http://smartcity.xxii-core.io/account/debug</a>	<a href="http://smartinfa.xxii-core.io/account/debug">http://smartinfa.xxii-core.io/account/debug</a>	<a href="http://qsr.xxii-core.io/account/debug">http://qsr.xxii-core.io/account/debug</a>



Une fois sur cette page, cliquer alors sur **Activé** puis sur **Enregistrer paramètres**.



Une fois que le processus de debug est activé, il est important de désactiver TOUTES les compétences actives sur la plateforme, puis réactiver la compétence souhaitée

Une fois ces manipulations effectuées vous pouvez alors vous rendre sur l'url dédiée au processus de debug :

<b>SMART CITY</b>	<b>SMART INFRA</b>	<b>QSR</b>
<a href="http://smartcity.debug.xxii-core.io/">http://smartcity.debug.xxii-core.io/</a>	<a href="http://smartinfra.debug.xxii-core.io/">http://smartinfra.debug.xxii-core.io/</a>	<a href="http://qsr.debug.xxii-core.io/">http://qsr.debug.xxii-core.io/</a>

#### Running tasks list

- 63b6e73149a677dd1f016967 DETECTION :  TRACKING :  RULES :  POST :  FIRE\_DETECTION :  default

Vous arrivez alors sur cette page, vous voyez qu'une compétence tourne, si vous cliquer sur le lien, vous pouvez alors voir l'algo tourner en temps réel.



Une fois que vous avez fini d'utiliser le processor de debug, il est important de désactiver le processor de debug et d'activer et désactiver à nouveau la compétence.

## 3. Logiciel d'orchestration.

### **3.1. Connexion en ssh à la machine**

#### 3.1.1 Depuis un PC Windows

Pour se connecter en ssh depuis un PC sous Windows, ouvrez un terminal, puis utiliser la commande ssh suivie de l'adresse IP ou du nom d'hôte de l'ordinateur auquel vous voulez vous connecter, ainsi que votre nom d'utilisateur et votre mot de passe.

Par exemple:

```
ssh username@hostname
```

Vous serez alors invité à entrer votre mot de passe. Une fois connecté, vous serez en mesure de naviguer et de travailler sur l'ordinateur distant comme s'il s'agissait de votre propre ordinateur.

### 3.1.2 Depuis un PC Linux

Pour se connecter en ssh depuis un PC sous Linux, ouvrez un terminal, puis utiliser la commande ssh suivie de l'adresse IP ou du nom d'hôte de l'ordinateur auquel vous voulez vous connecter, ainsi que votre nom d'utilisateur et votre mot de passe.

Par exemple:

```
ssh username@hostname
```

Vous serez alors invité à entrer votre mot de passe. Une fois connecté, vous serez en mesure de naviguer et de travailler sur l'ordinateur distant comme s'il s'agissait de votre propre ordinateur.

### 3.1.3 Depuis un PC MAC

Pour se connecter en ssh depuis un PC sous MAC, ouvrez un terminal, puis utiliser la commande ssh suivie de l'adresse IP ou du nom d'hôte de l'ordinateur auquel vous voulez vous connecter, ainsi que votre nom d'utilisateur et votre mot de passe.

Par exemple:

```
ssh username@hostname
```

Vous serez alors invité à entrer votre mot de passe. Une fois connecté, vous serez en mesure de naviguer et de travailler sur l'ordinateur distant comme s'il s'agissait de votre propre ordinateur.

## 3.2. Utilisation de Grafana

### 3.2.1. Introduction

Grafana est un logiciel utilisé pour afficher les métriques qui sont stockées par d'autres logiciels. Il permet d'avoir une vue globale d'un cluster, d'un nœud, d'un pod.

Les explications suivantes sont valable pour *XXII CORE REAL TIME V.2.X.X*.

Cette page n'explique pas en détail comment créer un tableau de bord, mais vous donne astuces pour vous aider.

### 3.2.2 Accéder à Grafana

Sur un de nos cluster, vous pouvez vous connecter à grafana avec le lien suivant (en adaptant votre fqdn) et en utilisant les identifiants suivant.

<b>SMART CITY</b>	<b>SMART INFRA</b>	<b>QSR</b>
<a href="http://smartcity.grafana.xxii-core.io">http://smartcity.grafana.xxii-core.io</a>	<a href="http://smartinfra.grafana.xxii-core.io">http://smartinfra.grafana.xxii-core.io</a>	<a href="http://qsr.grafana.xxii-core.io">http://qsr.grafana.xxii-core.io</a>
Identifiant:	admin	
Mot de passe:	prom-operator	

### 3.2.3. Liste des sources par défaut et celles de XXII

Dans notre solution nous utilisons actuellement:

- Prometheus (solution pour stocker les métriques et calculer les alertes).
- Loki (un scrapper de log)

#### 3.2.3.1. Prometheus

Prometheus est utilisé comme une base de données qui va stocker les métriques calculées par les applications déployées dans notre système. Il peut s'agir d'un composant spécifique (appelé "exportateur"), qui récupère des données et les stocke dans Prometheus pour permettre de créer un tableau de bord Grafana (Nvidia GPU, HiveMQ, Mongodb, Redis, ...), ou d'un composant interne qui stocke les données dans Prometheus.

Actuellement, les *Processors* et *MediaManager* stockent des données sur les images par seconde et la latence pour générer un tableau de bord.

Pour voir ce que prometheus stocke, cliquer sur les liens suivants:

<b>SMARTCITY</b>	<b>SMARTINFRA</b>
<a href="http://smartcity.prometheus.xxii-core.io/graph">http://smartcity.prometheus.xxii-core.io/graph</a>	<a href="http://smartinfra.prometheus.xxii-core.io/graph">http://smartinfra.prometheus.xxii-core.io/graph</a>

Sur cette page, saisissez des requêtes pour avoir accès à des informations attendu.

Ici nous allons écrire « container\_cpu\_user\_seconds\_total » pour avoir des informations sur le CPU. Cliquer sur "Execute" pour afficher les informations souhaitées.

Enable query history

Try experimental React [UI](#)

container\_cpu\_user\_seconds\_total

**Execute** - insert metric at cursor -

**Graph** Console

< Moment >

**Element**

```
container_cpu_user_seconds_total{endpoint="https://metrics"} id="r" instance="192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" node="node1" service="monitoring-prometheus-oper-kubelet"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="d" docker=20e46486ec0f0989817fc380d5bedde143679a25321097 refc2b692690" image="netdata/netdata" instance="192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" name="netdata" node="node1" service="monitoring-prometheus-oper-kubelet"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="k" kube pods="instance=192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" node="node1" service="monitoring-prometheus-oper-kubelet"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="b" kube pods/besteffort="instance=192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" node="node1" service="monitoring-prometheus-oper-kubelet"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="p" kube pods/besteffort/pod1b08fc4a-4ad6-44c5-9841-d45211dc4ac" instance="192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" namespace="monitoring" node="node1" pod="monitoring-prometheus-adapt-dd6d6fd79-7w4z7" service="monitoring-prometheus-oper-kubelet"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="s" kube pods/besteffort/pod24dbbc1c-34b6-47de-aee5-52d11101a6fe" instance="192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" namespace="monitoring" node="node1" pod="monitoring-prometheus-operator-kube-state-metrics-adapter-59b5bbdd-mvw88" service="monitoring-prometheus-oper-kubelet"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="m" kube pods/besteffort/pod48f66366-dt7e-4333-8d6c-6218919cd244" instance="192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" namespace="monitoring" node="node1" pod="monitoring-metrics-server-59b5bbdd-mvw88" service="monitoring-metrics-server"
container_cpu_user_seconds_total{endpoint="https://metrics"} id="g" kube pods/besteffort/pod4e266341-a351-4ac8-9a5d-bd1a1e36ddfe" instance="192.168.30.31:10250" job="kubeflet" metrics_path="/metrics/cadvisor" namespace="kube-system" node="node1" pod="kubernetes-metrics-scraper-54fbfd495-kppbg" service="kubernetes-metrics-scraper"
```

Vous pouvez voir quelques informations de la requête.

Maintenant, vous pouvez afficher uniquement les informations sur l'espace de nom "default":

```
container_cpu_user_seconds_total{namespace="default"}
```

Pour le nom du pod contenant la plateforme:

container\_cpu\_user\_seconds\_total{namespace="default", pod=~"platform.\*"}

En voyant les données, vous pouvez trouver quelles sont les clés disponibles pour la requête et adapter la regex:

```
container_cpu_user_seconds_total{container="POD",
endpoint="https-metrics",
id="/kubepods/burstable/pod0700a190-aa37-43a3-93ca-
4c616e613adc/9a0d365da2ddb0a2ef07e92b04f0e3d1ace7fd18042de5e1c429afe9b0f94e86"
,
image="k8s.gcr.io/pause:3.2",
instance="192.168.30.31:10250",
job="kubelet",
metrics_path="/metrics/cadvisor",
name="k8s_POD_platform-web-gateway-5ff9b97c4d-sffdb_default_0700a190-aa37-43a3-
93ca-4c616e613adc_0",
namespace="default",
node="node1",
pod="platform-web-gateway-5ff9b97c4d-sffdb",
service="monitoring-prometheus-oper-kubelet"}
```

RE2 est utilisé pour les regex.

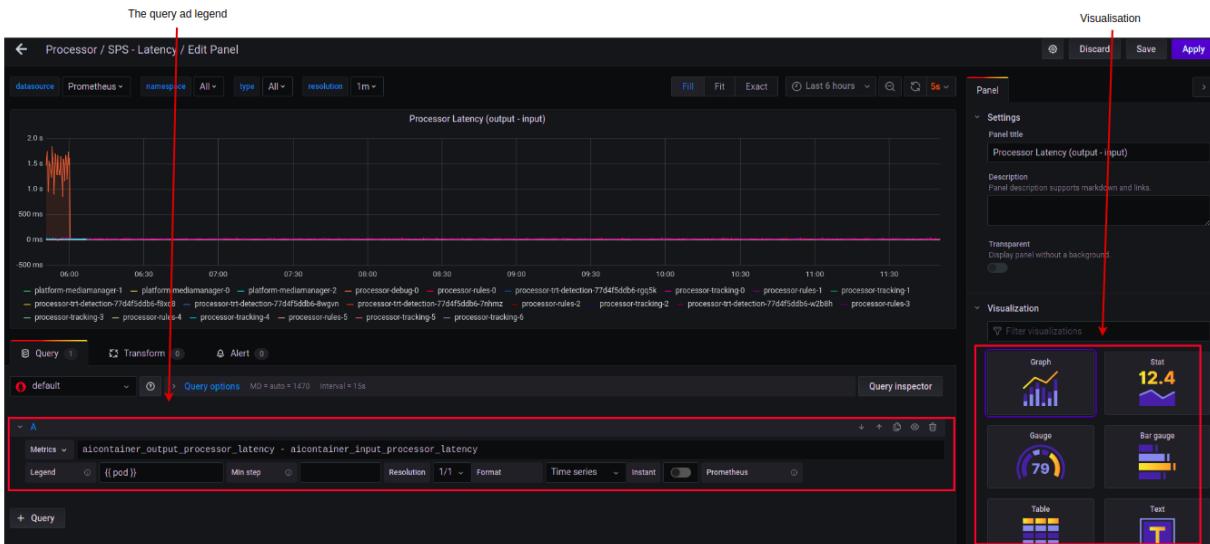
Sur chaque champ, vous pouvez effectuer une opération ou une fonction (taux, somme, ...) Pour plus d'informations, cliquez ici :

- <https://prometheus.io/docs/prometheus/latest/querying/basics/>
  - <https://prometheus.io/docs/prometheus/latest/querying/examples/>
  - <https://prometheus.io/docs/prometheus/latest/querying/functions/>

### 3.2.3.2. Dashboard

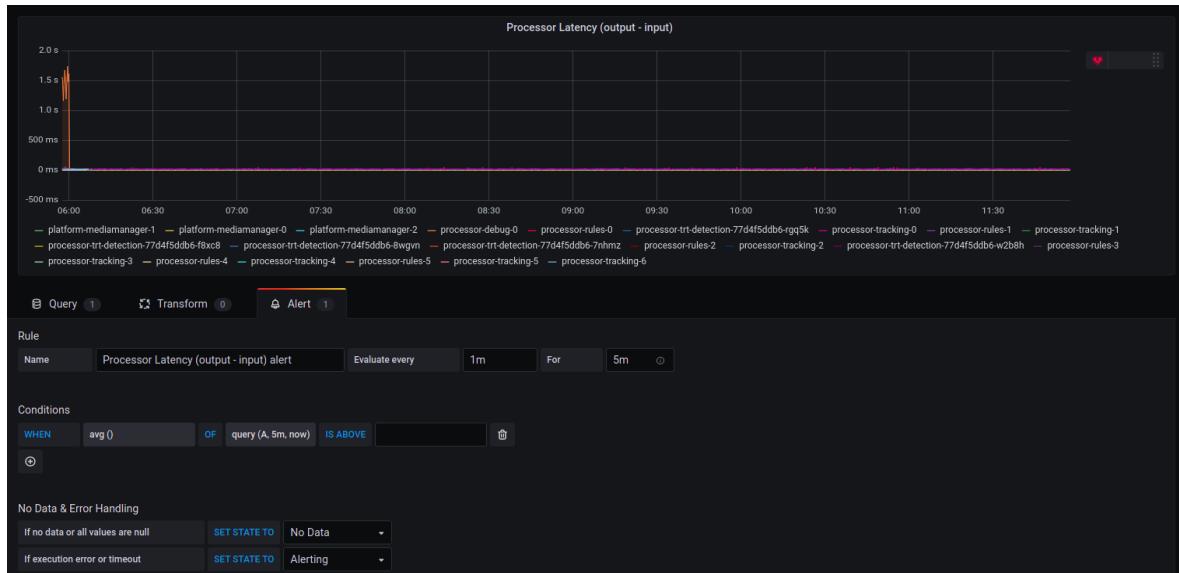


- Voir les panneaux affichés
  - Possibilité de sélectionner un élément en cliquant dessus dans la légende du panneau
  - Possibilité de sélectionner plusieurs éléments en utilisant CTRL
  - Possibilité de supprimer un élément en utilisant la touche CTRL si tous les éléments sont actuellement sélectionnés.
- Sélectionnez l'intervalle de temps
- Modifier les variables pour afficher ce qui vous intéresse
  - Ou ajouter des variables  
<https://grafana.com/docs/grafana/v9.0/variables/variable-types/global-variables/>
- Modifier un panneau (sera réinitialisé à chaque F5, sauf si vous le sauvegardez, le mettez dans GIT et déployez une nouvelle version)
- Ajouter un panneau
- Modifier les paramètres (changer les noms, ajouter des variables, ...)
- Modifier un panneau (en cliquant sur son nom et ensuite sur modifier)



- Vous pouvez faire tous les tests que vous voulez, les appliquer car aucune modification ne sera sauvegardée sauf si un autre déploiement est effectué.

### 3.2.3.4. Alertes



Les alertes peuvent être définies dans un panneau

Vous pouvez définir une alerte, basée sur une requête. Ainsi, si vous souhaitez créer des alertes spécifiques, vous devez d'abord créer le panneau, puis ajouter les alertes.

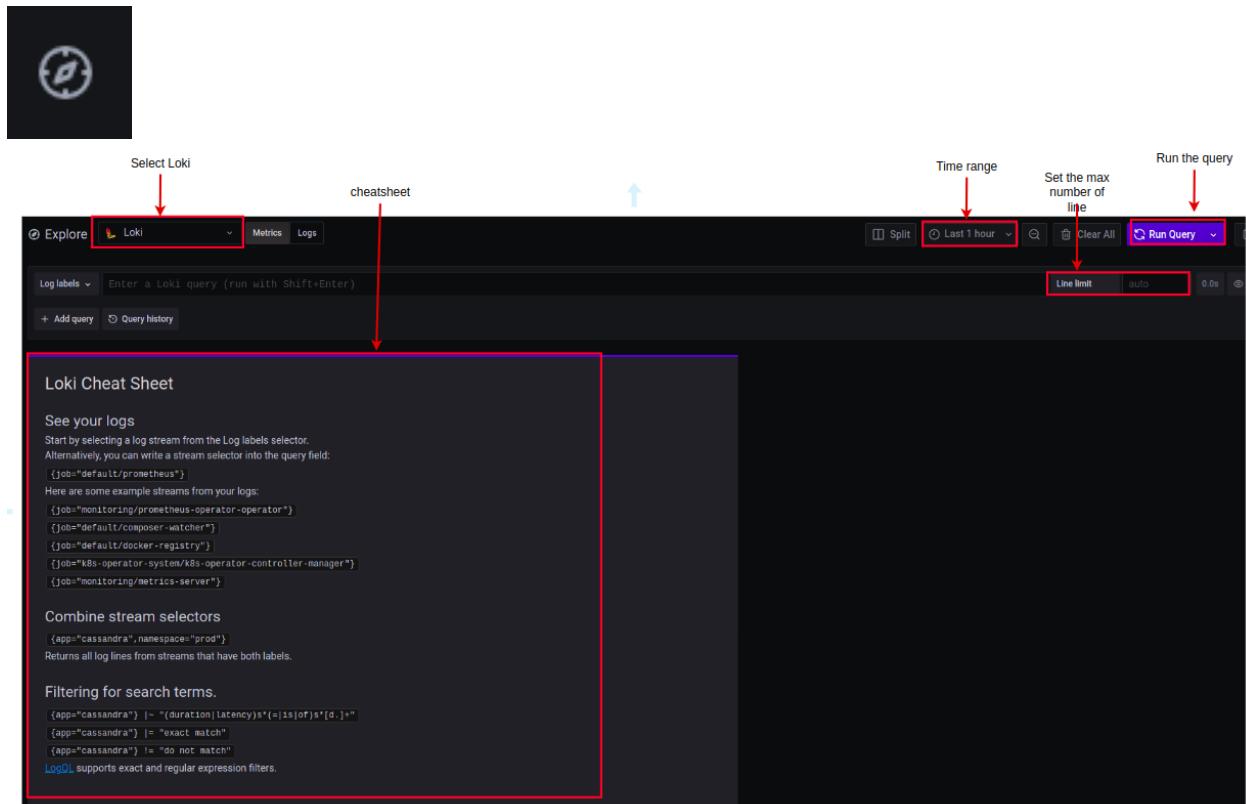
Certaines limitations existent, car vous pouvez utiliser des variables si vous voulez créer une alerte, donc parfois, vous devrez créer un panneau pour l'affichage, et un autre spécifiquement pour l'alerte.

Les alertes peuvent être :

- Affichées dans un panneau sous forme de liste (solution proposée)

### 3.2.3.5 Loki

Vous pouvez accéder à la vue de l'explorateur pour consulter les journaux en cliquant sur le logo de l'explorateur.

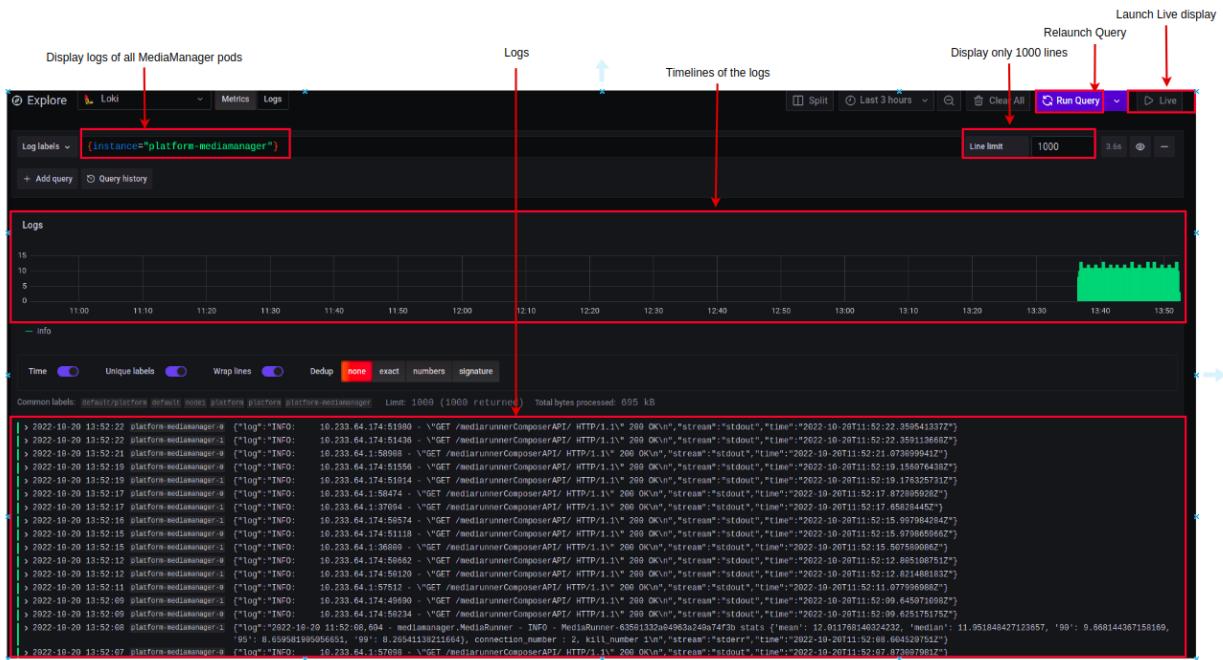


Vous pouvez sélectionner les journaux que vous voulez voir en cliquant sur "Log Labels".

- un pod : pod=xxx
- un service (groupe de pods en même temps): instance=xxx

Seul un nombre maximum de lignes sera affiché (basé sur les valeurs max, ou vos paramètres)

Pour le MediaManager:



Ici vous pouvez voir 1000 lignes, à cause du maximum réglé.

Un mode live existe pour afficher les nouveaux logs quand ils se produisent.

Vous pouvez zoomer sur une période spécifique en zoomant dans la ligne de temps.

Chaque log contient des métadonnées, vous pouvez cliquer dessus pour les afficher.

L'horodatage affiché n'est pas l'horodatage des logs mais l'horodatage du scrap. Si vous voulez voir l'horodatage du log, vous devez regarder dans le message directement.

Si vous voulez voir seulement la ligne avec le mot clé "stats" :

```
{job=~"default/platform"} |= "stats" | logfmt
```

Il est également possible de créer un tableau de bord affichant l'enregistrement dans le temps.

```
sum(count_over_time({job=~"default.+"}[5m])) by (instance)
```

## 3.2.4. Les tableaux

### 3.2.4.1. Nodes

Ce tableau de bord permet d'avoir une vision globale d'un serveur sur les éléments suivants :

- CPU
- RAM
- Disques
- Réseau



### 3.2.4.2. Node Exporter Full

Pour ajouter : <https://grafana.com/grafana/dashboards/1860-node-exporter-full/>

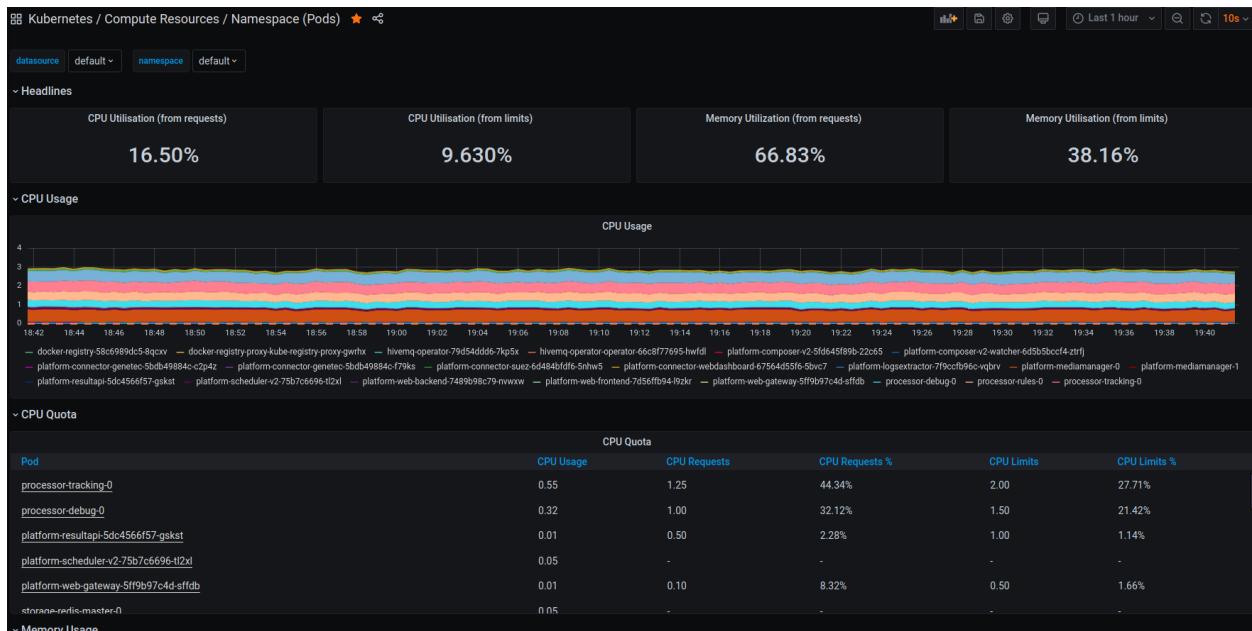
Afficher approximativement toutes les informations sur le serveur



### 3.2.4.3. Kubernetes / Compute Resources / Namespace (Pods)

Permet d'afficher des informations pour un espace de nom donné, et donc de sélectionner un ou plusieurs pods à afficher :

- CPU
- RAM
- Réseau

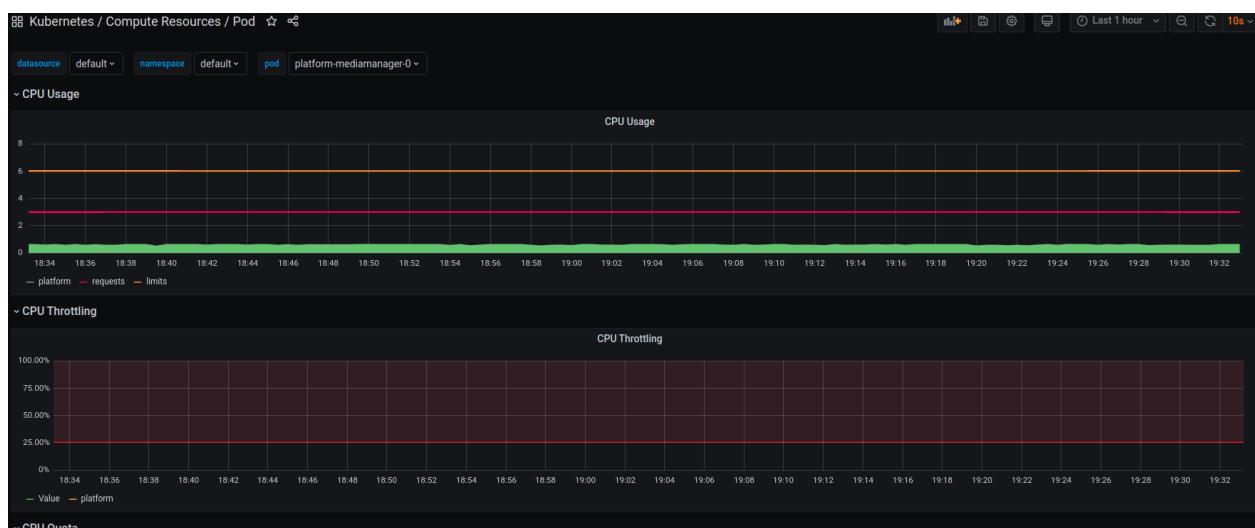


### 3.2.4.4. Kubernetes / Compute Resources / Pod

Permet de sélectionner un espace de nom et un pod.

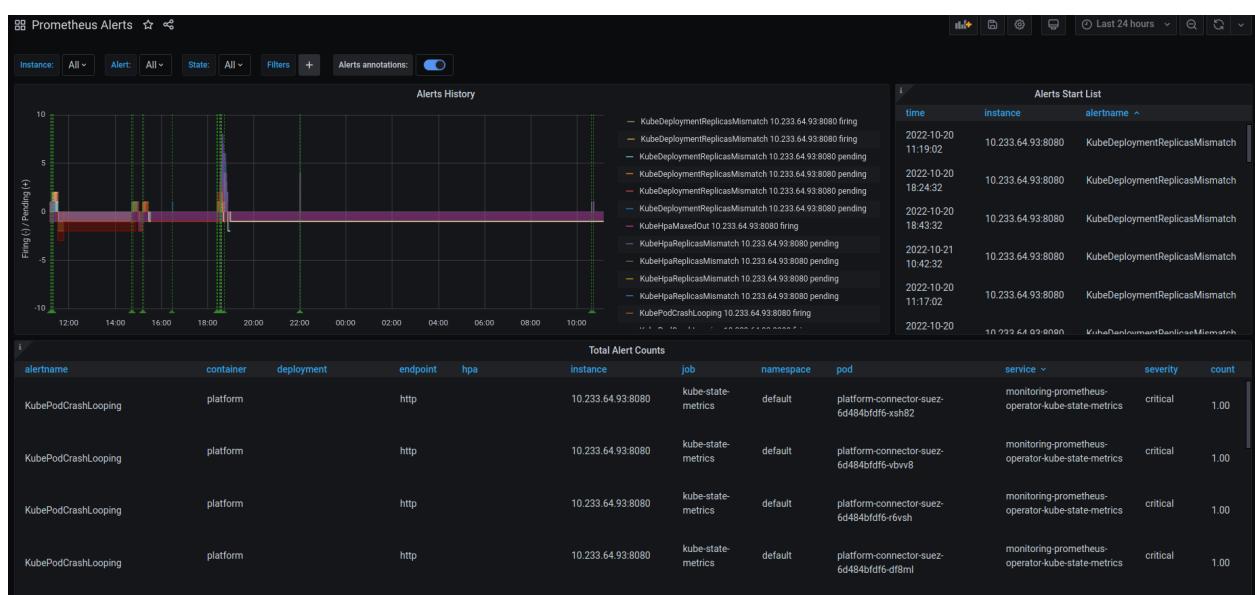
Vous pouvez voir :

- Consommation du CPU (seuils de demande et de limite)
- Restriction du CPU (si la restriction a lieu, la demande donnée n'est pas suffisante).
- Consommation de RAM (seuils de demande et de limite)
- Réseau



### 3.2.4.5. Prometheus Alerts

Afficher les alertes déclenchées dans prometheus pendant la période définie.



## 3.3. K9S

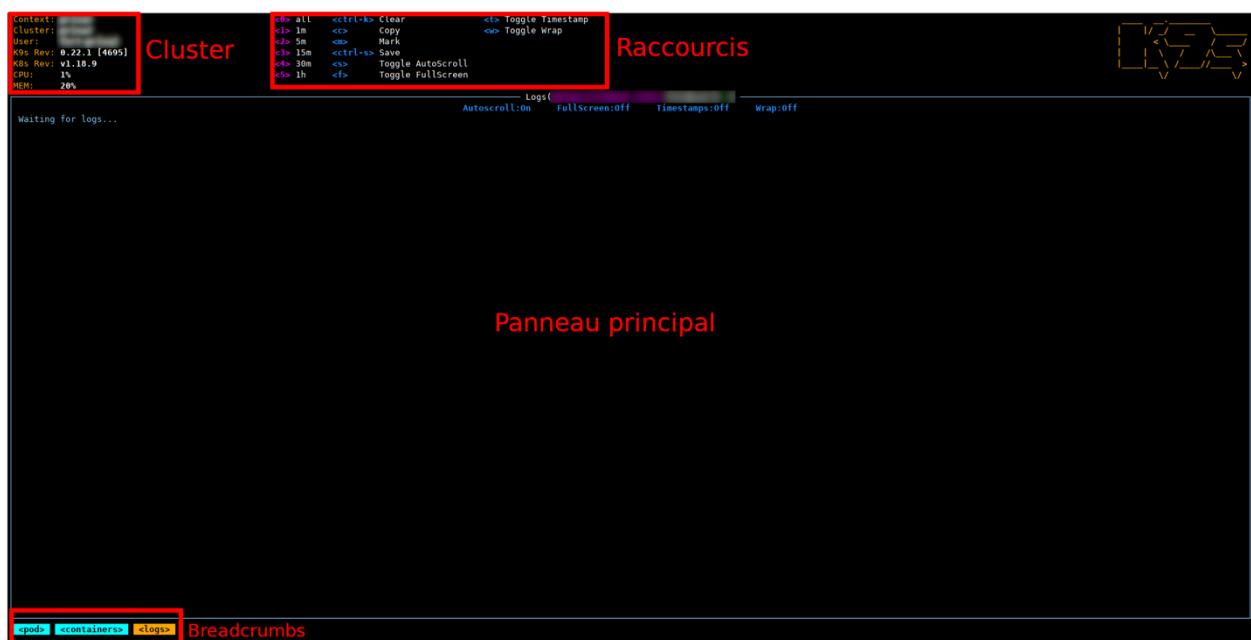
### 3.3.1. Introduction à K9S

K9S est un projet écrit en Go et publié en tant que binaire exécutable.

Pour interagir avec un cluster, k9s lit les informations du fichier `~/.kube/config`. Il vous suffit de lancer k9s une fois connecté en [ssh](#) depuis votre terminal et ce dernier utilisera automatiquement votre contexte Kubernetes courant.

### 3.3.2. Présentation de l'interface

Voici maintenant un aperçu de l'interface de k9s qui en résume les principales zones d'intérêt :



- En haut à gauche, quelques informations globales données par k9s telles que le nom du cluster auquel vous êtes connecté, le contexte actuellement utilisé, votre userid, votre version de k9s, etc.
- À droite, vous trouverez une liste de raccourcis utiles pour interagir avec k9s. Gardez dans un coin de votre tête que son contenu change en fonction de la vue active, nous y reviendrons plus tard. Notez aussi que la touche ? ouvrira l'aide complète à tout moment.
- Au milieu, on retrouve le panneau principal, c'est ici que se trouveront toutes les informations sur nos ressources. Son contenu changera lui aussi en fonction de la vue active et le nom de cette dernière vous sera rappelé dans le titre du panneau principal.

- Enfin tout en bas à gauche, un *fil d'Ariane* (ou *breadcrumbs*) nous donne la hiérarchie de la vue active et indique vers quelle vue nous retournerons lorsqu'on la quitte.

D'une manière générale, k9s reprend plusieurs principes de vim. La touche esc sert à revenir en arrière ou à annuler / fermer un prompt et on quitte le programme avec :q (ou ctrl + c).

Lorsque vous serez parfaitement à l'aise avec k9s, vous pourrez masquer le panneau principal en pressant `ctrl + e` afin d'économiser un peu de place, mais pour l'instant je vous invite à le laisser affiché.

### 3.3.3. Les différents modes

### 3.3.3.1. Le mode commande

La touche : ouvre un prompt chien ( ) au-dessus du panneau principal. C'est le mode de k9s qui permet d'exécuter des commandes. Il existe une commande par ressource kubernetes plus quelques autres spécifiques que nous verrons plus tard. Par exemple, :pod<sup>d</sup> bascule dans la vue Pod et ce sont désormais les informations sur les pods qui sont affichées dans le panneau principal.

Context: xxii-cluster	<ctrl> all	<ctrl> Attach	<ctrl> Logs	<ctrl> YARN									
Cluster: xxii-cluster	<ctrl> default	<ctrl> Delete	<ctrl> Logs Previous	<ctrl> Port-Forward									
User: admin	<ctrl> Describe	<ctrl> diff-if-r	<ctrl> Shell										
K9s Rev: v0.26.7	<ctrl> Edit	<ctrl> S											
K8s Rev: v1.23.12+k0s	<ctrl> Help	<ctrl> Show Node											
CRDs: 13%	<ctrl> k	<ctrl> Show PortForward											
MEM: 13%													
Pods (default) [18]													
NAME	PF	READY	RESTARTS	STATUS	CPU	MEM	%CPU/R	%CPU/L	%MEM/R	%MEM/L	IP	NODE	AGE
l1vem-operator-5f69cc8dddf-vlg6n	*	1/1	0	Running	6	4109	3	0	100	16	10.24.82.196	core-dev	13d
l1vem-operator-5f69cc8dddf-wx9cc		1/1	0	Running	1	2774	0	0	50	10	10.24.82.195	core-dev	13d
nvidia-smi	*	1/1	0	Running	1	2	n/a	n/a	n/a	10	10.24.82.126	core-dev	13d
platform-composer-v2-9db5b-mwj1	*	1/1	0	Running	3	108	n/a	n/a	n/a	10	10.24.82.78	core-dev	7d6h
platform-composer-v2-watcher-c8774747b-shv4p	*	1/1	0	Running	37	104	n/a	n/a	n/a	10	10.24.82.127	core-dev	7d6h
platform-connect-generics-c-0555f58749-9rl6b	*	1/1	0	Running	2	36	1	0	7	3	10.24.82.125	core-dev	22m
platform-connector-head-af5f5f58749-9rl6b	*	1/1	2	Running	2	15	1	0	5	4	10.24.82.85	core-dev	7d6h
platform-loggerator-7f76594908-2vh8	*	1/1	0	Running	1	37	0	0	7	3	10.24.82.120	core-dev	13d
platform-mediamanager-0	*	1/1	0	Running	3	154	0	0	6	3	10.24.82.115	core-dev	13d
platform-resultapi-f7cb576c-mnp7h	*	1/1	0	Running	2	58	0	0	2	1	10.24.82.111	core-dev	7d1h
platform-scheduler-0fb1bf888-rt7j3	*	1/1	0	Running	6	179	n/a	n/a	n/a	10	10.24.82.108	core-dev	13d
platform-web-settings-backend-6f806b5b96-6kgtx	*	1/1	0	Running	4	122	1	0	47	23	10.24.82.92	core-dev	7d6h
platform-web-settings-frontend-75f8fd9795-b98wq	*	1/1	0	Running	1	57	1	0	22	11	10.24.82.96	core-dev	7d6h
platform-web-settings-gateway-5584dd99b-9r4sq	*	1/1	0	Running	4	63	4	0	24	12	10.24.82.118	core-dev	7d6h
pxzreader-xx1api-179fd5c79-hg49h	*	1/1	11	Running	1	39	0	0	15	7	10.24.82.121	core-dev	13d
storage-0f99f-25kg	*	1/1	0	Running	1	57	n/a	n/a	n/a	10	10.24.82.103	core-dev	13d
storage-mongod-0	*	2/2	0	Running	19	776	n/a	n/a	n/a	10	10.24.82.101	core-dev	13d
storage-redis-master-0	*	2/2	0	Running	6	81	n/a	n/a	n/a	10	10.24.82.97	core-dev	3d7h

k9s fait de l'autosuggestion pour le nom des commandes. Quand je tape pod, il me propose la commande poddisruptionbudget.

- Les touches  et  permettent de naviguer circulairement entre les différentes propositions
  - Les touches , tab ou ctrl + f permettent de compléter automatiquement la suggestion dans le prompt (autocomplete), sans toutefois exécuter la commande

- Les touches **ctrl + u** ou **ctrl + w** permettent d'effacer la ligne sans quitter le mode commande

La liste complète des commandes est disponible par le raccourci **ctrl + a** ou avec **:aliases**.

RESOURCE	COMMAND	Aliases( <b>alt+1/2</b> )	APIGROUP
alertmanagers	alertmanager.alertmanagers.monitoring.coreos.com/v1/alertmanagers	a,alias,aliases	
aliases	apiregistration.k8s.io/v1/apiservices	a,alias,aliases	
apiservices	apiregistration.k8s.io/v1/apiservices.apiservice.apiservices	api,apm,k8s,elastic.co	apiregistration.k8s.io
apmserver	apiregistration.k8s.io/v1/apiservers.apmserver.apmservers	apm,k8s,elastic.co	apiregistration.k8s.io
beats	beats.v1.coreos.com/v1/beats,beats	be,benchmark,benchmarks	beat.k8s.elastic.co
benchmarks	beats.v1.coreos.com/v1/beats,beats	be,benchmark,benchmarks	beat.k8s.elastic.co
bindings	binding.bindings.v1/bindings		
certificatesigningrequests	certificates.k8s.io/v1beta1/certificatesigningrequests,certificatesigningrequest,certificatesigningrequests,csr		
ciliumnetworkpolicies	cilium.io/v2/ciliumnetworkpolicies	cnp,cilium,io,ciliumnetworkpolicies,ciliumclusterwidernetworkepolicy	cilium.io
ciliumendpoints	cilium.io/v2/ciliumendpoints,ciliumendpoints,ciliumendpoints,ciliummp	cilium,io	cilium.io
ciliumidentities	cilium.io/v2/ciliumidentities,ciliumid,ciliumidentities,ciliumidentity	cilium,io	cilium.io
ciliumnetworkpolicies	cilium.io/v2/ciliumnetworkpolicies,ciliumnetworkpolicies,ciliumnetworkpolicy,ciliummp,cnp	cilium,io	cilium.io
ciliumnodes	cilium.io/v2/ciliumnodes,cilium,ciliumnode,ciliumnodes,cn	cilium,io	cilium.io
ciliumrolebindings	cilium.io/v2/ciliumrolebindings,cr,rbac.authorization.k8s.io/v1/ciliumrolebindings	cilium,io	rbac.authorization.k8s.io
clusterroles	clusterrole,clusterroles,cr,rbac.authorization.k8s.io/v1/clusterroles	clusterrole,clusterroles,cr,rbac.authorization.k8s.io/v1/clusterroles	rbac.authorization.k8s.io
clusterworkflowntemplates	argoproj.io/v1alpha1/clusterworkflowntemplates,clusterwfmp,clusterworkflowtemplate,clusterworkflowtemplates,cwft	argoproj.io/v1alpha1/clusterworkflowntemplates,clusterwfmp,clusterworkflowtemplate,clusterworkflowtemplates,cwft	argoproj.io
componentstatuses	componentstatus,componentstatuses,cs,v1/componentstatuses	cm,configmap,configmaps,v1/configmaps	
configmaps	cm,configmap,configmaps,v1/configmaps	cm,configmap,configmaps,v1/configmaps	
controllerrevisions	apps/v1/controllerrevisions,controllerrevisions,controllerrevisions		
cronjobs	batch/v1beta1/cronjobs,cj,cronjob,cronjobs		
cronflows	batch/v1beta1/cronflows,cronmf,cronflow,cronflows,cwf		
controllerrevisions	batch/v1beta1/cronjobs,cj,cronjob,cronjobs		
cronflows	batch/v1beta1/cronflows,cronmf,cronflow,cronflows,cwf		
controllerrevisions	batch/v1beta1/cronjobs,cj,cronjob,cronjobs		
cronflows	batch/v1beta1/cronflows,cronmf,cronflow,cronflows,cwf		

On notera que toutes les commandes disposent d'un alias plus court qui permet d'accélérer la navigation (e.g. **:a** au lieu de **:aliases**)

Si vous entrez une commande qui n'existe pas k9s vous le fera savoir avec une cowsay qui malheureusement ne partira pas avec la touche esc. Pour sortir de cette vue, il faut rejouer une commande valide comme **:pod ↵** par exemple.

NAME	PF	READY	RESTARTS	STATUS	CPU	MEM	%CPU/R	%CPU/L	%MEM/R	%MEM/L	IP	NODE	AGE
hive-mq-operator-5f60c8ddd-f-v1g6n	●	1/1	0	Running	0	4100	0	0	100	15	10.244.82.106	core-dev	13d
hive-mq-operator-operator-7f9805b50c-8x9cc	●	1/1	0	Running	1	274	0	0	50	50	10.244.82.104	core-dev	14d
nvidia-smi	●	1/1	0	Running	1	2	n/a	n/a	n/a	n/a	10.244.82.126	core-dev	13d
platform-composer-v2-9db5db-mw1ql	●	1/1	0	Running	4	100	n/a	n/a	n/a	n/a	10.244.82.78	core-dev	7d7h
platform-composer-v2-web-546457c97-7b-sbv4p	●	1/1	0	Running	28	104	n/a	n/a	n/a	n/a	10.244.82.127	core-dev	7d7h
platform-composer-web-546457c96-80n7q	●	1/1	0	Running	2	30	1	0	7	7	10.244.82.128	core-dev	24h
platform-connector-webdashboard-6f55f587d4-9rl0b	●	1/1	2	Running	1	45	0	0	8	4	10.244.82.85	core-dev	7d7h
platform-logger-lopsextractor-767859498-2vh8z	●	1/1	0	Running	1	37	0	0	7	3	10.244.82.129	core-dev	13d
platform-mediamanager-0	●	1/1	0	Running	3	154	0	0	6	3	10.244.82.115	core-dev	13d
platform-resultapi-f7cd5705c-nmp7h	●	1/1	0	Running	3	58	0	0	2	1	10.244.82.111	core-dev	7d1h
platform-scheduler-f2-fbf079688-ewjx	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.82.84	core-dev	13d
platform-web-settings-frontend-75fdaf9795-b98w8	●	1/1	0	Running	4	7	23	0	92	23	10.244.82.92	core-dev	7d8h
platform-web-settings-gateway-5d84ddc9b-9r48w	●	1/1	0	Running	22	11	10.244.82.116	core-dev	7d6h				
ptzreader-xxiiapi-679fd95c79-sgh49	●	1/1	0	Running	24	12	10.244.82.118	core-dev	7d6h				
server-nginx-dff9779f-z50kg	●	1/1	0	Running	15	7	10.244.82.121	core-dev	13d				
storage-mongod-0	●	2/2	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.82.103	core-dev	14d
storage-redis-master-0	●	2/2	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.82.97	core-dev	3d7h

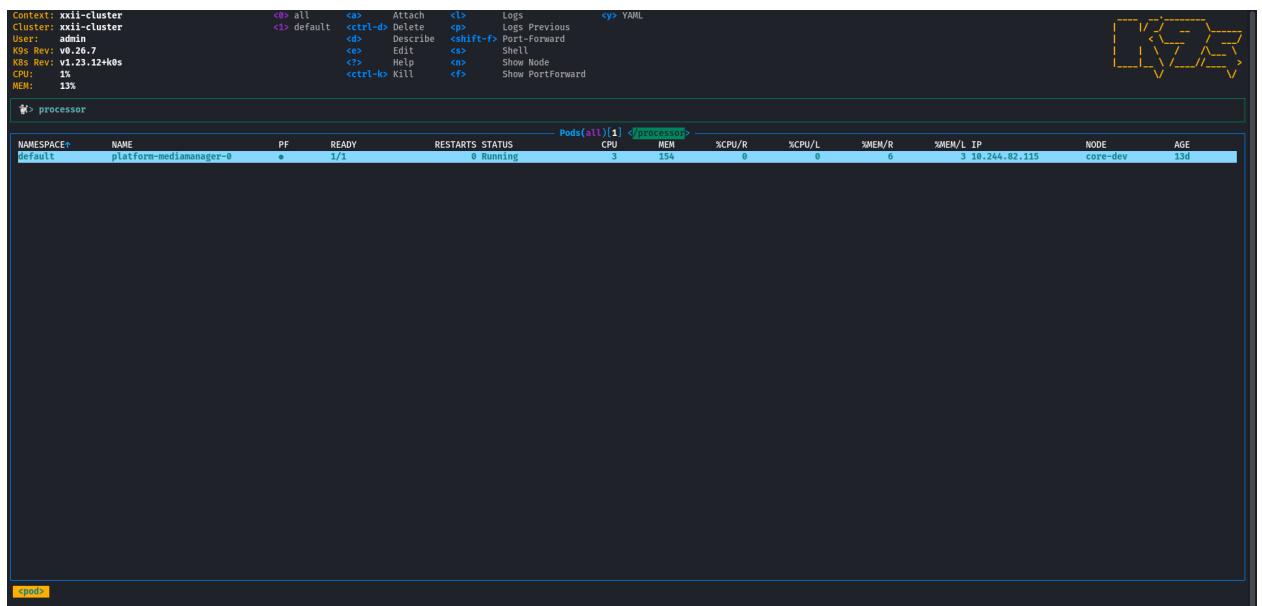
< Ruroh? "xxii" command not found >

Dismiss

### 3.3.3.2. Le mode filtre

La touche / ouvre un prompt caniche 🦴 qui permet d'appliquer un filtre sur la liste d'éléments affichés dans le panneau. C'est le mode filtre de k9s.

En reprenant notre exemple, si l'on est dans la vue pod (:pod) et que l'on tape /processor➡ on n'affichera que les *Pods* qui contiennent processor dans leur nom.



The screenshot shows the k9s application interface. At the top, there is a command-line style header with various keyboard shortcuts for actions like 'all', 'Attach', 'Logs', 'YAML', etc. Below this, a search bar contains the text '/processor'. The main area displays a table titled 'Pods(all)[1]' with a single row. The row shows a pod named 'platform-mediamanager-0' in the 'Default' namespace, which is 'READY' and has '0' restarts. The table includes columns for 'NAMESPACE', 'NAME', 'PF', 'READY', 'RESTARTS', 'STATUS', 'CPU', 'MEM', '%CPU/R', '%CPU/L', '%MEM/R', '%MEM/L', 'IP', 'NODE', and 'AGE'. The 'processor' filter is highlighted in blue in the table header. The bottom left corner of the interface has a small yellow button labeled 'pod'.

On peut aussi filtrer les ressources qui portent un label kubernetes avec la syntaxe /-l <nom\_du\_label>=<valeur\_du\_label>. Dans notre exemple /-l k8s-app=xxii liste tous les pods qui portent le label k8s-app=xxii (notez bien qu'on filtre uniquement sur l'existence de label ici et pas sur le nom des pods.)

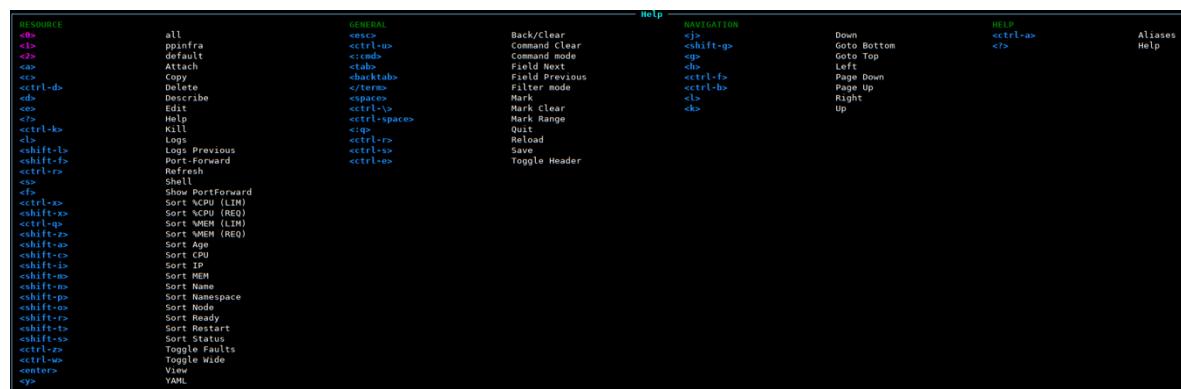
Ce prompt fonctionne dans *toutes* les vues de k9s et applique le filtre sur tout ce qui est affiché dans le panneau principal. Il n'est pas réservé aux vues des ressources Kubernetes. Voyez un exemple dans la vue aliases.

### 3.3.3.3. Le panneau des raccourcis

Intéressons-nous de plus près au panneau des raccourcis. Il contient tout ce qu'il vous faut pour vos opérations courantes. Rappelez-vous que son contenu change en fonction de la vue dans laquelle vous êtes, alors pensez à vous y référer régulièrement . Voici par exemple le contenu des raccourcis pour la vue pod :

```
<0> all      <a> Attach      <l> Logs
<1> ppinfra  <ctrl-d> Delete     <shift-l> Logs Previous
<2> default   <d> Describe    <shift-f> Port-Forward
                <e> Edit        <s> Shell
                <?> Help        <f> Show PortForward
                <ctrl-k> Kill       <y> YAML
```

Vous pouvez à tout moment appuyer sur la touche ? pour afficher l'aide complète.



Globalement, un raccourci s'applique à la ligne du panneau principal en surbrillance et vous pouvez changer de ligne en utilisant les flèches de votre clavier.

NAMESPACE↑	NAME
kube-system	cilium-operator
kube-system	cluster-autoscaler
kube-system	coredns
kube-system	coredns-autoscaler
kube-system	dns-controller
monitoring	metrics-server
monitoring	blackboxexport
monitoring	cloudwatch-exporter
monitoring	grafana
monitoring	kube-state-metrics
monitoring	prometheus-operator
monitoring	thanos-external
monitoring	thanos-querier
observability	jaeger-collector
observability	jaeger-operator
observability	jaeger-query
observability	jaeger-storage
ppango	argo-server

Les touches numériques affichées en magenta dans le panneau des raccourcis permettent de filtrer le contenu du panneau principal sur la base d'un paramètre qui change d'une vue à l'autre. Pour les pods par exemple, ces touches changent le namespace, mais pour les logs elles changent la plage temporelle à afficher. La valeur du paramètre est rappelée en magenta dans le titre du panneau principal

Dans l'exemple ci-dessus, mon écran se vide lorsque je sélectionne le namespace `default` (`<2>`) car je n'ai pas de pod sur ce namespace. Pensez donc à vérifier régulièrement la valeur de ces paramètres.

Vous avez maintenant toutes les billes pour commencer à faire des opérations sur vos ressources. Voici celles que j'utilise le plus fréquemment.

### 3.3.3.4. Afficher les logs

Dans la vue :pod, la touche `I` permet d'afficher des logs du pod sélectionné. Lorsque qu'un pod n'écrit pas beaucoup de logs, on est généralement confronté au message Waiting for logs....

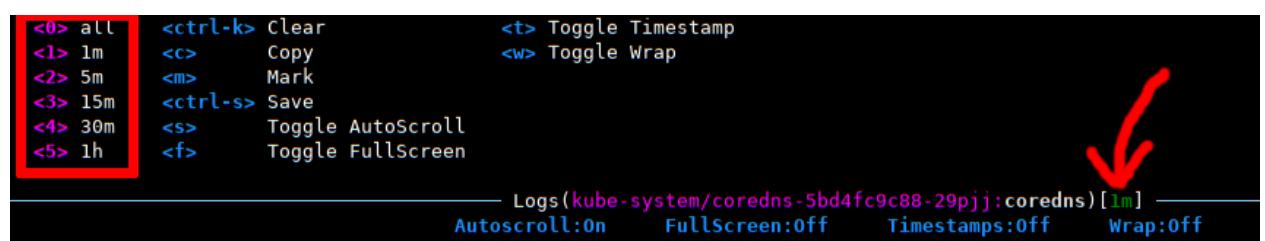
```

Context: xxli-cluster
Cluster: xxli-cluster
User: admin
K8s Rev: v0.26.7
K8s Env: v0.23.12+kos
CPU: 1%
Mem: 13%
Logs(default/platform-mediananager-0-platform)[1m]
Autoscroll:On FullScreen:Off Timestamps:Off Wrap:Off

INFO: 192.168.38.53:5438 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.53:5438 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 10.244.82.127:47598 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:42468 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:42598 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:42598 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 192.168.38.39:52286 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.39:52286 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 10.244.82.127:47979 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 10.244.82.127:48082 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:48082 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 192.168.38.39:53562 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.39:53562 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:46264 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:46264 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:48260 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 192.168.38.30:35254 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.30:35254 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:37934 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:37934 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 192.168.38.39:54160 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.39:54160 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 10.244.82.127:40800 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:40800 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:48816 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:48816 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 192.168.38.30:35179 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.30:35179 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 10.244.82.127:41392 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:41392 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:41398 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 10.244.82.127:41398 - "GET /mediarunnerComposerAPI/ HTTP/1.1" 200 OK
INFO: 192.168.38.30:49336 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK
INFO: 192.168.38.30:49336 - "GET /mediarunnerComposerAPI/healthz HTTP/1.1" 200 OK

```

Cette vue n'affiche en effet que les logs avec un timestamp vieux de maximum 1 minute et quand il n'y en a pas, elle attend alors qu'un log assez jeune arrive. Si vous cherchez des logs plus vieux, vous pouvez changer la plage temporelle avec les touches 0 à 5. La durée sélectionnée est rappelée dans le titre du panneau principal représenté par la flèche rouge dans l'image ci-dessous.



On peut naviguer dans les logs à l'aide des flèches  $\uparrow$  et  $\downarrow$ , toutefois, l'autoscroll est activé par défaut et forcera l'affichage vers le bas dès qu'un nouveau log arrivera. On peut activer / désactiver cette fonction avec la touche s. Aussi, on peut afficher / masquer l'affichage des *timestamps* avec la touche t

N'oubliez pas également que  fonctionne ici également.

### 3.3.3.5. Detruire un pod

ctrl + d permet de détruire une ressource. L'option cascade, activée par défaut, permet de supprimer en plus les ressources qui lui sont associées. Détruire un déploiement en mode cascade détruira également les pods associés à ce déploiement.

```

Container: xxi-cluster <> all <> Attach <> Logs <> YAML
Cluster: xxi-cluster <> default <> default <> Logs Previous
User: admin <> default <> Describe <> Port-Forward
K9s Rev: v0.26.7 <> Edit <> Shell
K9s Rev: v0.23.12+k0s <> Help <> Show Node
CPU: 1s <> Kill <> Show PortForward
MEM: 13% <> Pods(all) [58] -> Delete -> Propagation: Background

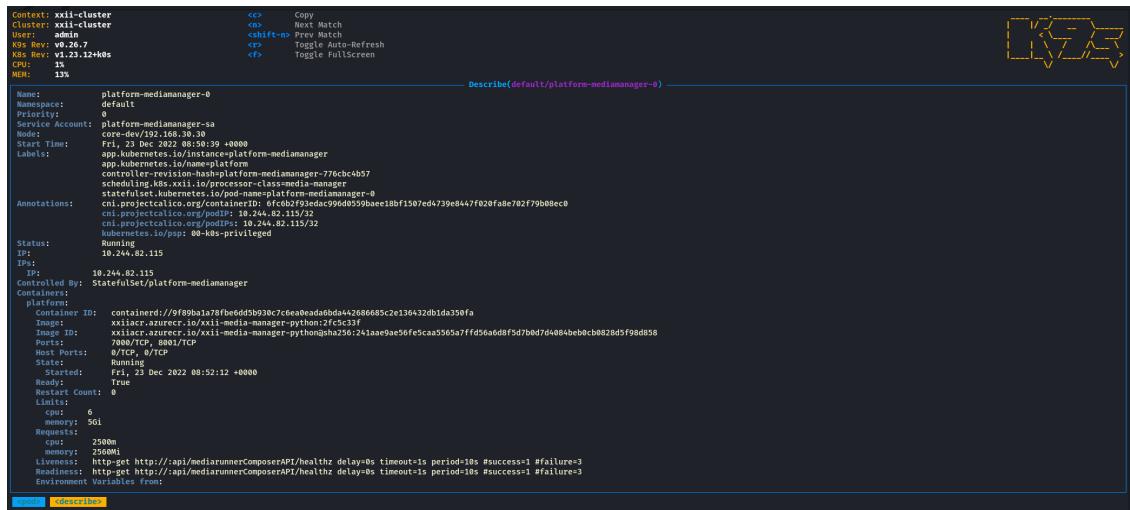
```

NAME/SPACE	NAME	PF	READY	RESTARTS	STATUS	CPU	MEM	%CPU	%CPU/L	%MEM	%MEM/L	IP	NODE	AGE
cattle-fleet-local-system	Fleet-agent-0bdcc869d6-mzm58	●	1/1	0	Running	2	79	n/a	n/a	n/a	n/a	10.244.28.94	core-dev	14d
cattle-fleet-system	Fleet-controller-7dd998df5-zjzfs	●	1/1	0	Running	14	63	n/a	n/a	n/a	n/a	10.244.28.90	core-dev	14d
cattle-system	gitaly-5955555555-5qkxw	●	1/1	0	Running	1	33	n/a	n/a	n/a	n/a	10.244.28.92	core-dev	14d
cattle-system	rancher-785dd07c-bhsqc	●	1/1	0	Running	31	888	n/a	n/a	n/a	n/a	10.244.28.99	core-dev	14d
cattle-system	rancher-webhook-f8785747-x1qg	●	1/1	0	Running	8	52	n/a	n/a	n/a	n/a	10.244.28.99	core-dev	14d
cert-manager	cert-manager-6f938693-5t5v4	●	1/1	0	Running	1	40	n/a	n/a	n/a	n/a	10.244.28.90	core-dev	14d
cert-manager	cert-manager-cns-5550555555-5vbkj	●	1/1	0	Running	1	69	n/a	n/a	n/a	n/a	10.244.28.88	core-dev	14d
cert-manager	cert-manager-webhook-f98d4885-vqtcc	●	1/1	0	Running	4	26	n/a	n/a	n/a	n/a	10.244.28.81	core-dev	14d
default	hivemq-operator-5f96c8dd-fvlgn	●	1/1	0	Running	7	4189	0	100%	10.244.28.94	core-dev	14d		
default	hivemq-operator-operator-f59d058d-8x9cc	●	1/1	0	Running	3	274	0	99%	10.244.28.104	core-dev	14d		
default	nvidia-smi	●	1/1	0	Running	1	2	n/a	n/a	n/a	n/a	10.244.28.126	core-dev	14d
default	platform-composer-v2-9d9b0d-wj1l	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.78	core-dev	7d7h
default	platform-connector-5955555555-7chp9	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.79	core-dev	7d7h
default	platform-connector-genetic-5fd9d56e-86ogm	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.95	core-dev	33m
default	platform-connector-webdashboard-6f5f5874d-9rnbh	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.95	core-dev	7d7h
default	platform-connector-worker-7687059496-zwh6s	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.95	core-dev	7d7h
default	platform-logger-5955555555-59q5n	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.95	core-dev	7d7h
default	platform-logger-5955555555-59q5n	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.95	core-dev	7d7h
default	platform-resultapi-f7cb575c-nmp7h	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.111	core-dev	7d1h
default	platform-scheduler-v2-6f79f79888-s1t3t	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.84	core-dev	13d
default	platform-web-settings-5955555555-59q5n	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.96	core-dev	7d6h
default	platform-web-settings-frontend-75fdf9709-5b9	●	1/1	0	Running	n/a	n/a	n/a	n/a	n/a	n/a	10.244.28.218	core-dev	7d6h
default	platform-web-settings-gateway-55d84addb9-9r45q	●	1/1	0	Running	4	63	n/a	n/a	n/a	n/a	10.244.28.73	core-dev	7d6h
default	platform-worker-5955555555-59q5n	●	1/1	0	Running	1	39	n/a	n/a	n/a	n/a	10.244.28.103	core-dev	14d
default	storage-mongodb-0	●	1/1	0	Running	1	57	n/a	n/a	n/a	n/a	10.244.28.29	core-dev	14d
default	storage-mongodb-0	●	2/2	0	Running	22	776	n/a	n/a	n/a	n/a	10.244.28.101	core-dev	14d
gpu-operator	gpu-discovery-d6n7c	●	2/2	0	Running	6	82	n/a	n/a	n/a	n/a	10.244.28.29	core-dev	3d7h
gpu-operator	gpu-operator-7778778fc-g4rm	●	1/1	0	Running	3	55	n/a	n/a	n/a	n/a	10.244.28.77	core-dev	14d
gpu-operator	gpu-operator-node-feature-discovery-master-78f859d9-h7bok	●	1/1	0	Running	2	37	1	0	37	10.244.28.69	core-dev	14d	
gpu-operator	gpu-operator-node-feature-discovery-worker-595tp	●	1/1	0	Running	4	67	n/a	n/a	n/a	n/a	10.244.28.70	core-dev	14d
gpu-operator	gpu-operator-nvidia-containers-5955555555-59q5n	●	1/1	0	Running	1	59	n/a	n/a	n/a	n/a	10.244.28.70	core-dev	14d
gpu-operator	nvidia-container-toolkit-dnsmonet-6pkpc	●	1/1	0	Running	0	12	n/a	n/a	n/a	n/a	10.244.28.71	core-dev	14d
gpu-operator	nvidia-device-validation-bmrx	●	0/1	0	Completed	0	0	n/a	n/a	n/a	n/a	10.244.28.76	core-dev	14d
gpu-operator	nvidia-device-validation-5955555555-59q5n	●	1/1	0	Running	1	355	n/a	n/a	n/a	n/a	10.244.28.73	core-dev	14d
gpu-operator	nvidia-device-plugin-validation-5955555555-59q5n	●	1/1	0	Running	1	25	n/a	n/a	n/a	n/a	10.244.28.73	core-dev	14d
gpu-operator	nvidia-device-plugin-validation-st5688	●	0/1	0	Completed	0	0	n/a	n/a	n/a	n/a	10.244.28.78	core-dev	14d
gpu-operator	nvidia-driver-dnsmonet-q32	●	1/1	0	Running	0	1300	n/a	n/a	n/a	n/a	10.244.28.75	core-dev	14d
gpu-share	nvidia-share-validator-5955555555-59q5n	●	1/1	0	Running	0	2	n/a	n/a	n/a	n/a	10.244.28.75	core-dev	14d
gpu-share	gpu-share-device-plugin-jdvs9	●	1/1	0	Running	1	28	2	0	9	5	192.168.30.30	core-dev	14d
gpu-share	gpu-share-scheduler-extender-8494479d9c-spqc9	●	1/1	0	Running	2	17	n/a	n/a	n/a	n/a	192.168.30.30	core-dev	14d

Il est aussi possible de faire un *kill* avec `ctrl + k` pour les pods, mais faites attention, cette opération ne demande pas de confirmation ! 💣

### 3.3.3.6. Décrire / Editer une ressource

Il est parfois intéressant de lire la description d'une ressource pour vérifier sa liste de labels par exemple. Pour cela on peut soit utiliser la touche d qui retourne la description formatée par k9s, soit la touche y pour obtenir le fichier YAML brut. Dans ces vues, lorsque l'on utilise le mode filtre et qu'il correspond à plusieurs résultats, on pourra naviguer vers l'occurrence suivante avec la touche n et l'occurrence précédente avec shift+n (pensez bien à valider le filtre avec ↵ avant d'appuyer sur ces touches).



```
Context: xxii-cluster          <=> Copy
Cluster: xxii-cluster          <=> Next Match
User: adm                         <=> Previous Match
K9s: v0.26.7                      <=> Toggle FullScreen
K9s Rev: v1.23.12+k8s           <=> Toggle FullScreen
CPU:   1%                         <=> Toggle FullScreen
MEM:  33%                         <=> Toggle FullScreen

Name: platform-mediamanager-0
Namespace: default
Priority: 0
Service Account: platform-mediamanager-sa
Node:   f1a23... 2022-08-15T20:40:0000
Start Time: Fri, 23 Dec 2022 08:52:12 +0000
Labels: app.kubernetes.io/instance=platform-mediamanager
        app.kubernetes.io/name=platform-mediamanager
        controller.k8s.xxii.io/processor-class=media-manager
Annotations: cni.projectcalico.org/endpointIPs: 10.244.82.115/32
              cni.projectcalico.org/peerdIPs: 10.244.82.115/32
              cni.projectcalico.org/podIPs: 10.244.82.115/32
              kubernetes.io/psp: 0:k8s-privileged
Status: Running
IP:   10.244.82.115
IPs:  10.244.82.115
Controlled By: StatefulSet/platform-mediamanager
Containers:
  platform:
    Image ID: xxiiacr.azurecr.io/xxii-media-manager-python:2fcisc3f
    Image: xxiiacr.azurecr.io/xxii-media-manager-python:sha256:241aae9ae56fe5caa55d6a7ff56a6d8f5d7bd7d4084beb8cbe828d5f98dd58
    Port:  80/TCP, 8000/TCP
    Ports: 80/TCP, 8000/TCP
    State:  Running
    Started: Fri, 23 Dec 2022 08:52:12 +0000
    Progress: 100%
    Restart Count: 0
    Limits:
      CPU: 6
      Memory: 5Gi
    Requests:
      CPU: 2500m
      Memory: 2500Mi
    Liveness: http-get http://:80/api/mediarunnerComposerAPI/healthz delay=0s timeout=1s period=10s #success=1 #failure=3
    Readiness: http-get http://:80/api/mediarunnerComposerAPI/healthz delay=0s timeout=1s period=10s #success=1 #failure=3
    Environment Variables From:
```

Il est également possible d'éditer le fichier YAML correspondant à une ressource en appuyant avec e. Cela ouvrira votre éditeur de texte par défaut et appliquera les modifications une fois le fichier sauvegardé.

### 3.3.3.7. Vues spéciales

**Pulses** est un dashboard global qui permet de voir l'état de santé de votre cluster. Il affiche le nombre de ressources OK / KO au cours du temps et permet via les touches numériques ou tab de passer rapidement dans la vue la plus pertinente pour surveiller un paramètre. On active sa vue avec :pulses

