



Webinar:

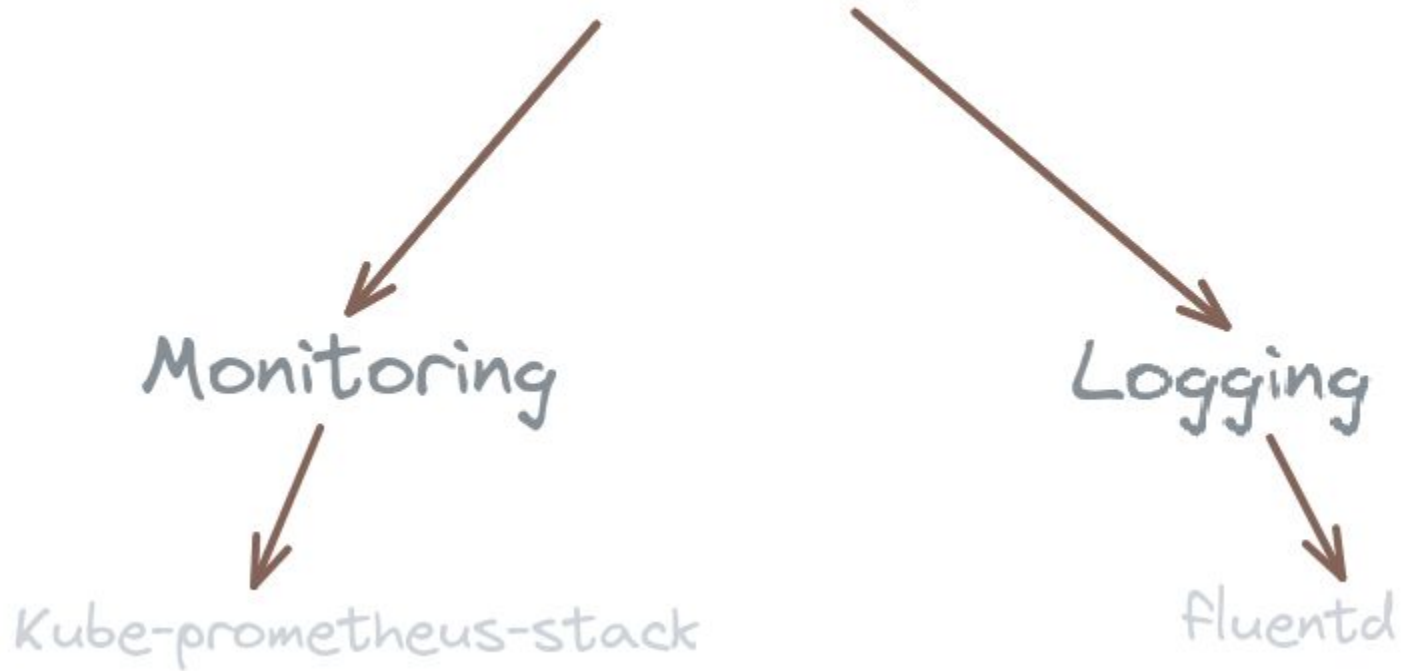
Integrating Kubernetes with the CERN monitoring stack

Diana Gaponcic, IT-CD-PI



Guillermo Facundo Colunga, IT-DA-ASM

What we currently offer



Monitoring

```
openstack coe cluster create monitoring-cluster  
  --merge-labels --labels monitoring_enabled=true ...
```

NAME	READY	STATUS	RESTARTS	AGE
alertmanager-cern-magnum-kube-prometheu-alertmanager-0	2/2	Running	0	42m
cern-magnum-grafana-7f4576cdb5-f58cn	3/3	Running	0	42m
cern-magnum-kube-prometheu-operator-64f6c6958d-c4db5	1/1	Running	0	42m
cern-magnum-kube-state-metrics-779c6977f9-mwcs5	1/1	Running	0	42m
cern-magnum-prometheus-adapter-774855d9fd-tw25m	1/1	Running	0	42m
cern-magnum-prometheus-node-exporter-crn8s	1/1	Running	0	42m
cern-magnum-prometheus-node-exporter-kfc5n	1/1	Running	0	42m
prometheus-cern-magnum-kube-prometheu-prometheus-0	2/2	Running	0	42m



Logging

```
openstack coe cluster create monitoring-cluster  
  --merge-labels --labels logging_producer=MYPRODUCER ...
```

```
cern-magnum-fluentd-9jtpc      1/1      Running    0        41m  
cern-magnum-fluentd-lrvvd     1/1      Running    0        41m
```

Logging

The configuration deployed by default:

 filters.conf.yaml	allow not install default fluent conf	6 months ago
 outputs.conf.yaml	allow not install default fluent conf	6 months ago
 sources.conf.yaml	allow not install default fluent conf	6 months ago

99f21fe8-9f3a-49f5-9412-092e0aa9ccbba and prometheus-adapter

DQL



~ 15 minutes ago → now

Refresh

+ Add filter

2 hits

Oct 3, 2024 @ 17:20:30.880 - Oct 3, 2024 @ 17:35:30.881 per Minute



Time _source

- > Oct 3, 2024 @ 17:22:38.446 `data.kubernetes.container_name:` prometheus-adapter `data.kubernetes.namespace_name:` kube-system `data.kubernetes.pod_name:` cern-magnum-prometheus-adapter-774855d9fd-88wlx
`data.kubernetes.container_image:` registry.cern.ch/kubernetes/prometheus-adapter:v0.9.1 `data.kubernetes.container_image_id:` registry.cern.ch/kubernetes/prometheus-adapter@sha256:d025d1a109234c28b4a97f5d35d759943124be8885a5bce22a91363025304e9d `data.kubernetes.pod_id:` 454bdf45-7bc7-4c72-9df9-3b2a546e4c58 `data.kubernetes.pod_ip:` 10.100.37.132
`data.kubernetes.host:` kops-registry-next-l7s4rzglb5ab-node-0 `data.kubernetes.labels.pod-template-hash:` 774855d9fd `data.kubernetes.labels.app_kubernetes_io/component:` metrics
`data.kubernetes.labels.app_kubernetes_io/instance:` cern-magnum `data.kubernetes.labels.app_kubernetes_io/managed-by:` Helm `data.kubernetes.labels.app_kubernetes_io/name:` prometheus
- > Oct 3, 2024 @ 17:22:38.446 `data.kubernetes.container_name:` prometheus-adapter `data.kubernetes.namespace_name:` kube-system `data.kubernetes.pod_name:` cern-magnum-prometheus-adapter-774855d9fd-88wlx
`data.kubernetes.container_image:` registry.cern.ch/kubernetes/prometheus-adapter:v0.9.1 `data.kubernetes.container_image_id:` registry.cern.ch/kubernetes/prometheus-adapter@sha256:d025d1a109234c28b4a97f5d35d759943124be8885a5bce22a91363025304e9d `data.kubernetes.pod_id:` 454bdf45-7bc7-4c72-9df9-3b2a546e4c58 `data.kubernetes.pod_ip:` 10.100.37.132
`data.kubernetes.host:` kops-registry-next-l7s4rzglb5ab-node-0 `data.kubernetes.labels.pod-template-hash:` 774855d9fd `data.kubernetes.labels.app_kubernetes_io/component:` metrics
`data.kubernetes.labels.app_kubernetes_io/instance:` cern-magnum `data.kubernetes.labels.app_kubernetes_io/managed-by:` Helm `data.kubernetes.labels.app_kubernetes_io/name:` prometheus

But...

Users asked for more flexibility

User defined configuration

In addition to the default configuration, it is possible to pass a user defined one. To do so, the cluster configuration is passed at cluster creation via a single yaml blob.

For this we need to:

- disable the default fluentd configuration
- pass the user defined one

```
cat << EOF >> cluster-user-config.yaml
fluentd:
  enabled: true
  deployDefaultConfiguration: false
  fileConfigs:
    01_sources.conf: |-
      ...
    02_filters.conf: |-
      ...
    03_dispatch.conf: |-
      ...
    04_outputs.conf: |-
      ...
EOF
```


Logging

Flexible enough to cover most use case

Logging

Flexible enough to cover most use case, but...

Logging


Some issues we experienced in time:

- Fluentd buffer filling up the disk on the node
- It became hard to tweak the fluentd buffer configuration
- Some clusters experienced issues after enabling fluentd

Logging

We started considering moving to  fluentbit

Logging

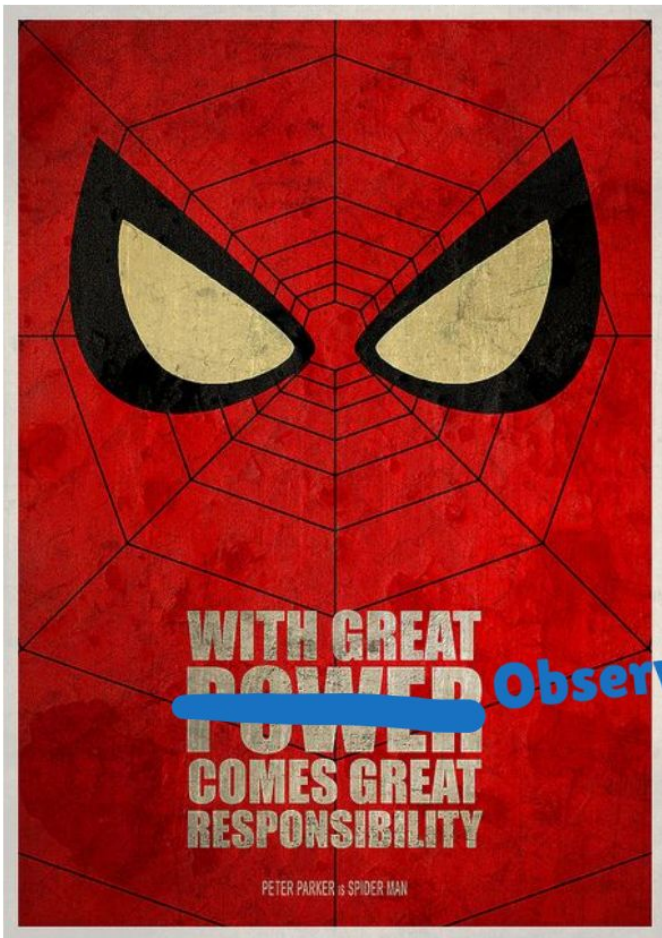
We started considering moving to  fluentbit
but no manpower available

What we can do instead?

What we can do instead?

Monitoring team to the rescue





Observability

[Source](#)

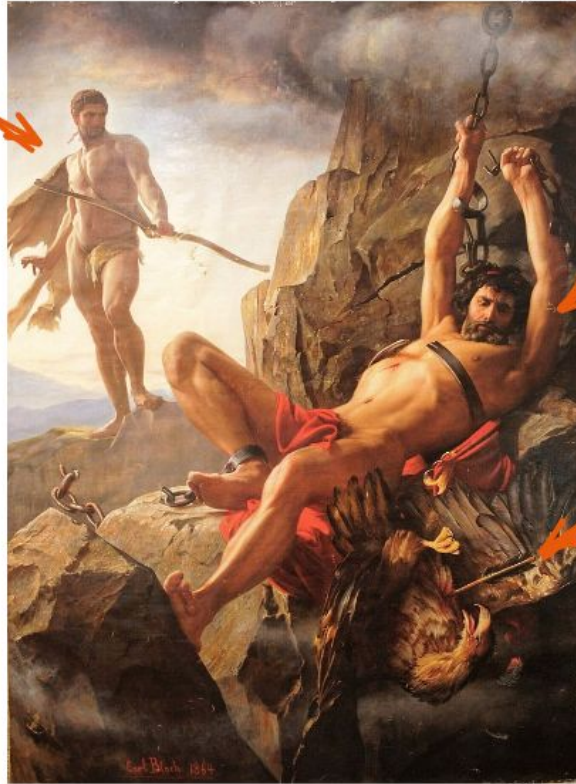


[Source](#)

Benefits from collaborating with the monitoring team:

- New features for metrics and logs infrastructure
- CERN monit integrations installed by default
- More domain knowledge from the monitoring experts

Monitoring
team



K8s-team-induced
torment

The Release of Prometheus, oil on canvas by [Carl Bloch](#), 1864.

Disclaimer

This will be available in the next releases. Follow the kubernetes and monitoring documentation, and the announcements.

- <https://kubernetes.docs.cern.ch/>
- <https://monit-docs.web.cern.ch/>

<https://gitlab.cern.ch/monitoring/helm-charts/kubernetes-monitoring>

CERN IT Monitoring Kubernetes Helm Chart

Overview

This Helm chart facilitates the deployment of a comprehensive monitoring solution for Kubernetes clusters at CERN. It enables the collection of metrics, logs, and, in the future, traces. The chart deploys and configures necessary components to gather and forward metrics and logs to the central monitoring system.

What IT Monitoring Team Provides for Kubernetes Monitoring?

A Helm chart to...

- Collect **metrics** from your kubernetes **cluster components** (ex. calico, control plane api, etc...).
- Collect **metrics** from the **applications** running inside your kubernetes clusters.
- Collect **logs** from your kubernetes **cluster components**.
- Collect **logs** from the **application** running inside your kubernetes clusters.

Store all that information within the IT Monitoring **central infrastructure**.

- For **metrics** the default **retention period** is **40 days and 150K metrics in a 2h window**.
- For **logs** the default **retention period** is either **10Gi or 30 days***.

Consume all that information with IT Monitoring **Grafana / OpenSearch**.

IT Monitoring for Kubernetes Helm Chart: First things first

You will need to **request a tenant** (kind of logical separation for users and quota allocation).

- Please create a **request** in cern.ch/monit-support.
- We will need you to **provide** us with:
 - Tenant **name** (ex. monitoring, swan, swan_qa, etc...).
 - If you **plan** to send **metrics, logs** or **both**.
 - Indicate **any special request** like provide more initial quota (can be increased later).

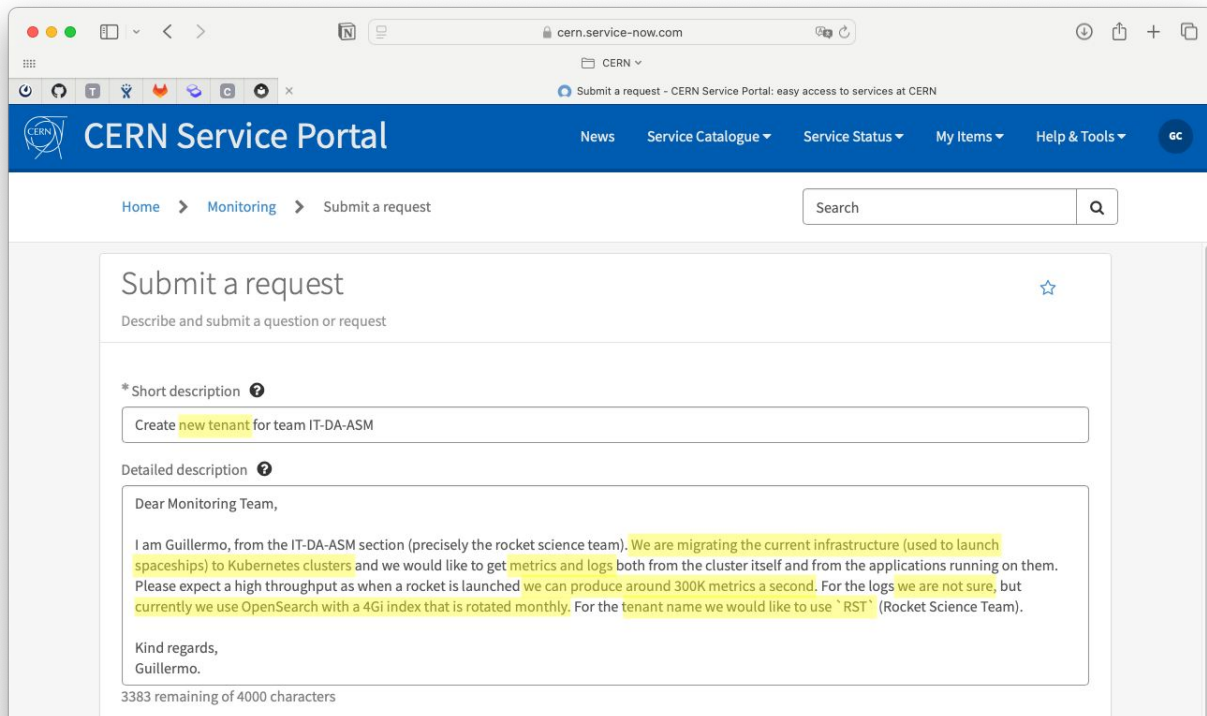
We will reply to you with your confirming your **tenant name** and **password** (we generate it).

- This **values** are **required to send metrics / logs** from Kubernetes.

A **single tenant** can be used from **multiple clusters** (recommended).

A **single cluster** can only send metrics to a **single tenant**.

IT Monitoring for Kubernetes Helm Chart: First things first



The screenshot shows a web browser window displaying the CERN Service Portal. The page title is "Submit a request" and the breadcrumb navigation is "Home > Monitoring > Submit a request". The form contains the following fields and text:

Short description ?
Create **new tenant** for team IT-DA-ASM

Detailed description ?
Dear Monitoring Team,
I am Guillermo, from the IT-DA-ASM section (precisely the rocket science team). We are migrating the current infrastructure (used to launch spaceships) to Kubernetes clusters and we would like to get **metrics and logs** both from the cluster itself and from the applications running on them. Please expect a high throughput as when a rocket is launched we can produce around 300K metrics a second. For the logs we are not sure, but currently we use OpenSearch with a 4Gi index that is rotated monthly. For the **tenant name we would like to use `RST`** (Rocket Science Team).

Kind regards,
Guillermo.

3383 remaining of 4000 characters

IT Monitoring for Kubernetes Helm Chart: Components

The Helm chart is **split** into **metrics** and **logs**. Both functionalities are **independent** and can be **enabled** or **disabled** without affecting the other.

Metrics:

- **Node Exporter** for metrics from **nodes**.
- **Kubestate** for metrics about **Pods**.
- **Kubernetes API** for metrics about **control plane**.
- **Prometheus Operator** as **collector** agent. Allows **extensibility** by users.
- **Fluent Bit** as **forwarding** agent to send metrics to **Open Telemetry** IT Monitoring endpoints.

Logs:

- **Fluent Bit** as a **collector** and **forwarder** to **Open Telemetry** IT Monitoring endpoints.
- **Kubernetes Fluent Bit filter** to **enrich logs** with Kubernetes API information.

IT Monitoring for Kubernetes Helm Chart: Helm Chart Values

Most important sections are **tenant**, **kubernetes**, **metrics** and **logs**. They allow us to configure and enable/disable the different chart components.

```
tenant:
  # -- username used for authenticating in the MONIT infrastructure
  name: nil
  # -- password (plain) used for authenticating in the MONIT infrastructure
  password: nil

kubernetes:
  # -- name of the kubernetes cluster to monitor. This value will be appended to every metric and log via k8s_cluster_name label
  clusterName: nil

metrics:
  # -- indicates if all metrics components should be enabled or not.
  enabled: true

logs:
  # -- indicates if logs metrics components should be enabled or not.
  enabled: false

fluentbit:
  # -- indicates if fluent bit logs component should be installed or not
  enabled: false
```

IT Monitoring for Kubernetes Helm Chart: Installing

First step is to install the Custom Resource Definitions that the monitoring chart uses.

```
helm install cern-it-monitoring-kubernetes-crds oci://registry.cern.ch/monit/cern-it-monitoring-kubernetes-crds \  
  --version 1.0.0 \  
  --namespace monitoring
```

Second step is to actually install the helm chart. (Notice myCustomValues.yaml file).

```
helm install cern-it-monitoring-kubernetes oci://registry.cern.ch/monit/cern-it-monitoring-kubernetes \  
  --version 1.0.0 \  
  --values myCustomValues.yaml \  
  --namespace monitoring
```

Example of the contents of the myCustomValues.yaml file.

```
tenant:  
  name: rst # Rocket Science Team  
  password: firstOnesToArrive  
kubernetes:  
  clusterName: gfacundo-dev
```

IT Monitoring for Kubernetes Helm Chart: Getting Started Metrics

The **basic** metrics configuration provides you with **kubernetes cluster metrics**.

You can **extend** this to gather **metrics** from **other components**.

```
metrics:
  enabled: true
  server:
    # -- The serviceMonitors array allows you to define extra scrape targets during installation (ex. magnum provided nginx).
    serviceMonitors:
      - name: my-nginx-servicemonitor
        spec:
          endpoints:
            - interval: 30s # How often prometheus will ask for metrics
              port: metrics # In which port of your service the metrics are exposed.
              path: /prometheus-metrics # If not set will be /metrics.
          namespaceSelector:
            matchNames:
              - kube-system
          selector:
            matchLabels:
              app.kubernetes.io/component: controller
              app.kubernetes.io/instance: cern-magnum
              app.kubernetes.io/name: ingress-nginx
```

IT Monitoring for Kubernetes Helm Chart: Customizing Metrics

For **user applications** metrics we recommend using a **ServiceMonitor**.

The installed **Prometheus** is **listening** for **any ServiceMonitor** registered within the cluster.

More information about ServiceMonitor can be found in [Prometheus Operator Docs](#).

```
myservice-backend/  
deployment.yaml  
service.yaml  
ingress.yaml  
servicemonitor.yaml
```

```
apiVersion: monitoring.coreos.com/v1  
kind: ServiceMonitor  
metadata:  
  name: myservice-backend-monitor  
  labels:  
    app: myservice-backend  
spec:  
  selector:  
    matchLabels:  
      app.kubernetes.io/name: myservice-backend-service  
  endpoints:  
  - port: http  
    path: /metrics  
    interval: 5s
```

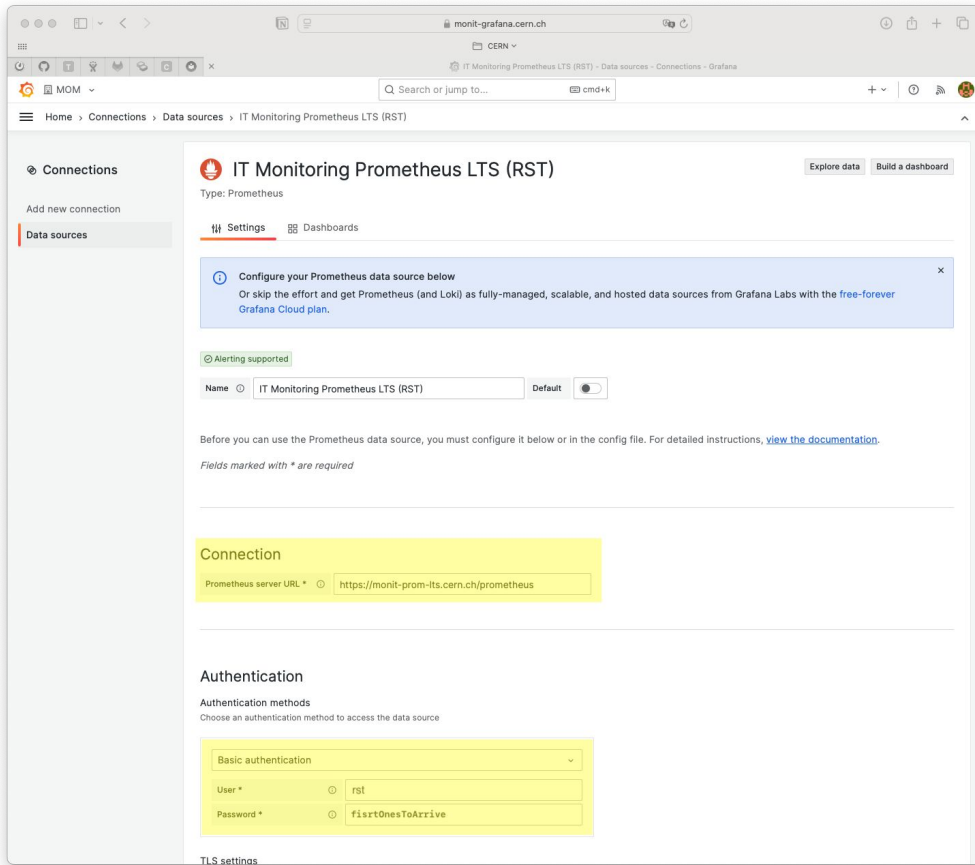
IT Monitoring for Kubernetes Helm Chart: Visualizing Metrics

You need a Grafana Organisation.

- You can request one via cern.ch/monit-support.
- Access via monit-grafana.cern.ch.

Add a new Prometheus Datasource.

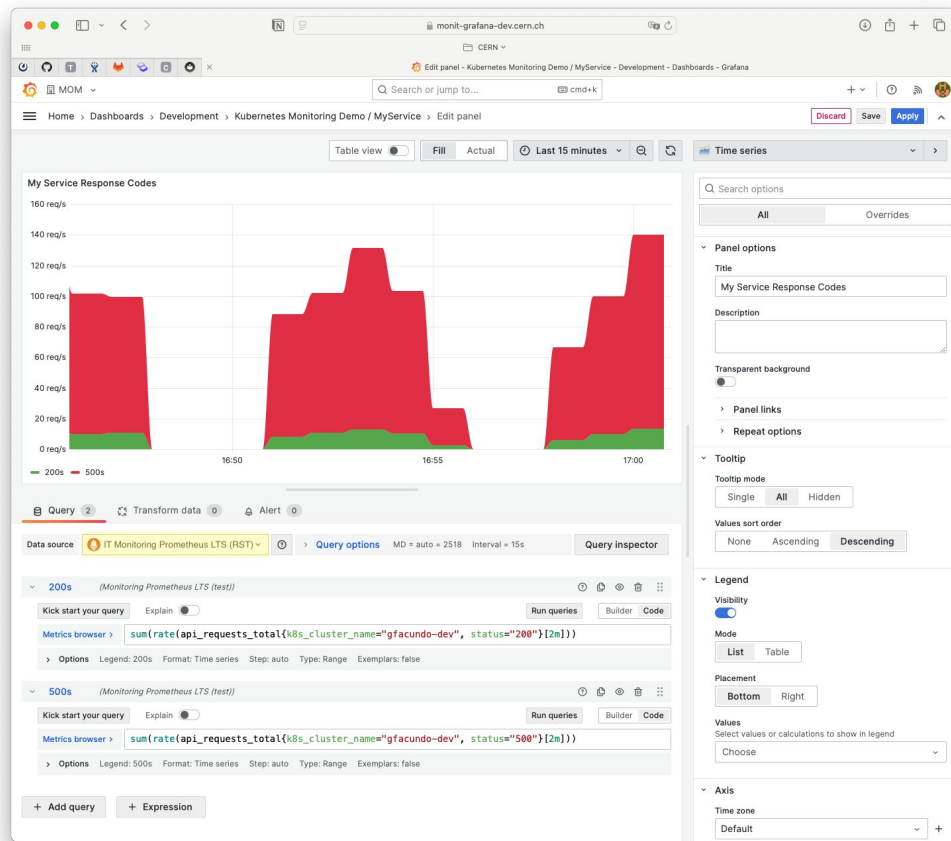
- **URL:** `https://monit-prom-lts.cern.ch/prometheus`
- **Authentication:** Basic Authentication
- **User:** `<your_tenant_name>`
- **Password:** `<your_tenant_password>`
- *No other parameter needs to be changed.*



IT Monitoring for Kubernetes Helm Chart: Visualizing Metrics

Then...

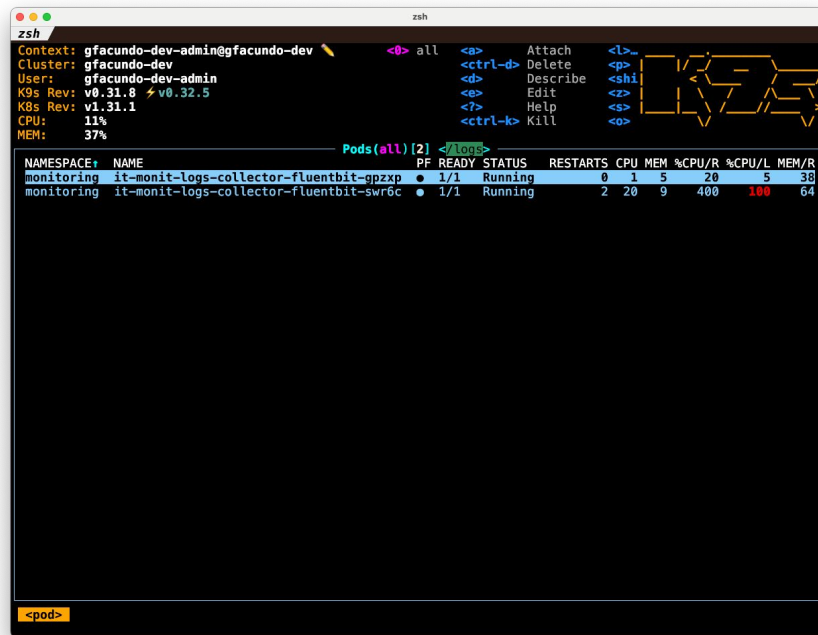
1. **Create a new Dashboard.**
2. **Create a new Visualization.**
3. **Select your new Datasource.**
4. **Add your Queries** 🎉.



IT Monitoring for Kubernetes Helm Chart: Getting Started [Logs](#)

Enabling logging components will send to us any container-produced logs (control plane and applications). Logs are gathered from the **stdout** and **stderr**, but you can **customize the inputs and parsers**.

```
logs:
  enabled: true
  fluentbit:
    enabled: true
    inputs: |
      [INPUT]
      Name tail
      Path /var/log/containers/*.log
      multiline.parser cri
      Tag kube.*
      Mem_Buf_Limit 20MB
      Skip_Long_Lines Off
    customParsers: |
      [PARSER]
      Name yourParser
      ...
```



The screenshot shows a terminal window with the following content:

```
zsh
Context: gfacundo-dev-admin@gfacundo-dev
Cluster: gfacundo-dev
User: gfacundo-dev-admin
K9s Rev: v0.31.8 ↗ v0.32.5
K8s Rev: v1.31.1
CPU: 11%
MEM: 37%
```

Pod status table:

NAMESPACE	NAME	PF	READY	STATUS	RESTARTS	CPU	MEM	%CPU/R	%CPU/L	MEM/R
monitoring	it-monit-logs-collector-fluentbit-gpzyx	●	1/1	Running	0	1	5	20	5	38
monitoring	it-monit-logs-collector-fluentbit-swr6c	●	1/1	Running	2	20	9	400	100	64

Pod architecture diagram:

```
graph TD
  subgraph Pod
    direction TB
    Container1[Container 1]
    Container2[Container 2]
    Container3[Container 3]
    Container4[Container 4]
    Container5[Container 5]
    Container6[Container 6]
    Container7[Container 7]
    Container8[Container 8]
    Container9[Container 9]
    Container10[Container 10]
  end
```

IT Monitoring for Kubernetes Helm Chart: Customizing Logs

You can **customize** the Fluent Bit **service, inputs, filters, parsers and outputs.**

Every section is **configured out of the box to work with the IT Monitoring Central Infrastructure.**

[INPUT]

```
Name tail
Path /var/log/containers/*.log
multiline.parser cri
Tag kube.*
Mem_Buf_Limit 20MB
Skip_Long_Lines Off
```

[FILTER]

```
Name          kubernetes
Match         kube.*
Kube_URL      https://kubernetes.default.svc:443
Kube_CA_File  /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
Kube-Token_File /var/run/secrets/kubernetes.io/serviceaccount/token
Kube_Tag_Prefix kube.var.log.containers.
Merge_Log     On
Merge_Log_Trim On
Keep_Log      On
K8S-Logging.Parser On
K8S-Logging.Exclude On
Annotations  Off
Labels        Off
```

[OUTPUT]

```
name opentelemetry
match *
host {{ .Values.otlp.endpoint }}
port {{ .Values.otlp.port }}
metrics_uri /v1/metrics
logs_uri /v1/logs
traces_uri /v1/traces
tls on
http_user {{ .Values.tenant.name }}
http_passwd {{ .Values.tenant.password }}
header tag_monit
header log_type kubernetes
header User-Agent {{ .Chart.Name }}/{{ .Chart.Version }}
logs_body_key imaginary_non_existing_field
```

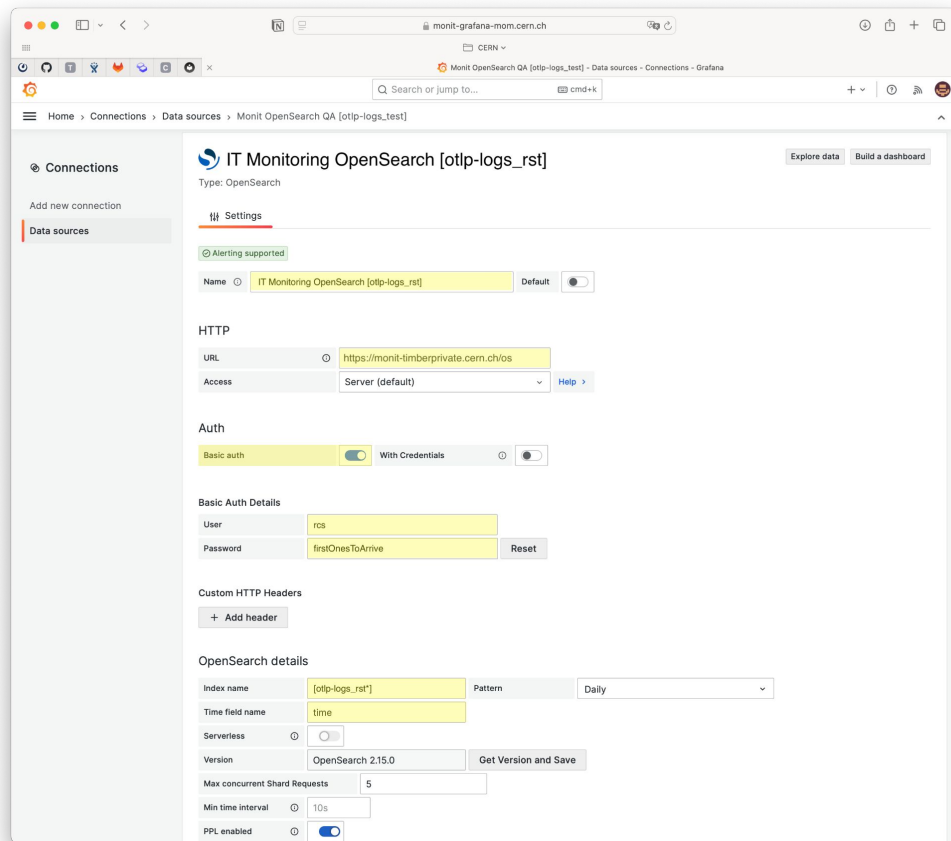

IT Monitoring for Kubernetes Helm Chart: Visualizing Logs

You need a Grafana Organisation.

- You can **request** one via cern.ch/monit-support.
- **Access** via monit-grafana.cern.ch.

Add a new **OpenSearch Datasource**.

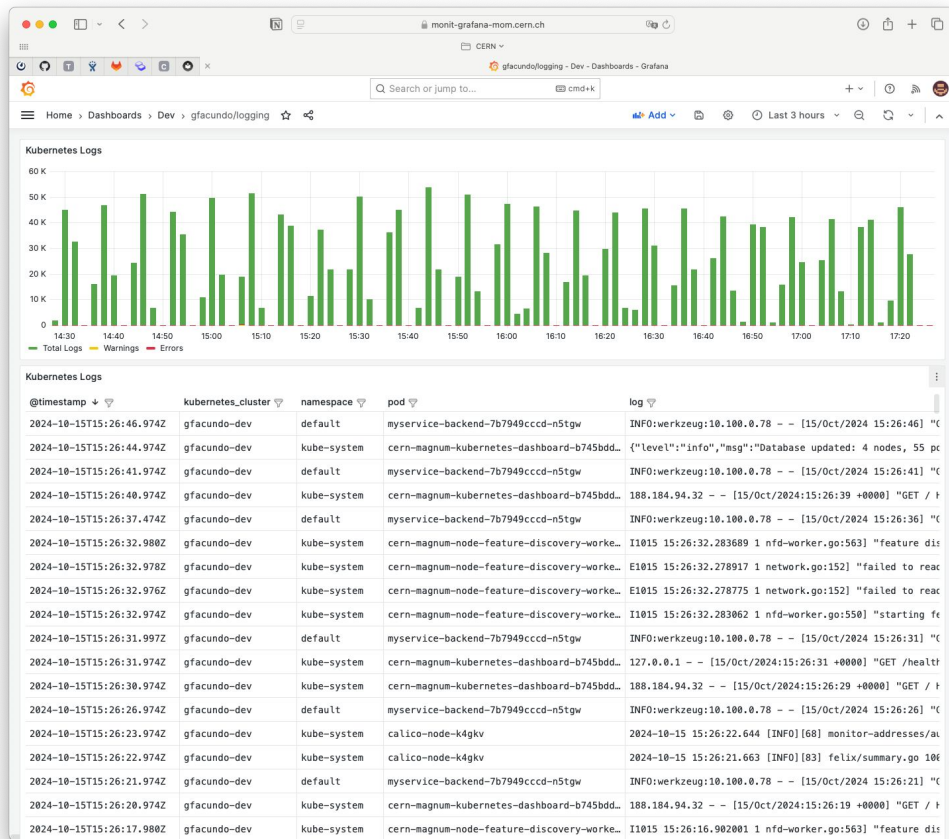
- **URL:** `https://monit-timberprivate.cern.ch/os`
- **Authentication:** Basic auth
- **User:** `<your_tenant_name>`
- **Password:** `<your_tenant_password>`
- *No other parameter needs to be changed.*



IT Monitoring for Kubernetes Helm Chart: Visualizing Logs

Then...

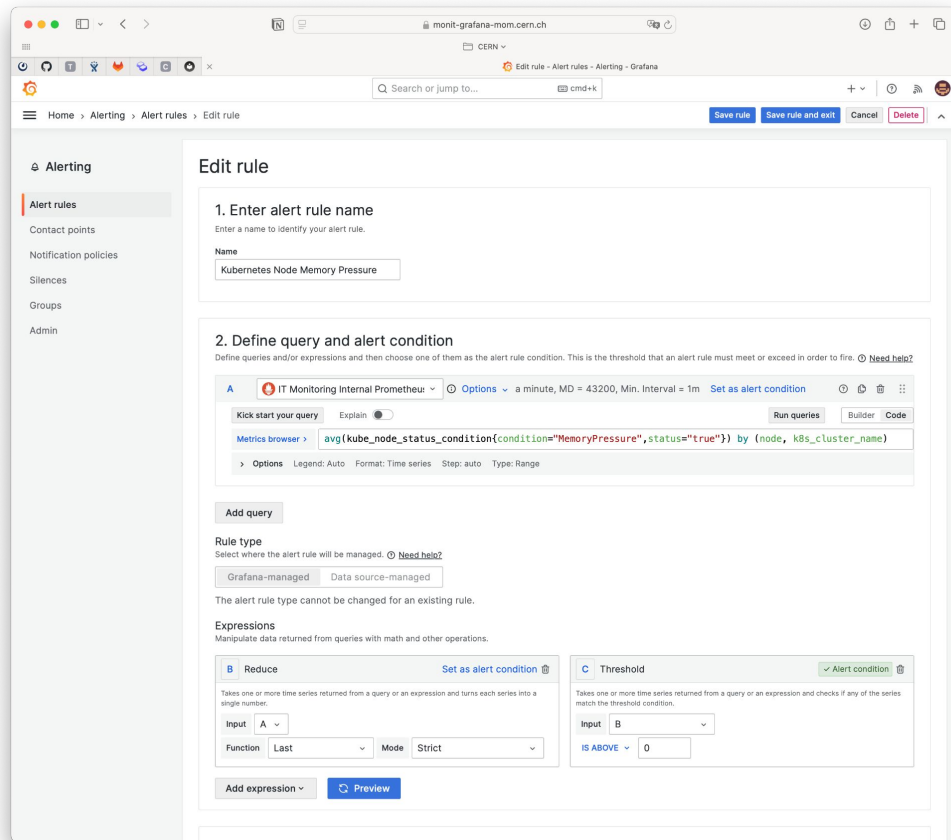
1. Create a new Dashboard.
2. Create a new Visualization.
3. Select your new Datasource.
4. Add your Queries 🎉



IT Monitoring for Kubernetes Helm Chart: Creating Alarms

From Grafana:

1. Go to **Alerting > Alert Rules**.
2. **Select the Datasource**.
3. **Query the metric** you want to eval.
4. **Set a threshold**.



*Wait until stable
release to use it.*

IT Monitoring for Kubernetes Helm Chart: Take Away

Helm chart to monitor your Kubernetes **clusters** and the **applications on them**:

- Chart **Repository**: gitlab.cern.ch/monitoring/helm-charts/kubernetes-monitoring
- Chart **Documentation**: gitlab.cern.ch/monitoring/helm-charts/kubernetes-monitoring/docs
- **Support**: cern.ch/monit-support

Store all that information within the IT Monitoring **central infrastructure**.

- For **metrics** the default **retention period** is **40 days** and **150K data points** in a **2h window**.
- For **logs** the default **retention period** is either **10Gi** or **30 days**.

Visualize your **metrics** and **logs** via:

- **Grafana**: monit-grafana.cern.ch
- **OpenSearch**: monit-timberprivate.cern.ch



Webinar:

Integrating Kubernetes with the CERN monitoring stack

Diana Gaponcic, IT-CD-PI



Guillermo Facundo Colunga, IT-DA-ASM