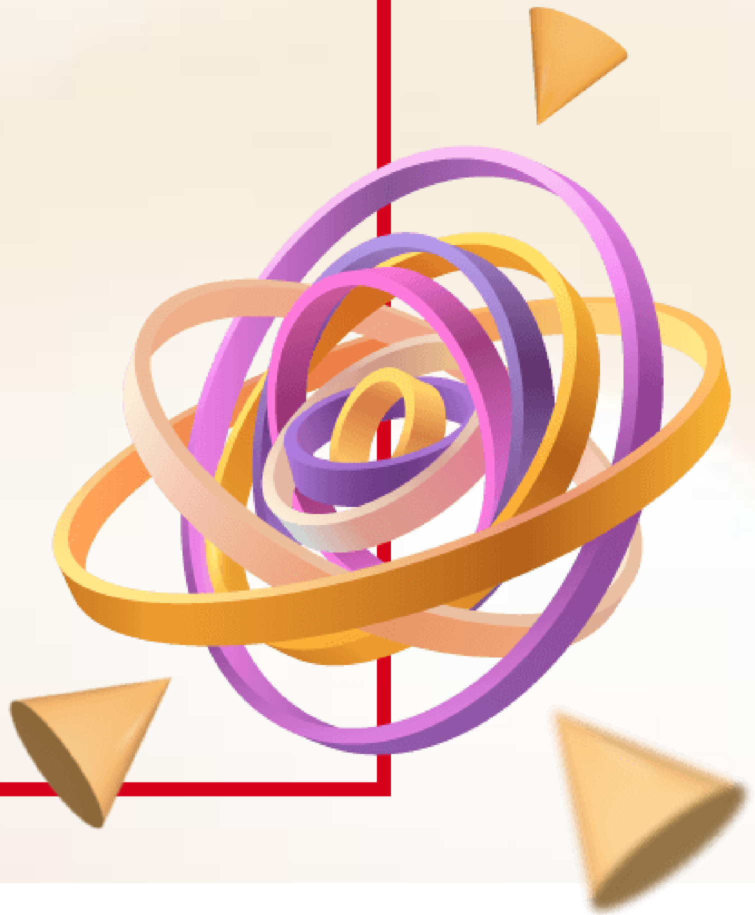
Monitoring with Prometheus and Grafana - Hands on



Nguyen Vu Ninh

Aug 2023

Nash Tech.

Agenda

Prometheus and Grafana Intro
- Prometheus Operator
- Installation with Helm

Prometheus Scrape Configuration
- ScrapeConfig vs ServiceMonitor
- ServiceMonitor setup

Grafana Dashboard
- Visualize metric on dashboard
- Connect Grafana with Cloudwatch

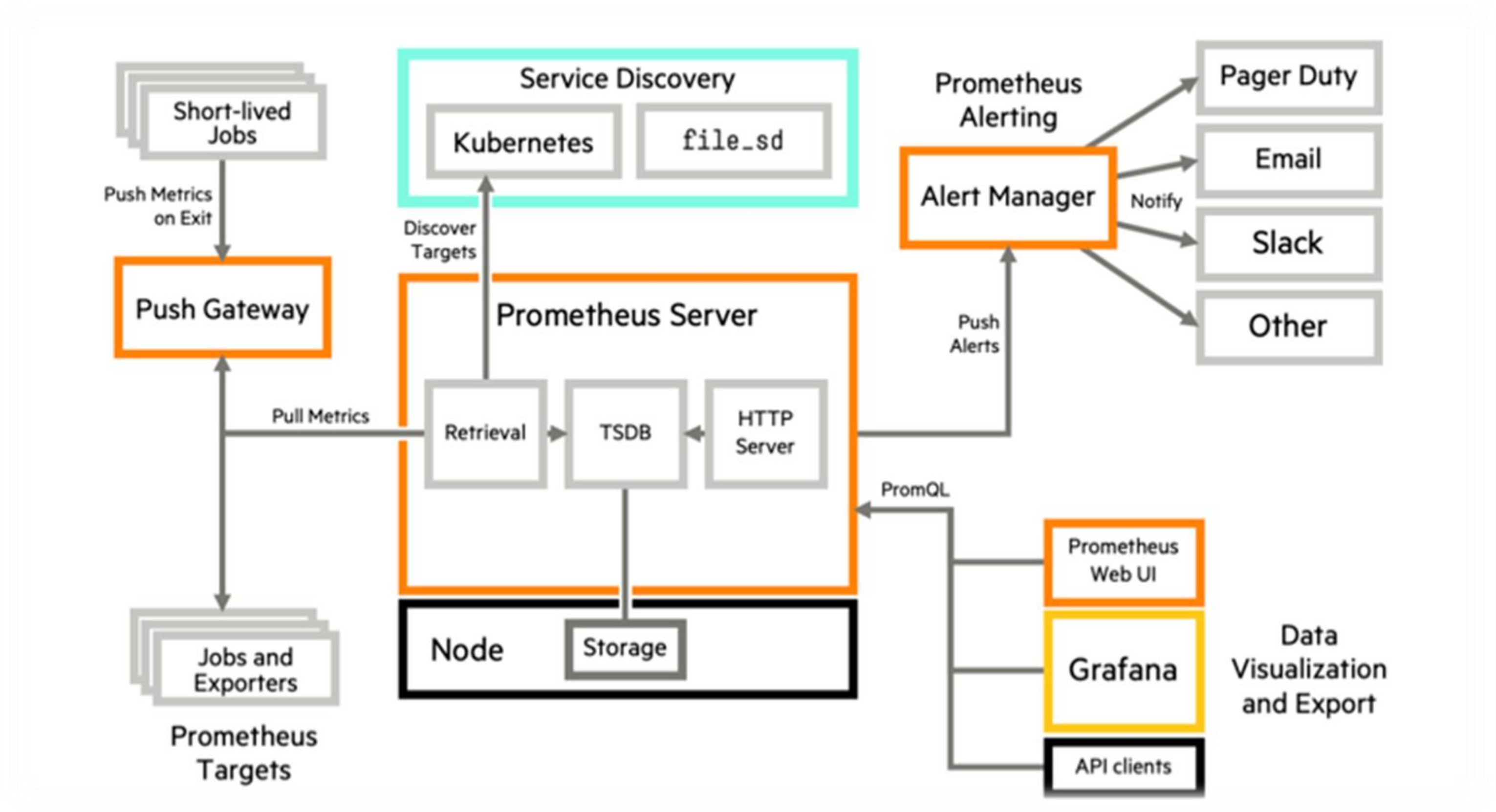
AlertManager vs Grafana Alerts
- Slack Notification with AlertManager
- Slack Notification with Grafana Alerts

Prometheus Adapter
- Setup custom rule for Prometheus Adapter
- HPA Scale based on custom-metric

Prometheus and Grafana Intro

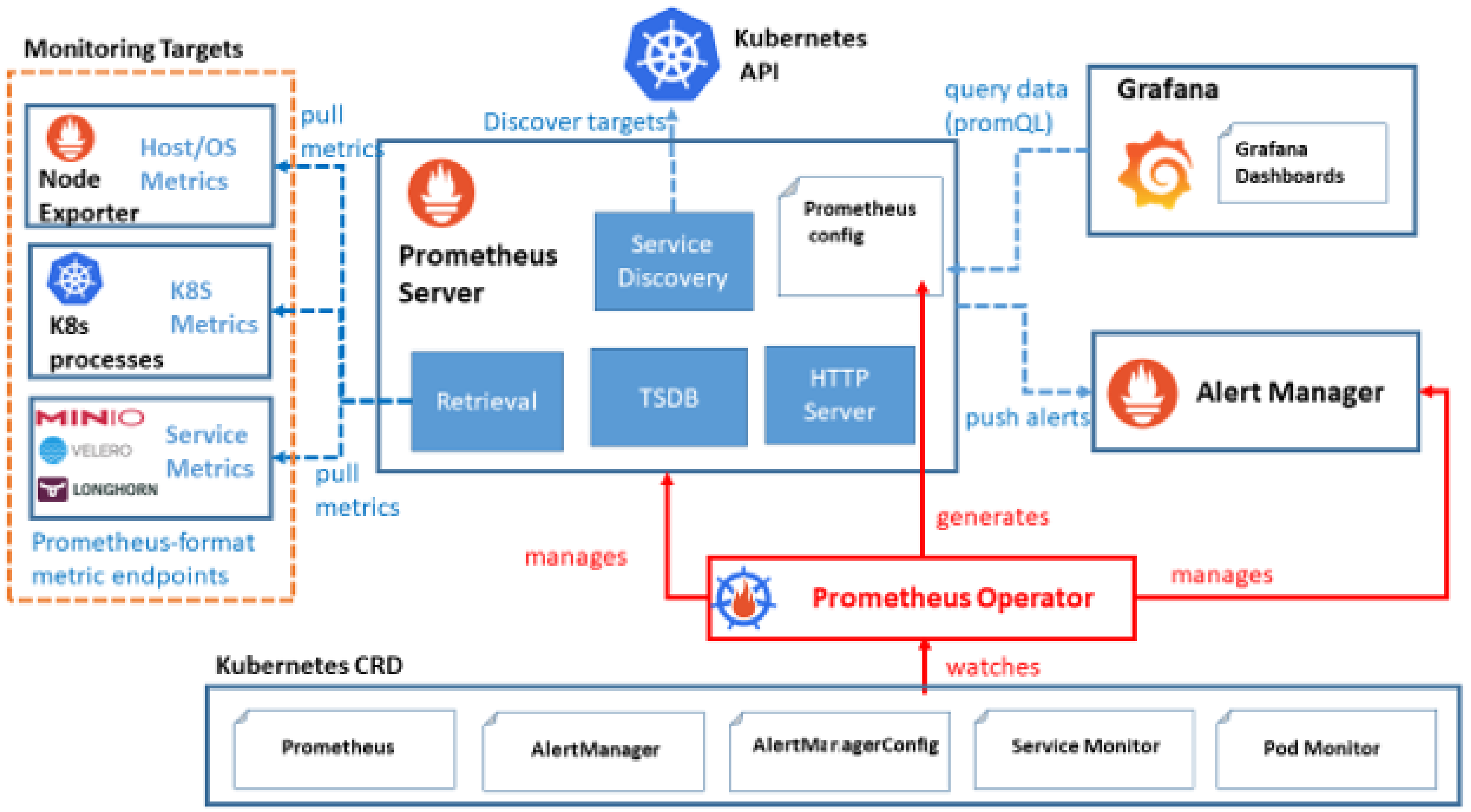
Prometheus and Grafana Intro

Prometheus and Grafana Overview



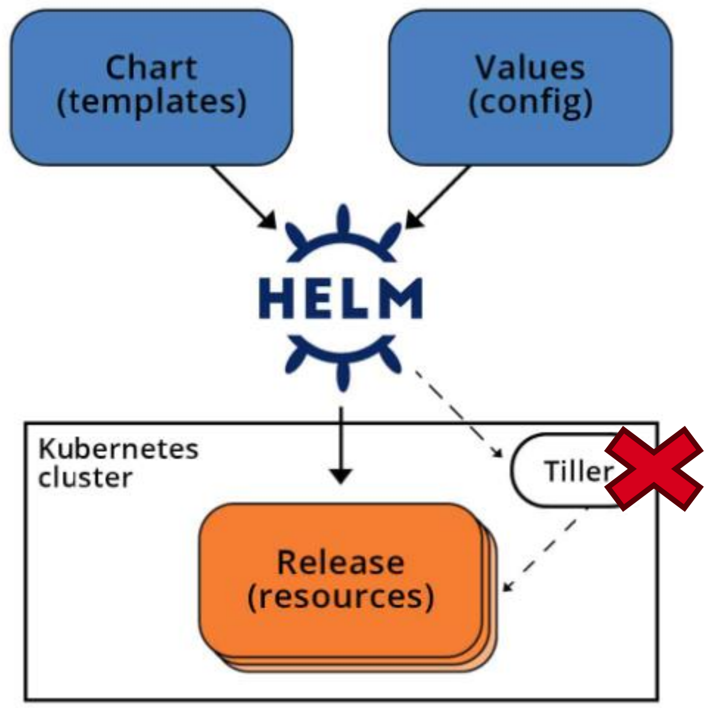
Prometheus and Grafana Intro

Prometheus Operator



Prometheus and Grafana Intro

Install with Helm



Helm is an open-source graduated [CNCF project] originally created by [DeisLabs] as a third-party utility, now known as the [package manager for Kubernetes].

A Helm chart is a set of YAML manifests and templates that describes Kubernetes resources (Deployments, Secrets, CRDs, etc.) and defined configurations needed for the Kubernetes application

Prometheus and Grafana Intro

Install with Helm

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

helm search repo prometheus |egrep "stack|CHART"
###NAME                                              CHART VERSION    APP VERSION        DESCRIPTION
###prometheus-community/kube-prometheus-stack        48.1.1           v0.66.0      kube-prometheus-stack collects Kubernetes
manif...
###prometheus-community/prometheus-stackdriver-exp... 4.3.0           0.13.0       Stackdriver exporter for Prometheus
###grafana/loki-stack                                2.9.10     v2.6.1            Loki: like Prometheus, but for logs.

helm pull prometheus-community/kube-prometheus-stack --version 48.1.1
tar -xzf  kube-prometheus-stack-48.1.1.tgz
cp kube-prometheus-stack/values.yaml ./kube-prometheus-stack-values.yaml
```

Prometheus and Grafana Intro

Install with Helm

Set following parameters in ./kube-prometheus-stack-values.yaml:

```
prometheus:
  prometheusSpec:
    podMonitorSelectorNilUsesHelmValues: false
    serviceMonitorSelectorNilUsesHelmValues: false
    ruleSelectorNilUsesHelmValues: false
```

###the known issue: https://github.com/helm/charts/issues/11310

###Install with following command:

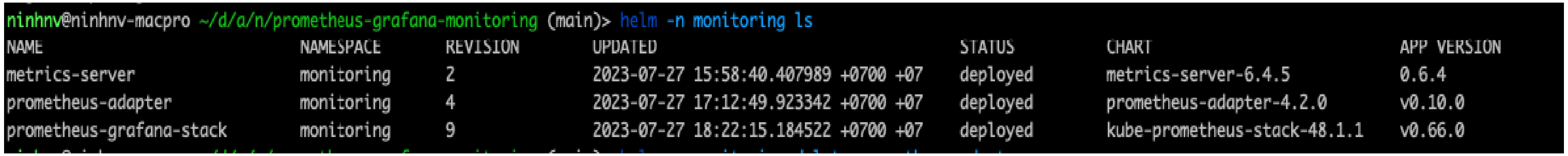
```
helm -n monitoring upgrade prometheus-grafana-stack --install  -f kube-prometheus-stack-values.yaml kube-prometheus-stack
```

###For the next time update the configuration:

```
helm -n monitoring upgrade prometheus-grafana-stack -f kube-prometheus-stack-values.yaml kube-prometheus-stack
```

Prometheus and Grafana Intro

Install with Helm



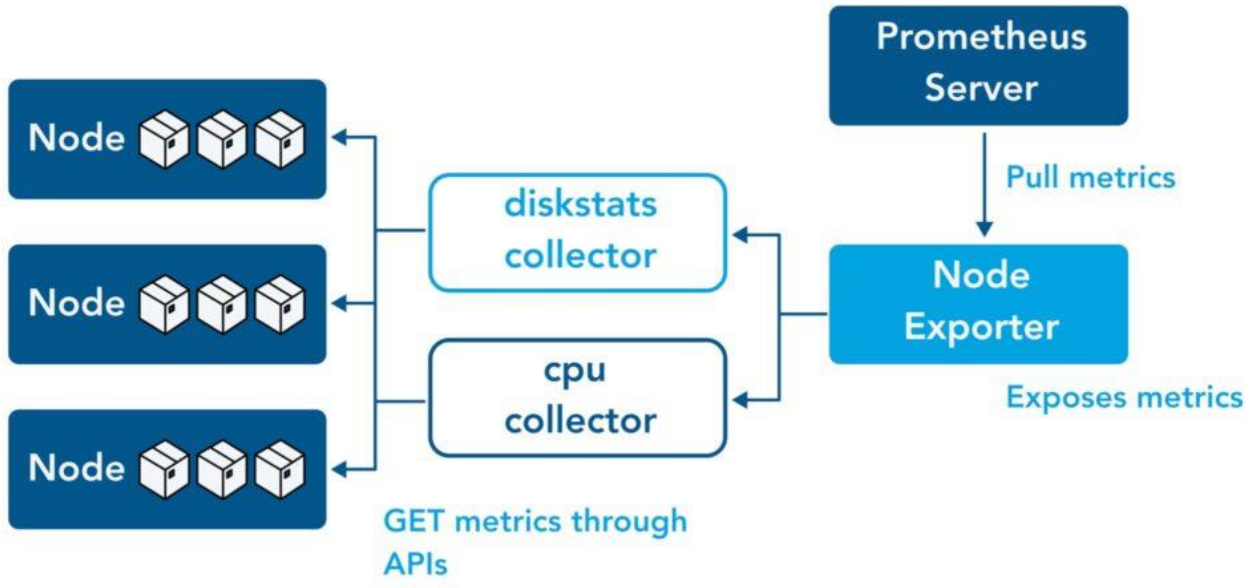
```
ninhnv@ninhnv-macpro ~/d/a/n/prometheus-grafana-monitoring (main)> helm -n monitoring ls
NAME                            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                           APP VERSION
metrics-server                  monitoring      2               2023-07-27 15:58:40.407989 +0700 +07    deployed        metrics-server-6.4.5            0.6.4
prometheus-adapter              monitoring      4               2023-07-27 17:12:49.923342 +0700 +07    deployed        prometheus-adapter-4.2.0        v0.10.0
prometheus-grafana-stack        monitoring      9               2023-07-27 18:22:15.184522 +0700 +07    deployed        kube-prometheus-stack-48.1.1    v0.66.0
```

Prometheus Scrape Config

Prometheus Scrape Config

Prometheus Exporter

A Prometheus Exporter can fetch statistics from an application in the format used by that system (i.e. XML), convert those statistics into metrics that Prometheus can utilize, and then expose them on a Prometheus-friendly URL. There is a vast library of applications that can export metrics from third parties and transform them into Prometheus metrics



Prometheus Scrape Config

scrapeConfig vs ServiceMonitor

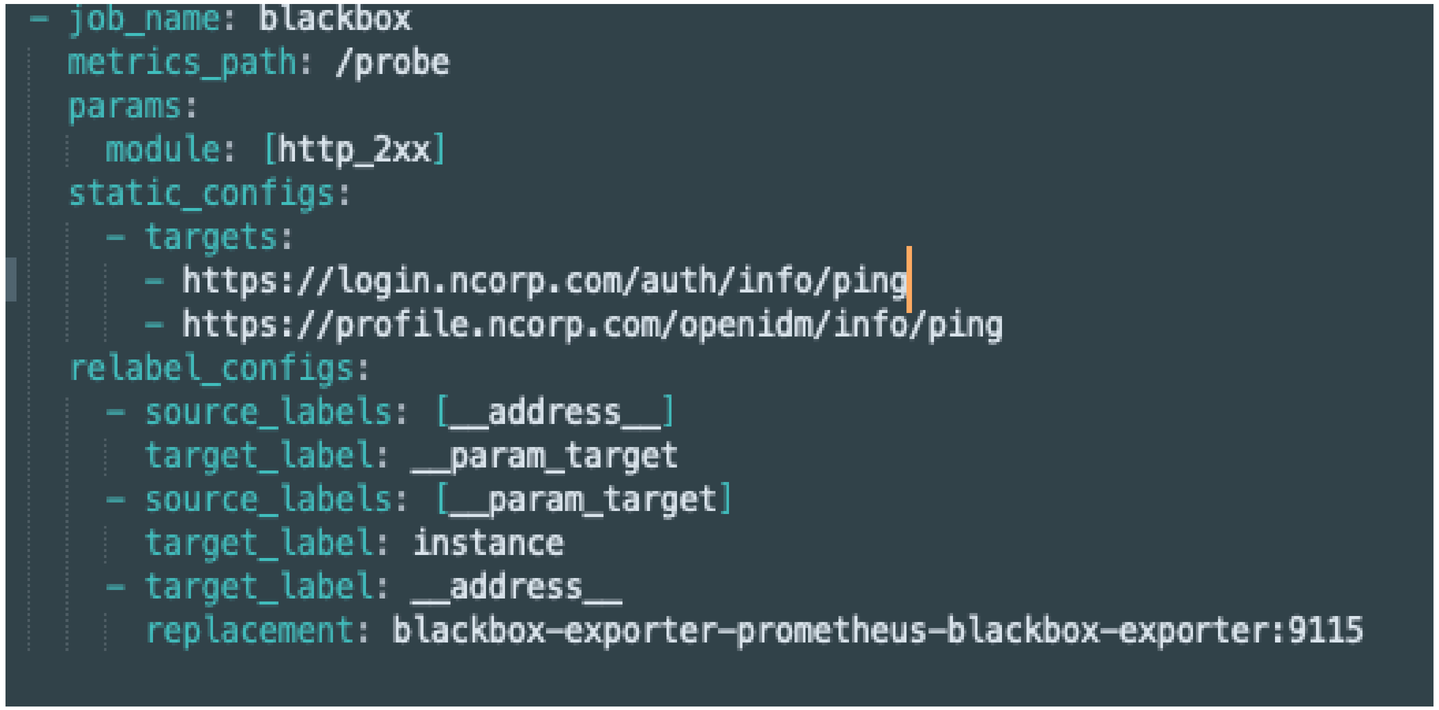
- A **scrape_config** specifies a set of targets and configuration parameters describing how to scrape them.
  In this case, for each target; one scrape configuration block needs to be

- A **ServiceMonitor** lets us create a job entry in scrape_config in an easier Kubernetes-native way. Internally Prometheus Operator translates the configuration from each ServiceMonitor resource to prometheus.yaml's scrape_config section.

- **ServiceMonitor** is suitable if you already have a Service for your pods. However, if in a certain scenario, you don't have it, then **PodMonitor** is the right choice

Prometheus Scrape Config

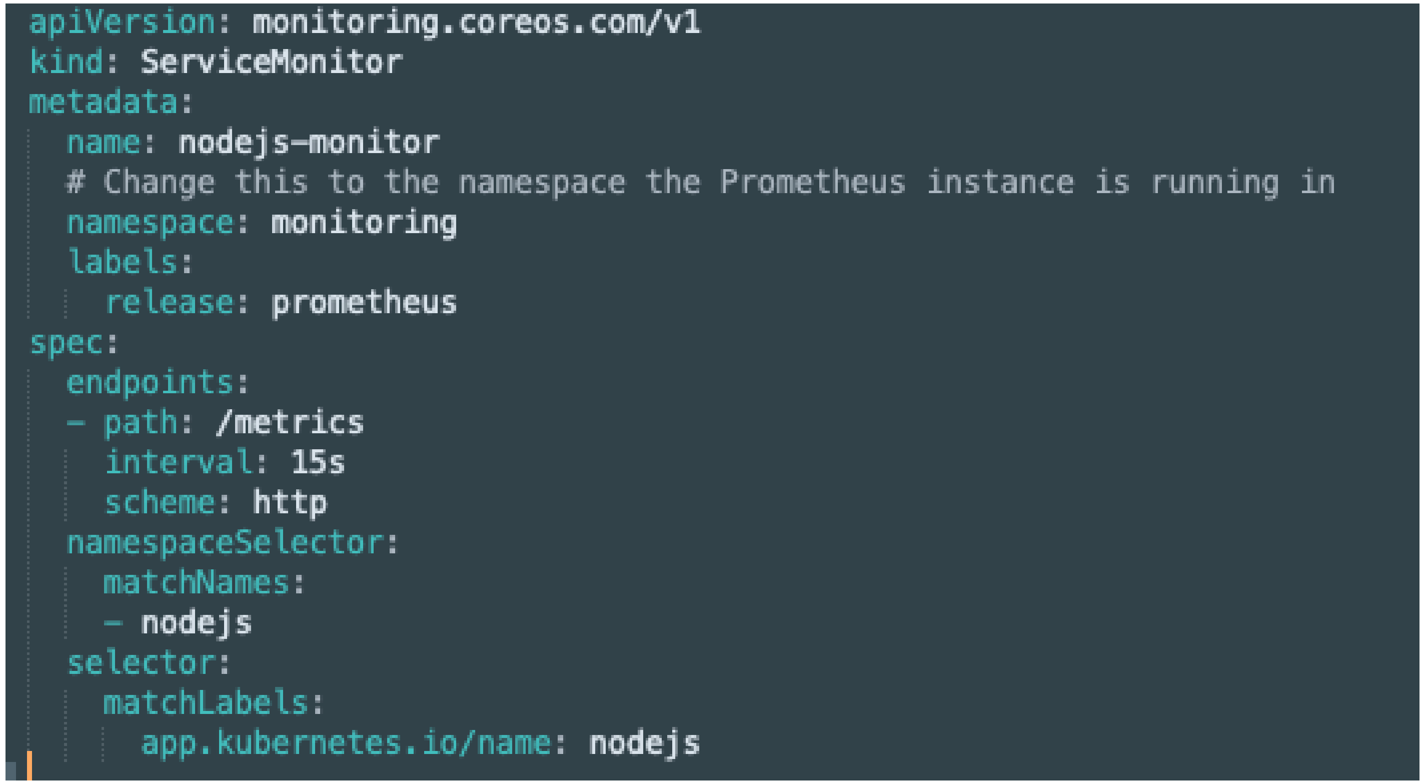
scrapConfig setup



```
- job_name: blackbox
  metrics_path: /probe
  params:
    module: [http_2xx]
  static_configs:
    - targets:
      - https://login.ncorp.com/auth/info/ping
      - https://profile.ncorp.com/openidm/info/ping
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: blackbox-exporter-prometheus-blackbox-exporter:9115
```

Prometheus Scrape Config

serviceMonitor setup

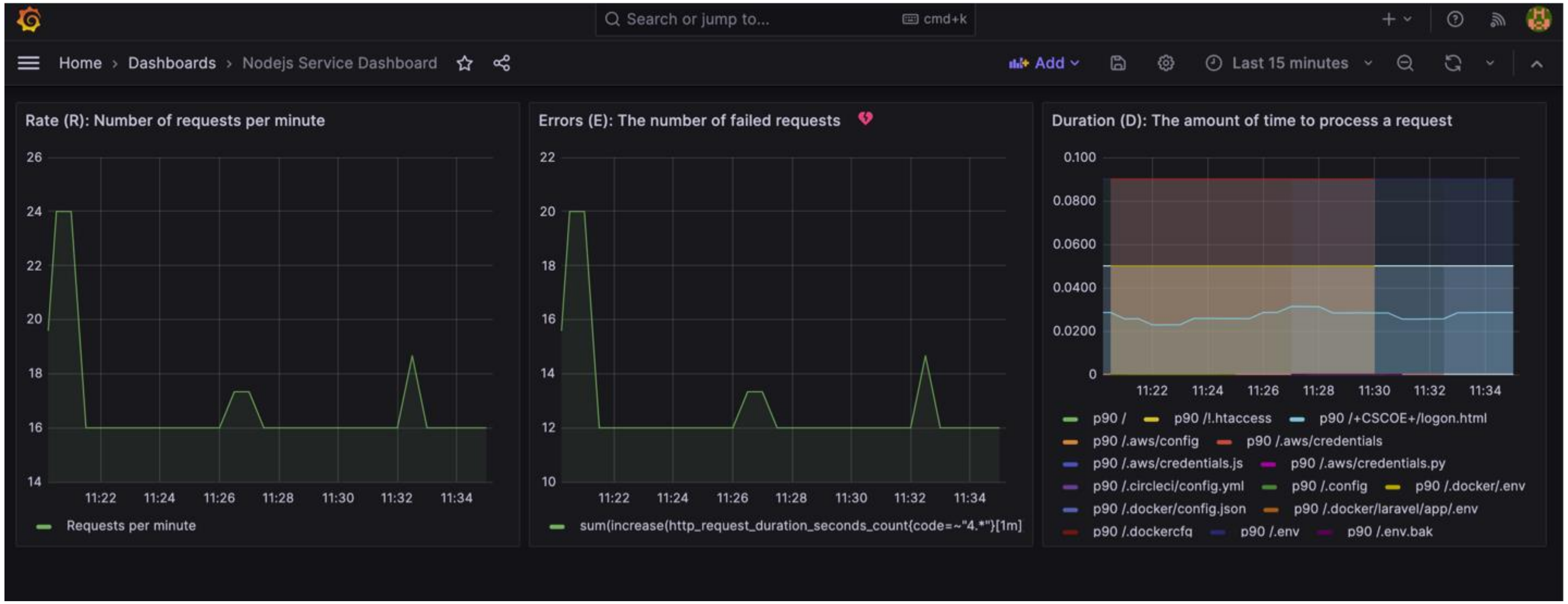


```yaml
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: nodejs-monitor
  # Change this to the namespace the Prometheus instance is running in
  namespace: monitoring
  labels:
    release: prometheus
spec:
  endpoints:
  - path: /metrics
    interval: 15s
    scheme: http
  namespaceSelector:
    matchNames:
    - nodejs
  selector:
    matchLabels:
      app.kubernetes.io/name: nodejs
```

Grafana Dashboard

Grafana Dashboard

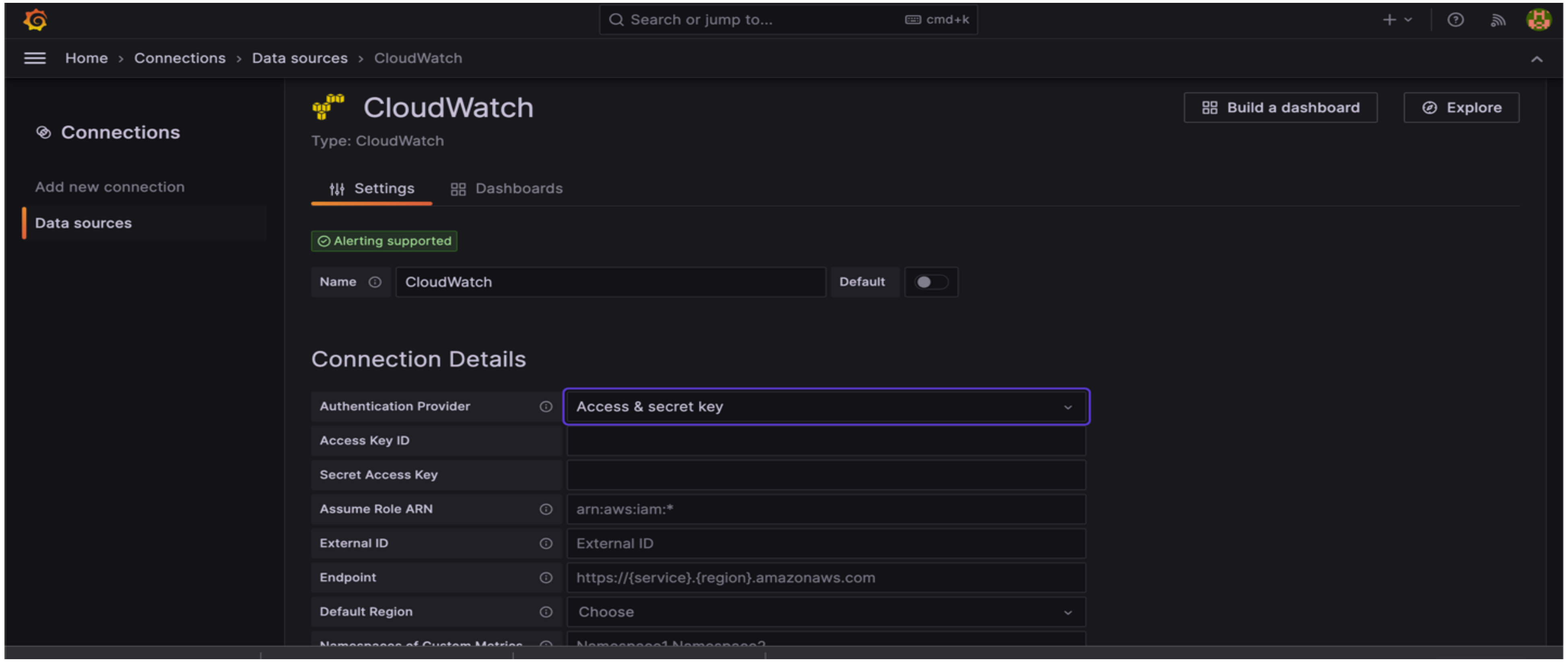
Dashboard setup



```
sum(increase(http_request_duration_seconds_count{code=~"4.*"}[1m]
```

Grafana Dashboard

Cloudwatch - Data Source setup


Home › Connections › Data sources › CloudWatch

Connections
Add new connection
Data sources

CloudWatch
Type: CloudWatch

Build a dashboard | Explore

Settings | Dashboards

Alerting supported

Name: CloudWatch   Default: off

Connection Details

Authentication Provider	Access & secret key
Access Key ID	
Secret Access Key	
Assume Role ARN	arn:aws:iam:*
External ID	External ID
Endpoint	https://{service}.{region}.amazonaws.com
Default Region	Choose

AlertManager vs Grafana Alerts

AlertManager vs Grafana Alerts

Slack Notification with AlertManager

- Setup Webhook URL
- Update Alertmanger config
- Update Prometheus Rules to fire an alert

AlertManager vs Grafana Alerts

Slack Notification with Grafana Alert

- Setup alert rules
- Set Notification policy

Prometheus Adapter

Prometheus Adapter

Setup custom rule for Prometheus Adapter

- Metric server in Kubernetes: 03 types
- Setup custom rule to monitor custom_metric from Application

Prometheus Adapter

HPA Scale based on custom-metric

- Setup HPA with custom_metric
- Trigger load test to demonstrate the Pod scaling

Any Questions…?

Thank you

Nash Tech.