Operadores, Dashboards y Monitorización en Kubernetes

KEEPCODING
Tech School

Dashboards y Monitorización

- En esta sección:
  - Dashboards (UI) para gestión del cluster
  - Operadores en Kubernetes
  - Sistemas de monitorización
  - Kubernetes Observability
    - Prometheus y Grafana
    - Elastic Stack
  - Dashboards para el manejo del cluster.

Dashboards / UI

KEEPCODING
Tech School

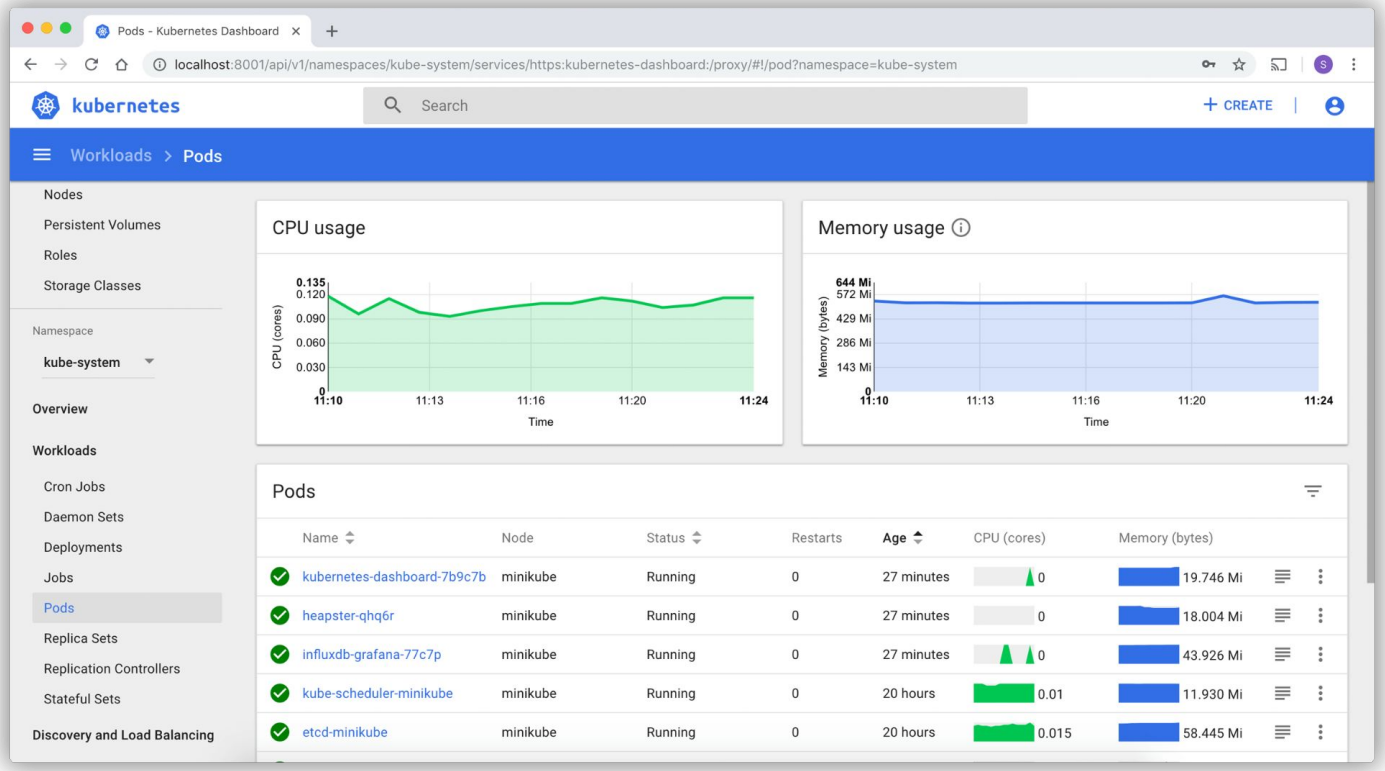
Dashboards para manejar Kubernetes

- Kubernetes Dashboard
- Lens → https://k8slens.dev/
- k9s → https://k9scli.io/
- Kubernetic → https://www.kubernetic.com/

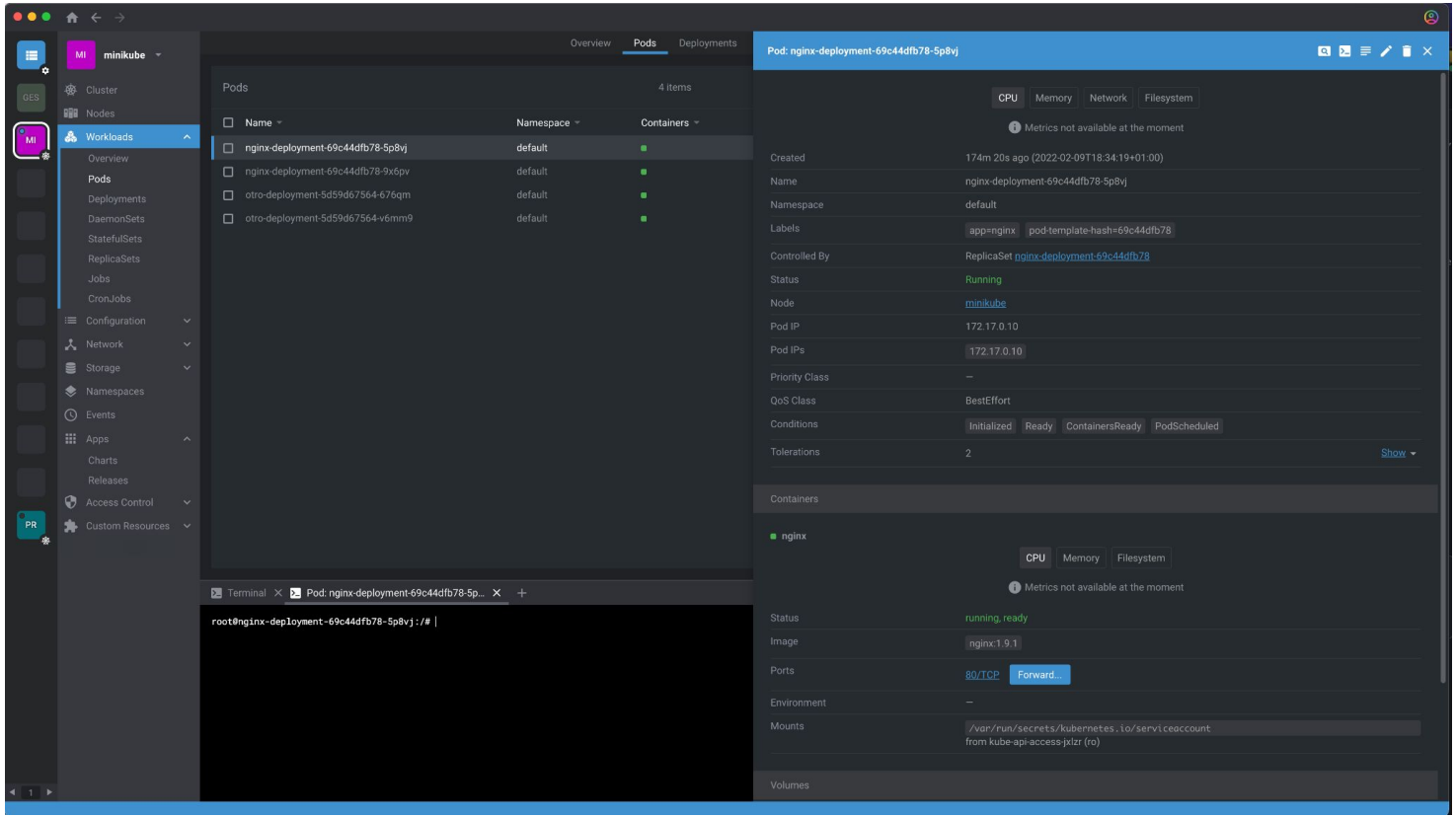
Kubernetes Observability

- Prometheus & Grafana → Para métricas / monitorización.
- Elastic Stack → Para observability (métricas, logs, APM, SIEM, Machine Learning, etc).

Kubernetes Dashboard

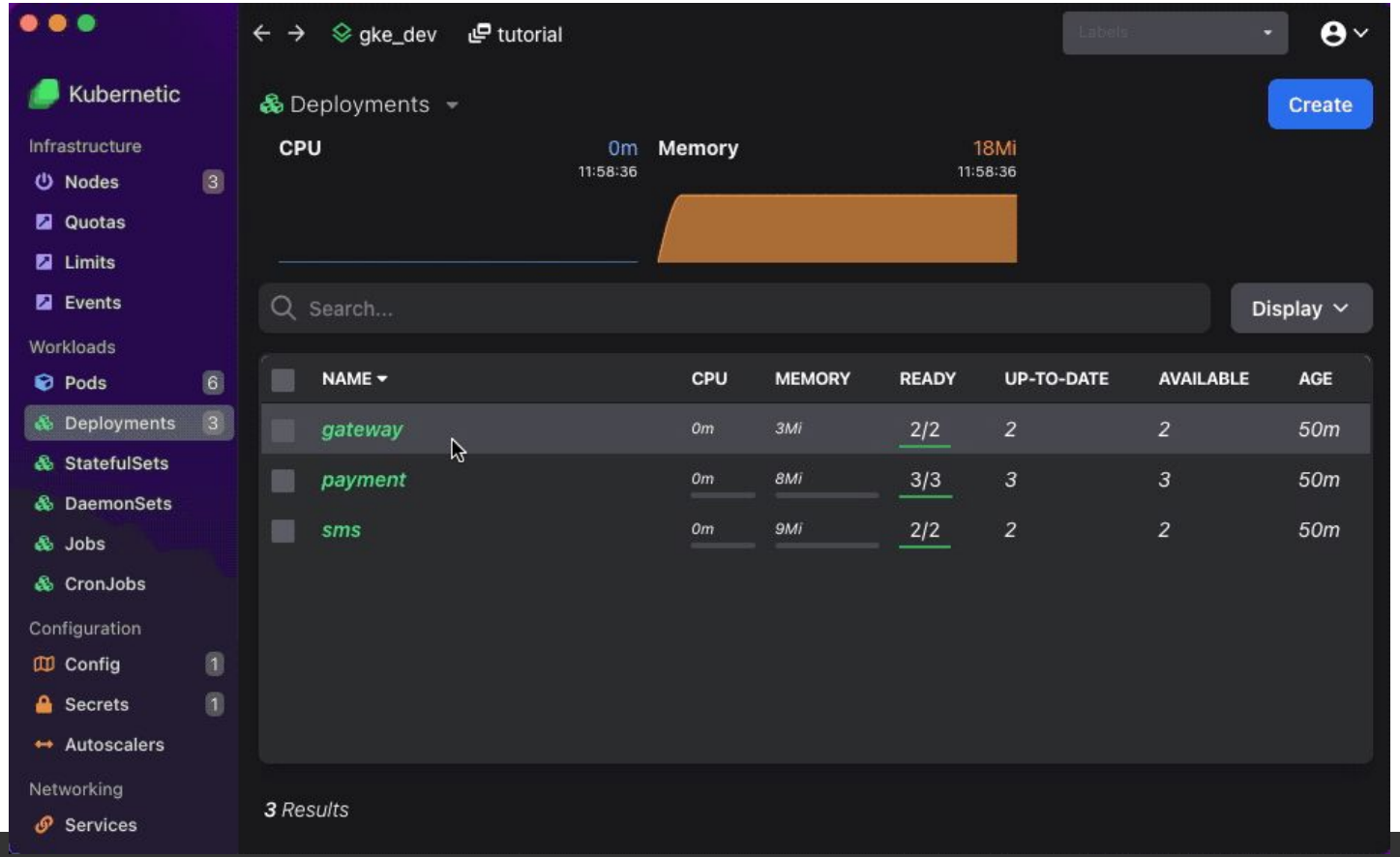


Lens



k9s

Kubernetic



Operadores en Kubernetes

KEEPCODING
Tech School

Operadores en Kubernetes

- ¿Qué es un Operador?
  - Es un controlador específico de aplicaciones que amplía las funciones de Kubernetes para poder crear, configurar, y gestionar las instancias de esas aplicaciones.
  - Utilizarán recursos básicos de Kubernetes pero tienen el conocimiento específico de la aplicación para automatizar todo el ciclo de vida del software que gestiona.

Operadores en Kubernetes

- ¿Por qué son necesarios los operadores?
  - Para manejar aplicaciones con estado que pueden requerir conocimientos específicos que Kubernetes no posee.
- Componentes de un Operador
  - Definición de objetos (CRDs / Custom Resource Definitions)
    - Metadatos que se instalan en el cluster
  - Controlador
    - Software que corre en el cluster (como pod)

Operadores en Kubernetes

- ¿Cual es el valor añadido de usar operadores?
  - El operador cuidará de las aplicaciones mejor de lo que puede hacerlo Kubernetes, ya que conoce cómo han de realizarse las operaciones y el mantenimiento.
  - Un Deployment cuida de sus pods, y un StatefulSet de los suyos pero con unas limitaciones que pueden ser negativas para algunas aplicaciones como bases de datos.

Sistemas de Monitorización

KEEPCODING
Tech School

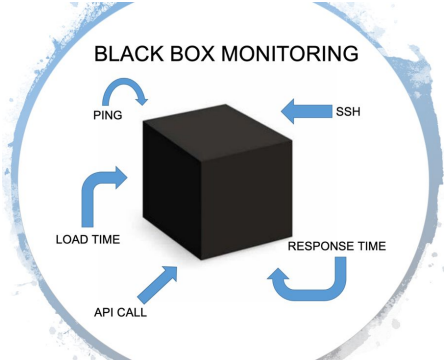
Sistemas de Monitorización

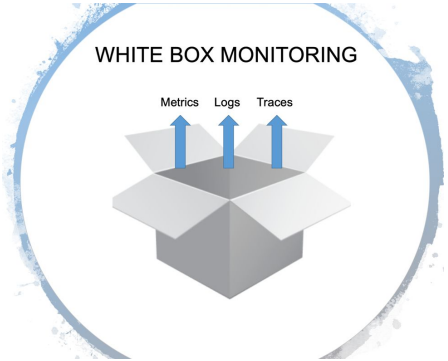
- Monitorización caja blanca y caja negra
- Monitorización mediante HTTP Checks
- Monitorización basada en logs
- Prometheus
- Instrumentar una aplicación con Prometheus

Sistemas de monitorización

"Asegurar que tanto aplicaciones como sistemas funcionan como deberían"

- Sistemas de monitorización manuales o automáticos
- Sistemas de caja negra o caja blanca





Sistemas de caja negra

- Desconocemos cómo funciona el sistema internamente.
- Utilizamos sistemas externos para monitorizar y comprobar que funciona correctamente.
- Establecemos métricas y kpis nosotros de forma externa, no tienen por qué ser las correctas.
- Sistemas de monitorización caja negra conocidos:
  - Nagios
  - Sensu
  - Heartbeat (Elastic Stack) → Uptime application
  - Pingdom (HTTP Checks)
  - Stackdriver (HTTP Checks)
  - Statuscake (HTTP Checks)

HTTP / TCP checks

- Exponer un endpoint (normal o especial) que permita monitorizar:
  - El estado del sistema.
  - Tiempo de respuesta, contenido de respuesta, etc.
- Existen sistemas como servicio o alojados por nosotros mismos.

Sistemas de caja blanca

- Mediante la monitorización de caja blanca son los propios sistemas y aplicaciones los que exponen sus propias métricas que nosotros podemos leer y actuar frente a unos resultados u otros.
- Elementos principales:
  - Métricas (de todo tipo, dependiendo del sistema monitorizado).
  - Logs
  - Application Performance Metrics (APM)

Sistemas basados en métricas

- Sistemas de obtención de Métricas:
  - Requieren un sistema específico y adaptar nuestras aplicaciones.
  - Muy ligeros
  - Sistemas baratos
  - Fácil de correlar con otros sistemas.
  - Ejemplos
    - Prometheus
    - Elastic Stack

Sistemas basados en logs

- Los sistemas basados en logs son sencillos de implementar
- Permiten expresar mucha información de manera muy sencilla.
- Costoso computacionalmente y a nivel de storage (requieren buen dimensionamiento)
- Debemos expresar los logs en algún formato estructurado (JSON) para facilitar la ingesta, procesado y su explotación posterior.
- Debemos implementar algún sistema de recolección de logs (Filebeat / Elastic Stack, Fluentd) y de ingesta centralizada (Elastic Stack, Stackdriver, Cloudwatch).
- Los sistemas as a service tienen buena escalabilidad pero son caros.

Visualización

- Tanto para la recolección de logs como para las métricas o APM necesitamos algún sistema para visualizar y analizar los datos.

- Existen diversos sistemas para visualizar los datos (métricas, logs, datos de APM, …).
  - Kibana (Elastic Stack)
  - Grafana (soporta multitud de backends, como prometheus)
  - Stackdriver
  - Splunk

Prometheus y Grafana

KEEPCODING
Tech School

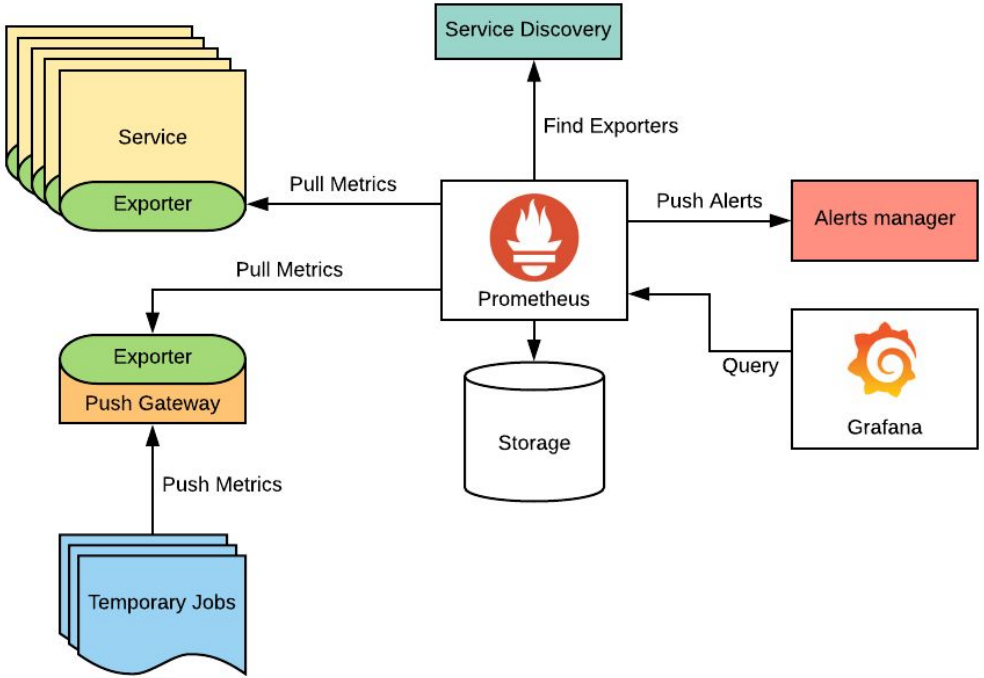
Prometheus

- Existen sistemas especialmente diseñados para extraer y almacenar métricas que posteriormente pueden ser analizadas y visualizadas.
- Uno de los sistemas más conocidos y que es nativo en Kubernetes es **Prometheus**.
  - Inspirado en el sistema de monitorización Borgmon de Google
  - Comenzó en 2012 por ex-googlers en Soundcloud
  - Escrito en Go
  - Hecho público en 2015
  - Proyecto graduado de la CNCF

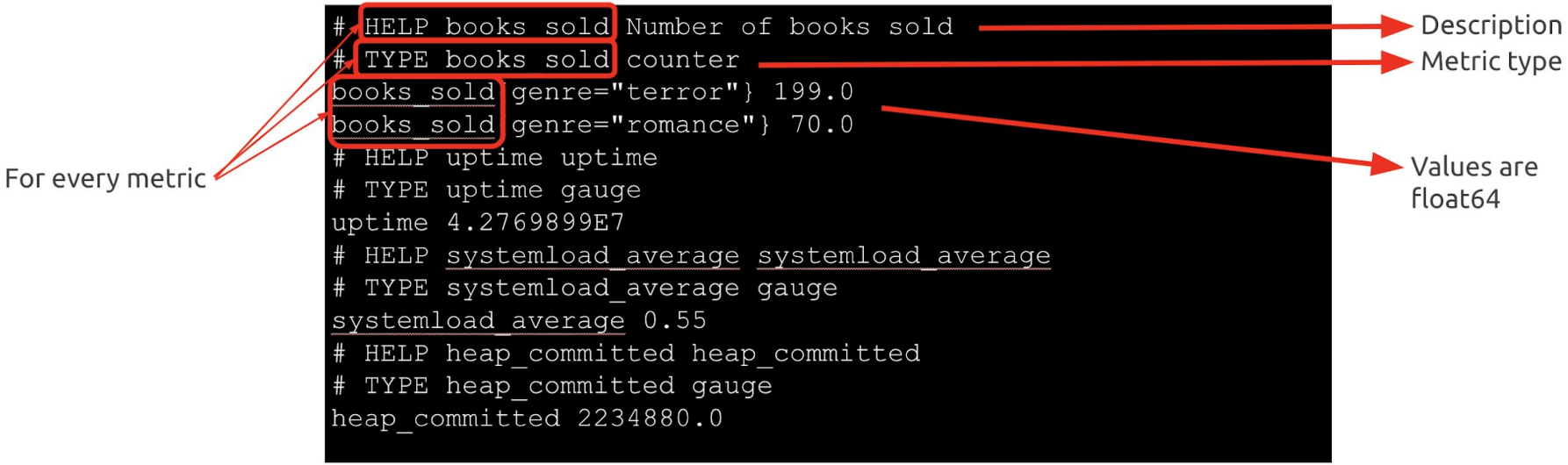
¿Por qué Prometheus?

- Monitorización de caja blanca
- Basado en métricas tanto de sistemas como de negocio
- Robusto y eficaz, alta disponibilidad.
- Eficiente, un solo servidor puede manejar millones de métricas
- Service discovery integrado
- Se integra fácilmente

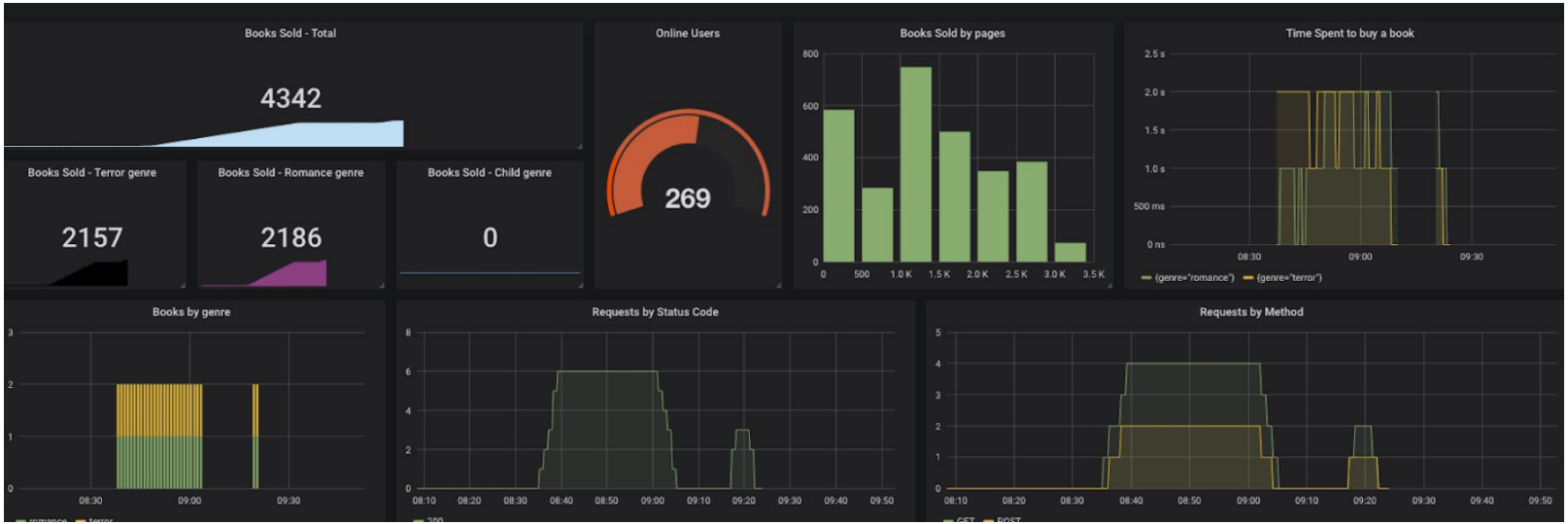
Arquitectura de Prometheus



Lo que lee Prometheus Server

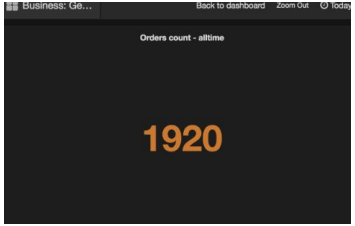


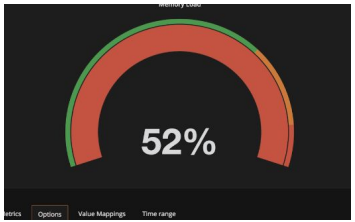
Visualización

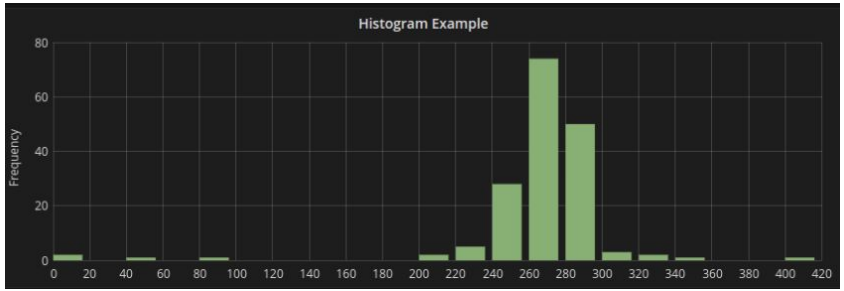


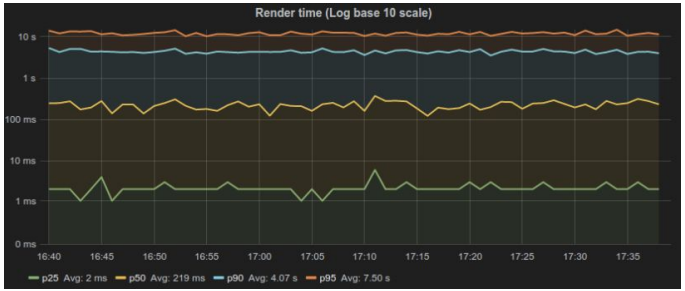
Tipos de métricas

- Contador: Sólo puede incrementarse o resetearse
- Gauge (medida): Se puede incrementar o decrementar
- Histograma: Valores agrupados en buckets
- Summary: Tiempos medios y percentiles









Prometheus Exporters

- Un exporter se encarga de convertir métricas ya existentes internas de la aplicación a métricas de prometheus.

- Existen múltiples exporters:
  - Bases de datos
    - ElasticSearch
    - MySQL
    - PostgreSQL
    - Redis
  - Hardware
  - Sistemas de mensajería
  - HTTP
    - Apache
    - Nginx
  - Almacenamiento
  - APIs
    - Github
    - Dockerhub

Métricas en Kubernetes

- Kubernetes dispone de muchas métricas que podemos exponer con diversos exporters.
  - **cAdvisor**: Se encarga de exponer todo tipo de métricas relacionadas con contenedores
  - **kube-state-metrics**: Se encarga de exponer métricas relacionadas con el clúster y su funcionamiento.
  - **nodeExporter**: Métricas del nodo a nivel de SO.

https://github.com/google/cadvisor

https://github.com/kubernetes/kube-state-metrics


```
sudo docker run \
  --volume=/:/rootfs:ro \
  --volume=/var/run:/var/run:ro \
  --volume=/sys:/sys:ro \
  --volume=/var/lib/docker/:/var/lib/docker:ro \
  --volume=/dev/disk/:/dev/disk:ro \
  --publish=8080:8080 \
  --detach=true \
  --name=cadvisor \
  google/cadvisor:latest
```

Métricas expuestas por kube-state-metrics

- CronJob
- DaemonSet
- Job
- LimitRange
- Node
- PersistentVolume
- PersistentVolumeClaim
- Pod
- Pod Disruption Budget
- ReplicaSet
- ReplicationController
- ResourceQuota
- Service
- StatefulSet
- StorageClass
- Namespace
- Horizontal Pod Autoscaler
- Endpoint
- Secret
- ConfigMap
- Ingress
- CertiricateSigningRequest
- VerticalPodAutoScaler

https://github.com/kubernetes/kube-state-metrics/tree/master/docs

Métricas de aplicación (APM)

- Con Prometheus podemos además obtener métricas personalizadas de nuestras aplicaciones.
- Primero haremos un análisis y después definiremos las métricas.
- Para poder hacerlo instrumentamos nuestra aplicación con la librería de Prometheus que corresponda.
- Expondremos las métricas mediante un endpoint HTTP.
- Clientes disponibles
  - Go
  - Java, Scala
  - Python
  - Ruby
  - No oficiales: bash, PHP, node.js, etc

Demostración Instalación Prometheus

- https://prometheus-operator.dev/docs/getting-started/installation/#install-using-helm-chart

Kubernetes Observability con Elastic Stack

KEEPCODING
Tech School

Elastic Stack

- Componentes:
  - Beats / Elastic Agent:
    - Filebeat (logs)
    - Metricbeat (métricas) prometheus
    - Heartbeat (uptime)
    - Packetbeat (tráfico de red)
  - APM
  - Elasticsearch
  - Kibana

Elastic Stack

- ¿Qué ofrece el Stack de Elastic?
  - Full observability integradas en un único sistema.
  - Machine Learning, Elastic Security (SIEM)
  - Correlar logs, métricas y performance de aplicaciones.

KEEPCODING

Tech School

Madrid | Barcelona | Bogotá