

Monitorización de Rendimiento con Prometheus para Servidores

Introducción

En este archivo encontrarás consultas útiles para monitorizar el rendimiento de servidores utilizando Prometheus. Las consultas proporcionadas te permitirán obtener métricas relacionadas con el uso de la CPU, GPU, red, disco y memoria RAM. Estas métricas son fundamentales para identificar cuellos de botella, optimizar recursos y mantener la salud de tu infraestructura.

Consultas

1. Uso de la CPU

Consulta para obtener el uso de la CPU por cada core en diferentes servidores.

```
sum by (mode, instance) (irate(node_cpu_seconds_total{mode="idle"}[5m]))
```

Explicación:

Esta consulta calcula el tiempo de CPU utilizado por cada core en los últimos 5 minutos, agrupado por instance que representa cada servidor y mode que indica el estado de la CPU.

2. Uso de la GPU

Consulta para monitorizar el uso de la GPU en diferentes servidores con GPU NVIDIA.

```
nvidia_gpu_duty_cycle
```

Explicación:

`nvidia_gpu_duty_cycle` es una métrica específica para GPUs NVIDIA que indica el porcentaje de tiempo que la GPU ha estado activa durante el último período de muestreo.

3. Uso de la red

Consulta para obtener el uso de la red por interfaz de red en diferentes servidores.

```
sum by (instance, device) (rate node_network_receive_bytes_total[5m]))
```

Explicación:

Esta consulta calcula la tasa de bytes recibidos por interfaz de red en los últimos 5 minutos. `instance` te permite distinguir de qué servidor provienen los datos y `device` indica la interfaz de red específica.

4. Uso del disco

Consulta para monitorizar el uso del disco en diferentes servidores.

```
node_filesystem_free_bytes{instance=~"server1|server2", mountpoint="/"}
```

Explicación:

`node_filesystem_free_bytes` muestra el espacio libre en bytes en el sistema de archivos. `instance` permite filtrar los datos por servidor utilizando expresiones regulares (en este ejemplo, "server1" o "server2").

5. Uso de la memoria RAM

Consulta para obtener el uso de la memoria RAM en diferentes servidores.

```
node_memory_Active_bytes{instance=~"server1|server2"}
```

Explicación:

`node_memory_Active_bytes` muestra la cantidad de memoria RAM activa en bytes en el servidor. `instance` permite filtrar los datos por servidor utilizando expresiones regulares.

6. Uso de la CPU por proceso

Consulta para obtener el uso de CPU por proceso en diferentes servidores.

```
sum by (instance, process) (rate(process_cpu_seconds_total[5m]))
```

Explicación:

`process_cpu_seconds_total` muestra el tiempo total de CPU consumido por procesos individuales en segundos desde que se iniciaron. `instance` te permite distinguir de qué servidor provienen los datos y `process` indica el nombre del proceso.

7. Uso de la GPU por proceso

Consulta para monitorizar el uso de la GPU por proceso en diferentes servidores con GPU NVIDIA.

```
nvidia_gpu_process_memory_bytes{instance=~"server1|server2",  
process_name="my_process"}
```

Explicación:

`nvidia_gpu_process_memory_bytes` muestra la cantidad de memoria utilizada por un proceso en la GPU. `instance` permite filtrar los datos por servidor utilizando expresiones regulares y `process_name` ajusta al proceso específico que deseas monitorear.

8. Latencia de red

Consulta para monitorizar la latencia de red en diferentes servidores.

```
probe_http_duration_seconds{job="blackbox", instance=~"server1|server2",  
instance="www.example.com"}
```

Explicación:

`probe_http_duration_seconds` muestra la duración de las solicitudes HTTP realizadas por el job de blackbox exporter a la instancia especificada. `instance` permite filtrar los datos por servidor utilizando expresiones regulares.

9. Uso del disco por volumen

Consulta para monitorizar el uso de disco por volumen en diferentes servidores.

```
node_filesystem_free_bytes{instance=~"server1|server2", mountpoint="/mnt/data"}
```

Explicación:

`node_filesystem_free_bytes` muestra el espacio libre en bytes en el sistema de archivos. `instance` permite filtrar los datos por servidor utilizando expresiones regulares y `mountpoint` ajusta al punto de montaje del volumen que deseas monitorear.

10. Uso de la memoria RAM por proceso

Consulta para obtener el uso de memoria RAM por proceso en diferentes servidores.

```
process_resident_memory_bytes{instance=~"server1|server2", job="my_job"}
```

Explicación:

`process_resident_memory_bytes` muestra la cantidad de memoria física utilizada por un proceso. `instance` permite filtrar los datos por servidor utilizando expresiones regulares y `job` ajusta al trabajo específico que estás monitoreando.

Conclusión

Estas consultas te proporcionan una base sólida para comenzar a monitorizar y analizar el rendimiento de servidores utilizando Prometheus. Asegúrate de ajustar las consultas según tus necesidades específicas y las métricas disponibles en tu entorno.