



MEUS CURSOS



CALENDÁRIO



CERTIFICADOS



SUPORTE



LEARNINGFLIX

[Página inicial](#) / ORQUESTRAÇÃO DE CONTAINERS COM KUBERNETES - 18/03/2024

Voltar para o curso

Iniciado em	Thursday, 21 Mar 2024, 19:20
Estado	Finalizada
Concluída em	Thursday, 21 Mar 2024, 19:37
Tempo empregado	17 minutos 20 segundos
Notas	5,00/5,00
Avaliar	10,00 de um máximo de 10,00(100%)



Questão **1**

Correto

Atingiu 1,00 de 1,00

Considerando um pod com múltiplos containers (i.e. um multi-container pod), qual flag pode ser passada ao comando `kubectl logs` para especificar o container cujos registros gostaríamos de visualizar?

- ☐ a. -f
- ☐ b. -s
- ☐ c. -p
- ☒ d. -c



Sua resposta está correta.

Como indicado na documentação do `kubectl logs` (<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#logs>), as opções `--container` ou `-c` podem ser usadas para este fim.

A resposta correta é:

-c



Questão **2**

Correto

Atingiu 1,00 de
1,00

"Containers que executam conjuntamente com o container 'principal' em um pod, e estem a funcionalidade deste sem modificá-lo. Frequentemente utilizado em tarefas como envio de logs para servidores externos, monitoramento de rede ou sincronização de código do container principal". A definição apresentada encaixa-se melhor com qual tipo de container pattern?

- ☒ a. Sidecar
- ☐ b. Init
- ☐ c. Ambassador
- ☐ d. Sidekick



Sua resposta está correta.

Conforme explicado em diversos artigos (como este <https://medium.com/bb-tutorials-and-thoughts/kubernetes-learn-sidecar-container-pattern-6d8c21f873d>) e na documentação oficial do Kubernetes, o Sidecar Container Pattern é um paradigma para uso de multi-container pods que cumpre os objetivos apresentados na definição. Do ponto de vista de troubleshooting e manutenção, o uso de um sidecar container como coletor e agregador de logs é particularmente interessante.

A resposta correta é:

Sidecar

Questão **3**

Correto

Atingiu 1,00 de 1,00

O metrics-server é uma solução simples e bastante leve para coleta de uso de recursos em um cluster Kubernetes, expondo esses dados através da Metrics API. Das características mostradas abaixo, qual não se encaixa com esse software?

- ☒ a. Fonte precisa de métricas, ideal para soluções de monitoramento ✓
- ☐ b. Ideal para autoscaling horizontal baseado em CPU e memória
- ☐ c. Suporta até 5000 cluster nodes
- ☐ d. Solução leve e autoescalável, que coleta métricas a cada 15 segundos por padrão

Sua resposta está correta.

Como indicado em sua página oficial (<https://github.com/kubernetes-sigs/metrics-server>), o metrics-server não substitui soluções de monitoramento completas -- como o Prometheus -- e presta-se tipicamente à implantação de autoscaling horizontal e vertical em clusters Kubernetes.

A resposta correta é:

Fonte precisa de métricas, ideal para soluções de monitoramento



Questão **4**

Correto

Atingiu 1,00 de
1,00

O Kubernetes dispõe de diversas soluções populares para monitoramento do cluster. Qual das opções abaixo NÃO representa uma dessas opções?

- ☐ a. Prometheus
- ☒ b. Harbor
- ☐ c. Weave Scope
- ☐ d.
- ☐ e. Jaeger



Sua resposta está correta.

Diferentemente das demais opções, o Harbor é uma solução para implantação de Container Registries no Kubernetes, não tendo qualquer relação com funções de monitoramento.

A resposta correta é:

Harbor



Questão **5**

Correto

Atingiu 1,00 de 1,00

Qual seria uma alternativa viável para visualizar logs de containers caso o kube-apiserver encontre-se indisponível em seu cluster Kubernetes?

- ☐ a. Reiniciar o computador
- ☒ b. Utilizar o comando ``docker logs``
- ☐ c. Visualizar registros no `/var/log`
- ☐ d. Reinstalar o cluster do zero



Sua resposta está correta.

Como indicado na parte teórico deste módulo, os containers continuam em operação mesmo com a indisponibilidade do kube-apiserver (afinal, o kubelet é responsável por levantar esses containers, juntamente com o container runtime em uso no cluster). Assim, comandos que permitem a visualização independente de registros de containers, como o ``docker logs``, funcionam e devem ser utilizados para fins de debugging.

A resposta correta é:

Utilizar o comando ``docker logs``

[< Voltar](#)[< Tarefa 4](#)[Seguir para...](#)[Conteúdo do Módulo >](#)

