

Nome: Número:

ENGENHARIA INFORMÁTICA

Aplicações e Serviços de Computação em Nuvem

Teste

07 de Janeiro de 2023

Duração: 1h30min

1 Durante o semestre foram utilizados diferentes componentes de software, tais como o *Swap*, *Ghost*, *MySQL*, e *Redis*, que podiam ser configurados e instalados seguindo uma arquitetura distribuída. Indique **uma vantagem e um desafio** a ter em conta quando se pretende seguir uma instalação distribuída para um dado componente de software. **Justifique** a sua resposta.

Vantagem de uma Instalação Distribuída:

Escalabilidade e Desempenho:

- **Justificativa:** Uma instalação distribuída permite que o componente de software, como o *MySQL* ou o *Redis*, seja escalado horizontalmente, distribuindo a carga de trabalho entre vários nós. Isso melhora o desempenho do sistema, permitindo que ele lidere grandes volumes de dados ou requisições de maneira eficiente. No caso do *Redis*, por exemplo, a replicação e o particionamento de dados em múltiplos nós podem aumentar a capacidade de leitura e a resiliência do sistema.

Desafio de uma Instalação Distribuída:

Complexidade na Gestão e Manutenção:

- **Justificativa:** A instalação distribuída exige a coordenação e o gerenciamento de múltiplos nós, o que pode se tornar complexo em termos de configuração, monitoramento e resolução de falhas. Por exemplo, configurar corretamente a replicação do *MySQL* ou garantir a consistência de dados no *Redis* em um ambiente distribuído pode ser desafiador. Além disso, a manutenção do sistema, como atualizações ou backups, precisa ser cuidadosamente planejada para evitar interrupções ou perda de dados, exigindo ferramentas e práticas adequadas.

O que é uma instalação distribuída?

- Instalação distribuída significa distribuir o software ou sistema em vários servidores ou máquinas (nós) ao invés de usar apenas uma máquina. Cada nó pode ter uma parte do trabalho, o que ajuda a melhorar a performance e resiliência do sistema.

Vantagem:

- O que você ganha ao usar uma instalação distribuída. Por exemplo, um melhor desempenho ou escalabilidade.

Desafio:

- O que pode complicar ou dificultar a instalação e manutenção do sistema em uma arquitetura distribuída. Por exemplo, gerenciar múltiplos servidores ou manter a consistência de dados entre os nós.

Exemplo de Resposta:

- **Vantagem:** Melhora o desempenho e a escalabilidade, permitindo que o sistema lide com mais dados ou mais requisições, distribuindo a carga entre múltiplos servidores.
- **Desafio:** Aumenta a complexidade de manutenção, pois exige a configuração e coordenação entre os vários servidores, o que pode ser mais difícil de gerenciar.

Nome: Número:

ENGENHARIA INFORMÁTICA

Aplicações e Serviços de Computação em Nuvem

Teste

07 de Janeiro de 2023

Duração: 1h30min

2 Embora ambas sejam consideradas tecnologias de virtualização, máquinas virtuais e *containers* adequam-se a diferentes tipos de aplicações e serviços. Indique **um** tipo de aplicação ou serviço onde é mais apropriada a utilização de *containers*. **Justifique** a sua resposta.

Tipo de Aplicação/Serviço: Aplicações Web (por exemplo, APIs REST, microservices)

Justificativa:

Os containers são ideais para aplicações baseadas em microserviços ou aplicações web devido às seguintes características:

1. **Leveza e Rápida Inicialização:** Containers compartilham o mesmo kernel do sistema operacional host, o que os torna **mais leves e rápidos** em comparação com máquinas virtuais. Isso é uma vantagem quando se trata de aplicações que precisam ser **iniciadas rapidamente** e escaladas dinamicamente, como APIs REST ou serviços web.
2. **Escalabilidade e Portabilidade:** Em uma arquitetura de microserviços, cada componente da aplicação (como diferentes serviços ou APIs) pode ser executado em **containers separados**. Isso permite que a aplicação seja **escalada horizontalmente** de maneira eficiente, criando ou removendo containers conforme a demanda. Além disso, containers são **portáveis** entre diferentes ambientes, facilitando o desenvolvimento, teste e produção em várias plataformas.
3. **Isolamento e Flexibilidade:** Embora containers compartilhem o mesmo kernel, eles oferecem um **alto nível de isolamento**, garantindo que diferentes partes de uma aplicação possam ser executadas de forma independente e sem interferir umas nas outras. Isso é especialmente útil quando diferentes microserviços ou APIs necessitam de **dependências específicas**, mas ainda assim precisam se comunicar eficientemente.
4. **Eficiência em Recursos:** Containers consomem significativamente menos **recursos de sistema** do que máquinas virtuais. Em ambientes de nuvem ou servidores com recursos limitados, isso pode resultar em **melhor utilização dos recursos** e menor custo operacional, ideal para aplicações que precisam ser executadas em grande escala.

Exemplo:

- Uma aplicação web composta por múltiplos microserviços (autenticação, pagamento, catálogo de produtos) pode ser facilmente dividida em containers, com cada microserviço rodando em seu próprio container. Isso facilita a **atualização, escalabilidade e gerenciamento** da aplicação sem afetar os outros serviços.

Conclusão:

Os containers são mais apropriados para aplicações web e microserviços devido à sua **eficiência, escalabilidade e rapidez** na inicialização e execução, além da **portabilidade** que permite uma implementação consistente em diferentes ambientes.

Aplicações e Serviços de Computação em Nuvem

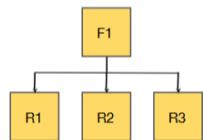
Teste

07 de Janeiro de 2023

Duração: 1h30min

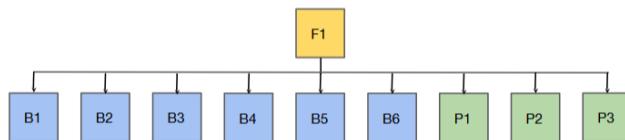
3 Imagine que tem de escolher as melhores configurações para um sistema de armazenamento que tem como principais propósitos garantir **alta disponibilidade** e, simultaneamente, **reduzir o espaço ocupado** pelos dados persistidos. Das funcionalidades tipicamente suportadas por sistemas de armazenamento que estudou, quais é que sugeria? **Justifique** a sua resposta.

- **Erasure-Codes:** data is broken into fragments, with redundant information, that are then spread across several servers for availability and load balancing purposes



- ### – **Replication:** Exact Replicas (of F1)

- E.g., Replication factor = 3
 - ▶ Tolerates 2 failures
 - ▶ 3X storage overhead



- Erasure-codes:** Original data (F1) is divided into k Blocks and m Parity blocks

- E.g., Reed-Solomon ($k = 6$, $m = 3$)

 - ▶ Tolerates 3 failures
 - ▶ $1.5 \times$ storage overhead

O que são Erasure Codes?

Os Erasure Codes são uma técnica de codificação que divide os dados originais em múltiplos fragmentos, gerando fragmentos adicionais (redundantes) a partir dos dados originais. Esses fragmentos são espalhados por diferentes servidores ou nós. Mesmo que alguns desses fragmentos se percam devido a falhas de hardware, os dados originais podem ser recuperados a partir dos fragmentos restantes, utilizando a redundância e a informação adicional dos fragmentos redundantes.

Aplicações e Serviços de Computação em Nuvem

Teste

07 de Janeiro de 2023

Duração: 1h30min

4 Os engenheiros informáticos responsáveis por gerir aplicações em ambientes *Kubernetes* recorrem muitas vezes a ferramentas de monitorização.

4.1 Indique **uma das razões** pela qual as *Kubernetes* são frequentemente combinadas com ferramentas de monitorização. **Justifique** a sua resposta.

4.2 Indique **uma vantagem** de, ao ambiente com *Kubernetes* e monitorização, acrescentar também uma ferramenta de avaliação experimental (*benchmark*). **Justifique** a sua resposta.

4.1 Razão pela qual as *Kubernetes* são frequentemente combinadas com ferramentas de monitorização:

Razão: Monitoramento de desempenho e saúde dos pods e serviços em tempo real.

- Justificativa: O *Kubernetes* gerencia ambientes complexos compostos por múltiplos **pods**, **containers**, **nós** e **serviços**. O monitoramento é essencial para garantir que as aplicações e recursos do sistema estejam funcionando corretamente. Ferramentas de monitorização, como **Prometheus**, **Grafana** ou **Datadog**, permitem monitorar em tempo real a saúde dos pods, o desempenho das aplicações e o uso de recursos (como CPU, memória e disco). Com isso, é possível detectar gargalos de desempenho, falhas de containers ou problemas de escalabilidade, garantindo que o ambiente *Kubernetes* esteja sempre otimizado e saudável.
- Sem monitoramento adequado, seria difícil identificar rapidamente problemas como **recursos sobrecarregados**, **falhas de rede** ou **consumo excessivo de recursos**, que podem afetar a disponibilidade e a eficiência das aplicações em execução no *Kubernetes*.

4.2 Vantagem de adicionar uma ferramenta de avaliação experimental (*benchmark*) ao ambiente com *Kubernetes* e monitorização:

Vantagem: Avaliação do desempenho sob carga e otimização de recursos.

- Justificativa: Ao adicionar uma ferramenta de **benchmark** (como **Apache JMeter**, **K6** ou **wrk**) ao ambiente *Kubernetes* com monitorização, é possível realizar **testes de carga** e **estresse** para avaliar o desempenho da aplicação sob diferentes condições de tráfego. Isso ajuda a entender como a aplicação e o ambiente *Kubernetes* se comportam quando expostos a alto volume de requisições, picos de tráfego ou condições extremas.
- A ferramenta de **benchmarking** permite **identificar limites de desempenho** e pontos fracos no sistema, como **gargalos em recursos** (CPU, memória, rede) ou **latência excessiva**. Com essa informação, a equipe de engenharia pode **ajustar a configuração do Kubernetes**, como a alocação de recursos, a **autoescalabilidade** ou o **balanceamento de carga**, melhorando a eficiência e robustez do sistema. Além disso, o **benchmarking** ajuda a **validar** se as modificações no sistema resultam em melhorias de desempenho, com **dados objetivos** sobre o impacto das mudanças.

Conclusão:

- Monitorização em *Kubernetes* é fundamental para garantir o bom funcionamento e a saúde do ambiente, permitindo a detecção precoce de problemas.
- A adição de uma ferramenta de **benchmarking** oferece uma avaliação prática do desempenho sob carga, permitindo **otimizações** mais eficientes e a **melhoria contínua** do sistema.