PSPD – Programação p Sistemas Paralelos e Distribuídos, Prof.: Fernando W. Cruz

Projeto de pesquisa: Monitoramento/observabilidade de aplicações em clusters K8S

1. Objetivos:

O Kubernetes (https://kubernetes.io) é uma plataforma interessante para o desenvolvimento e disponibilização de aplicações conteinerizadas por conta da sua flexibilidade no gerenciamento da infraestrutura provida para os serviços instanciados. O objetivo deste projeto é explorar as estratégias de monitoramento e observabilidade de aplicações baseadas em microserviços em ambiente kubernetes, com foco na métrica de desempenho.

2. Requisitos para alcançar o objetivo proposto

Para atender ao que foi proposto, os alunos devem *(i)* escolher uma aplicação baseada em microserviços, *(ii)* preparar o *framework* kubernetes em modo cluster, *(iii)* realizar testes de carga baseados em cenários previamente desenhados.

Sobre a aplicação baseada em microserviços

A aplicação baseada em microserviços deve seguir a especificação feita em atividade extraclasse realizada pelos alunos previamente, cujos módulos colaborativos estão explicitados na Figura 1 (vide material disponível no Moodle para relembrar os detalhes solicitados para a aplicação).

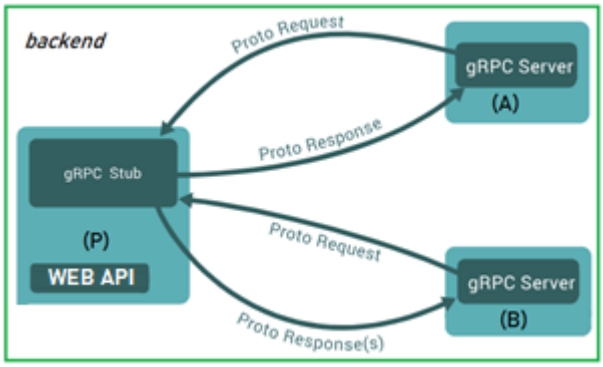

Figura 1 – Arquitetura da aplicação baseada em microserviços

Ou seja, a aplicação deve ser baseada em microserviços gRPC e composta pelos módulos P, A e B que trabalham em conjunto para atender as demandas do usuário, da seguinte forma:

1. As requisições dos usuários chegam à aplicação pelo módulo WEB API.
2. Em seguida, o processamento acontece pela interação colaborativa de P com os microserviços A e B ilustrados na Figura 1.
3. Ao final, a aplicação gera um resultado que é fruto da consolidação das interações gRPC, baseado na requisição feita.

Observações:

- Embora a aplicação a ser entregue deva ser a mesma que foi desenvolvida no trabalho extraclasse, eventuais alterações necessárias para facilitar a observabilidade e monitoramento da aplicação são admitidos, desde que estejam documentados no relatório de entrega.
- Alternativamente os alunos podem substituir a aplicação solicitada por uma outra, desde que sejam preservadas as características apresentadas na Figura 1.

Sobre a preparação do *framework* kubernetes, da ferramenta de monitoramento e dos testes de carga

Para comportar a aplicação citada no item anterior, deve-se estruturar uma instalação kubernetes em modo *cluster*, composto por um nó mestre (plano de controle) e pelo menos dois nós escravos (*worker nodes*), incluindo interface web de monitoramento do cluster e recursos de *autoscaling*, conforme ilustrado na parte direita da Figura 2.

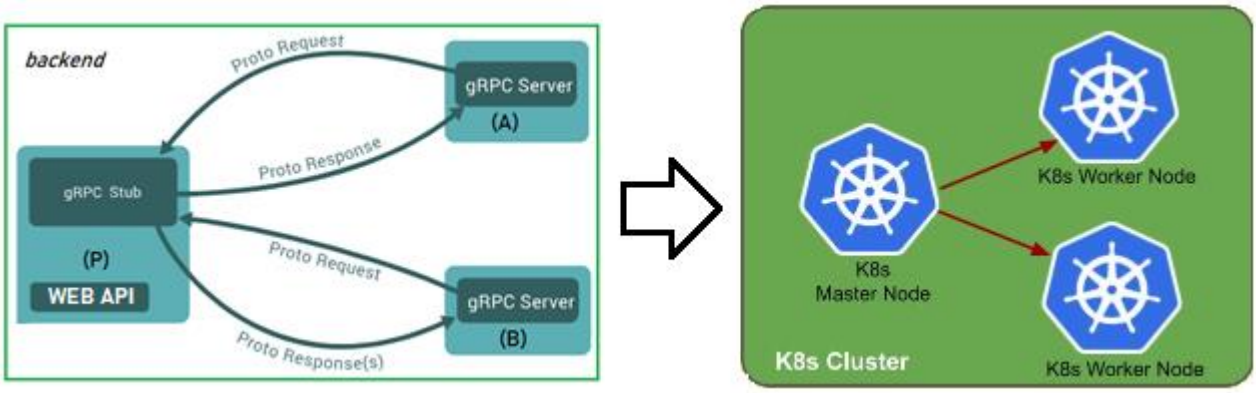

Figura 2 – Instanciação da aplicação no cluster K8S

- Todos os passos e ferramentas utilizados para a criação do cluster kubernetes devem ser documentados.
- Deve-se priorizar a compreensão dos mecanismos disponíveis no kubernetes que permitem fazer com que uma aplicação conteinerizada se adapte a diferentes demandas de uso (por exemplo, o mecanismo de autoscaling do K8S).
- Para compreensão do que significa monitoramento e observabilidade, sugere-se a leitura dos Capítulos 15 e 16 do livro utilizado como referência nessa especificação (Seção 5, a seguir) e veja se o que está proposto na Seção 3 se encaixa. Caso contrário, veja que outras mudanças poderiam ser propostas para viabilizar ou melhorar aspectos de monitoramento e observabilidade.
- Como ferramenta de análise da aplicação instanciada no K8S, deve-se estudar e instalar, no K8S, o Prometheus (https://prometheus.io/), uma ferramenta voltada para observar e monitorar o comportamento de aplicações em clusters kubernetes baseado em métricas. As informações de instalação e recursos utilizados devem ser incluídas no projeto, de forma resumida.

3. Metodologia para garantir observabilidade e monitoramento

A aplicação baseada em microserviços é modular e suas partes podem ser acomodadas de diferentes formas no K8S. A título de exemplo, pode-se considerar que a forma mais simples é considerar os três módulos (P, A e B) executando num único contêiner, enquanto que a forma mais distribuída é aquela na qual cada parte da aplicação está num contêiner instanciado em um *worker node* diferente. Por outro lado, sabe-se que o K8S possui elasticidade – ou seja, mecanismos que conseguem acomodar as aplicações em execução a diferentes demandas garantindo performance aceitável, independentemente da quantidade de requisições sofridas pela aplicação. No entanto, dependendo da arquitetura da aplicação, nem todos os arranjos são admitidos e, em alguns casos, a aplicação pode não funcionar adequadamente.

A proposta metodológica tem o objetivo de identificar formas de arranjo desta aplicação no K8S em modo cluster, visando performance e elasticidade, por meio de testes comparativos. Portanto, deve-se considerar que as análises permitidas pelo Prometheus sejam voltadas prioritariamente para situações e testes de carga na aplicação especificada na Seção 2. Tais comparações devem ser feitas considerando o seguinte:

a) Inicialmente os alunos devem planejar uma forma de simular uma grande quantidade de requisições de modo a "*estressar*" a aplicação e identificar seus limites. Nesse caso, é preciso utilizar alguma ferramenta de teste de carga, cuja escolha e critérios adotados também devem ser documentados. Algumas opções de teste podem ser *(i)* ferramentas de teste de sites web, *(ii) scripts* personalizados construídos pelos próprios alunos, ou *(iii)* ferramentas específicas para o mundo K8S, como o locust (https://locust.io/).

b) Em seguida, os alunos devem definir uma configuração base para a aplicação – Ou seja, instancia-se a aplicação num cenário bem simples, com praticamente nenhuma opção de paralelização e apenas com a distribuição inerente ao gRPC. Nesse caso, aplicam-se testes de carga de modo a identificar *(i)* o tempo médio para atender uma requisição, e *(ii)* a máxima quantidade de requisições recebidas/atendidas pela aplicação por segundo.

c) Com os valores básicos da aplicação anotados, cabe agora o desenho de cenários variando características da aplicação e do cluster K8S que podem impactar no desempenho da aplicação. Além das alternativas de variação da associação da aplicação em contêiners, é possível alterar *(i)* a quantidade de instâncias de cada módulo da aplicação, *(ii)* a quantidade de contêineres nos *worker nodes*, *(iii)* o número de *worker nodes* disponibilizados no cluster, *(iv)* a variação da carga de trabalho submetida, entre outros. A escolha deve ser feita de modo a garantir os requisitos de monitoramento e observabilidade da aplicação.

Em qualquer cenário de teste, é importante:

- Documentar os atributos/métricas que serão testados
- Uso do Prometheus para monitorar/observar a aplicação e o ambiente testado
- Uso de ferramental de teste para submissão da aplicação a diferentes cargas de trabalho (demandas)
- Garantir as mesmas condições de teste de infraestrutura para os testes de modo a não contaminar os resultados
- Para cada cenário montado, fazer teste de carga, observar o comportamento da aplicação e anotar as conclusões

4. Questões de Ordem

Para essa especificação valem as seguintes regras:

- O experimento pode ser feito por grupos de 4 a 5 alunos; não serão aceitos trabalhos individuais. Nesse caso, basta que um dos alunos do grupo faça a postagem das entregas no Moodle (arquivo zipado).
- A entrega é composta por *(i)* códigos, instruções de uso e todas as informações necessárias para esclarecimento e uso dos programas entregues, *(ii)* um relatório, cuja estrutura e conteúdo estão descritos a seguir, *(ii)* um vídeo gravado pelos membros participantes, com apresentação do projeto. Nesse caso, considerar uma média de 4 a 6 minutos por aluno para que possam demonstrar como participaram e conhecimentos adquiridos.
- As comparações de desempenho são válidas se o programa mantiver sua execução adequada durante as execuções.
- O projeto entregue deve seguir a especificação feita neste documento. No entanto, alterações podem ser propostas associadas a uma justificativa, que será analisada para ponderar a nota que será atribuída ao projeto entregue.
- Projetos baseados em outras propostas são aceitos desde que haja referência ao projeto original e indicação das alterações e promovidas pelo grupo.
- Os alunos podem realizar o experimento em qualquer plataforma, inclusive em equipamentos locais, mas devem estar preparados para demonstração da aplicação funcionando *in loco*.
- O relatório a ser entregue deve conter o máximo conjunto de informações sobre o experimento (textos explicativos, figuras, roteiros de instalação, arquivos de configuração, parâmetros usados, etc.), a fim de dar qualidade ao relatório. Mais especificamente o relatório deve conter os seguintes pontos:
  - Dados do curso, da disciplina/turma, data e identificação dos alunos participantes
  - Introdução – pequena descrição da solicitação feita e uma visão geral sobre o conteúdo do relatório
  - A metodologia utilizada (como cada grupo se organizou para realizar a atividade, incluindo um roteiro sobre os encontros realizados e o que ficou resolvido em cada encontro)

- Uma seção sobre a experiência de montagem do Kubernetes em modo cluster
- Uma seção sobre monitoramento e observabilidade, incluindo informações sobre a configuração e uso do Prometheus
- Uma seção sobre a aplicação e as características adotadas em sua versão básica
- Uma seção sobre os cenários de teste com relatos do teste, resultados encontrados e conclusão em cada um deles
- Conclusão – texto conclusivo em função da experiência realizada, comentários sobre dificuldades e soluções encontradas. Ao final, cada membro do grupo abre uma subseção para comentários pessoais sobre a pesquisa, indicando as partes que mais trabalhou, aprendizados e uma nota de autoavaliação.
- Referências utilizadas – cuidado para não utilizar materiais de terceiros sem a devida citação.
- Anexos (opcional) – com eventuais informações não apresentadas anteriormente, tais como arquivos de configuração, comentários sobre os códigos construídos, instruções de execução e informações adicionais para permitir replicação do laboratório pelo professor. Arquivos e informações adicionais que não puderem ser postadas no Moodle podem ser disponibilizadas via *GitHub*.

- O projeto será avaliado sob dois aspectos: *(i)* qualidade das entregas, e *(ii)* participação, envolvimento com o experimento (descritos no vídeo). Com relação à qualidade das entregas, a nota é proporcional aos resultados apresentados. Por exemplo, bons testes/descobertas, percepção de equilíbrio na distribuição das tarefas do projeto entre os membros do grupo, boa documentação (incluindo vídeo) contam positivamente para obtenção de uma boa nota.
- A nota emitida levará em conta os seguintes atributos/pesos: *(i)* 20% para qualidade das entregas (relatório, vídeo, etc.), *(ii)* 80% para o nível técnico e de exploração das solicitações feitas.
- Ponto extra para funcionalidades não solicitadas. Por exemplo, proposição de outras formas de monitoramento e observabilidade para a aplicação, tal como a montagem de um pipeline de observabilidade considerando outras métricas não discutidas aqui.

5. Referências

[1] Arundel, J. and Domingus, J. Cloud Native DevOps with Kubernetes – Building, Deploying and Scaling Modern Applications in the Cloud, O´Reilly, 2019.