Relatório do Desafio

Primeiramente, tentei instalar o ubuntu no Windows, como foi proposto. Sem sucesso com o mikrok8s, resolvi partir para máquina virtual, instalando o VMware Workstation. Estudei como instalar o microk8s e como fazer a aplicação funcionar. Consegui, com algum esforço, fazer a aplicação funcionar, embora não tenha conseguido acessar a aplicação pelo localhost, e sim pelo cluster-ip do productpage.

À medida que eu avançava nos estudos para rodar uma aplicação de monitoramento, a máquina virtual ficou lenta demais e, então, resolvi desistir dessa ideia. Peguei emprestado um outro computador, instalei o ubuntu em dual mode (com o Windows) e passei então a trabalhar nele.

Embora o trabalho tenha melhorado sensivelmente a partir daí, ainda obtive travamentos e lentidões constantes ao longo do processo. Consegui, mais uma vez, rodar a aplicação e rodar também o monitoramento (usando o istio, que é composto com a aplicação GRAFANA, que mostra graficamente tanto os bytes trafegados, como mostra graficamente o uso da CPU, da memória, fluxo de rede, e uma série de informações da aplicação em questão). Resolvi adotar o istio por ser uma ferramenta que vem já embutida no microk8s, sendo necessário apenas habilitá-la.

Feito isso, passei a estudar como se faziam os commits de repositório, e resolvi usar o git. Então, gerei os commits para adições dos stacks, que não foram alterados em relação ao seu código. Os stacks são: bookinfo.yaml (aplicação), bookinfo-gateway.yaml (necessário para expor os serviços via NodePort) e o logging-stack.yaml (para futuro monitoramento de log). Todos os arquivos estão disponíveis no próprio demo do istio na internet.

Para o logging-stack, tive o cuidado de verificar o namespace “logging”, enquanto que a aplicação bookinfo está no namespace “default” e o de monitoramento no próprio “istio-system”, visto que esta era uma das especificações do desafio.

Já aproveitando, fui estudar como fazer o monitoramento de log e vi que já existe também uma ferramenta para tal finalidade no microk8s. São três ferramentas na verdade, que trabalham em conjunto: o ElasticSearch, o Kibana e o FluentD. O ElasticSearch é uma ferramenta escalável de busca em tempo-real para estruturas e textos e geralmente é usado para pesquisa em grandes quantidades de log. Ele é usado em conjunto com o Kibana, que é a parte gráfica, onde são mostrados os dados. O FluentD é a aplicação que coleta e filtra os logs e manda para o ElasticSearch.

O sistema de monitoramento de log então foi testado, e foi visto que estava funcionando.

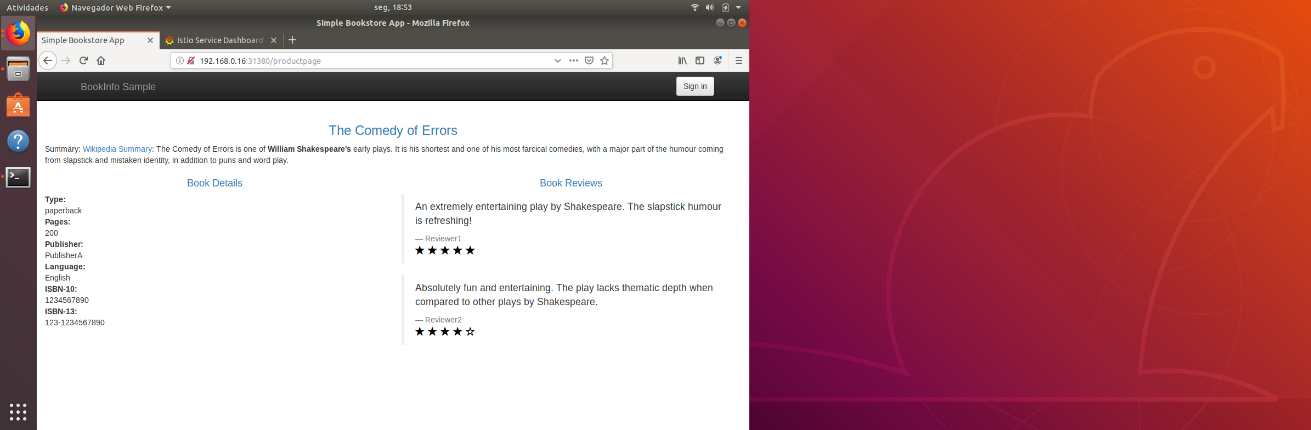
Depois de testados todos os dois sistemas, o escopo básico foi concluído, dando seguimento assim ao escopo desejado. Para o escopo desejado, foram utilizados dois arquivos: “destination-rule-all-mtls.yaml” e “virtual-service-reviews.yaml”. O destination rules, como o próprio nome já diz, serve para referenciar um serviço endereçável da rede para o qual a solicitação será enviada após o processamento de uma regra de roteamento. Para esse caso, o destionation rules são as regras “default” (padrão) para a aplicação bookinfo. Já o virtual-service-reviews é o código que abstrai os headers das páginas e aponta para um review específico.

Após criar e testar os stacks do escopo desejado, o resultado foi satisfatório, como desejado. Depois foi aberto o GRAFANA novamente, onde há uma opção para separar o monitoramento por serviço, entre outras informações.

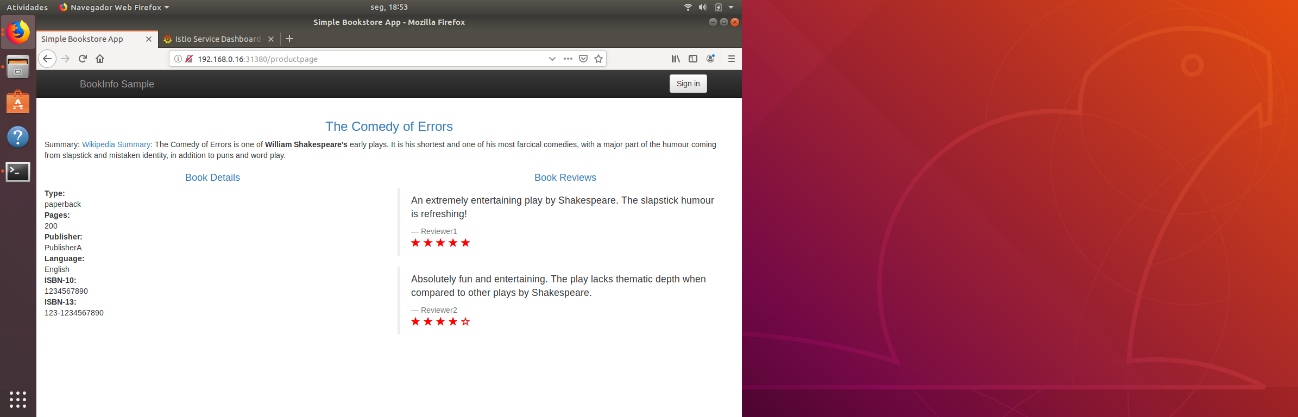
Fiz cinco scripts: bookinfo, limpar, istio, escopo2 e limpar2. O bookinfo cria tudo que é necessário para rodar a aplicação, monitoramento e log. O Istio serve para habilitá-lo no microk8s, caso não esteja habilitado. O “limpar”, como o próprio nome sugere, apaga tudo da memória criada do projeto. O escopo2 cria o destination rule e o virtual-service-reviews. E, por fim, o limpar2 limpa tudo do escopo2, retornando a aplicação bookinfo para o estado do escopo 1 (básico).

A respeito do escopo ótimo do desafio, comecei a estudar como limitar os requests, mas não consegui concluir.

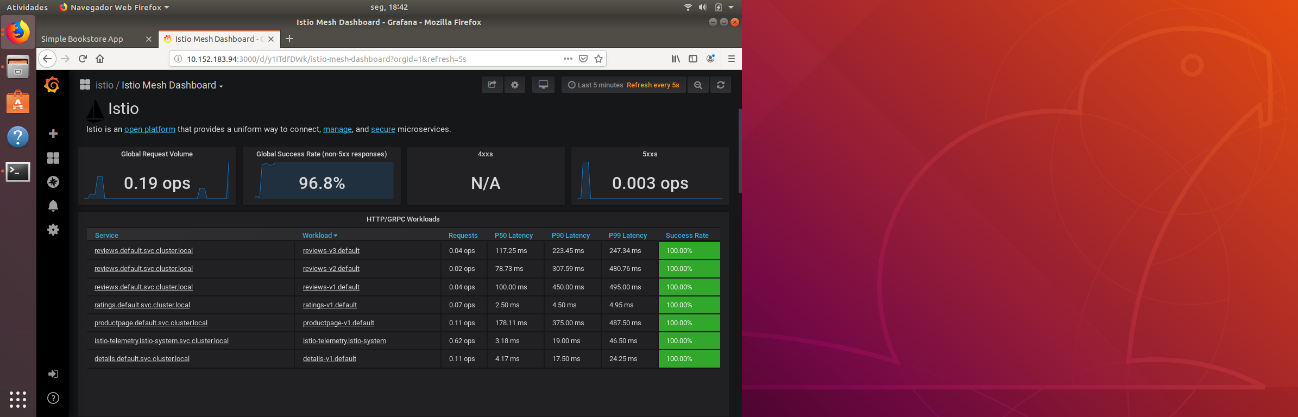
Anexos



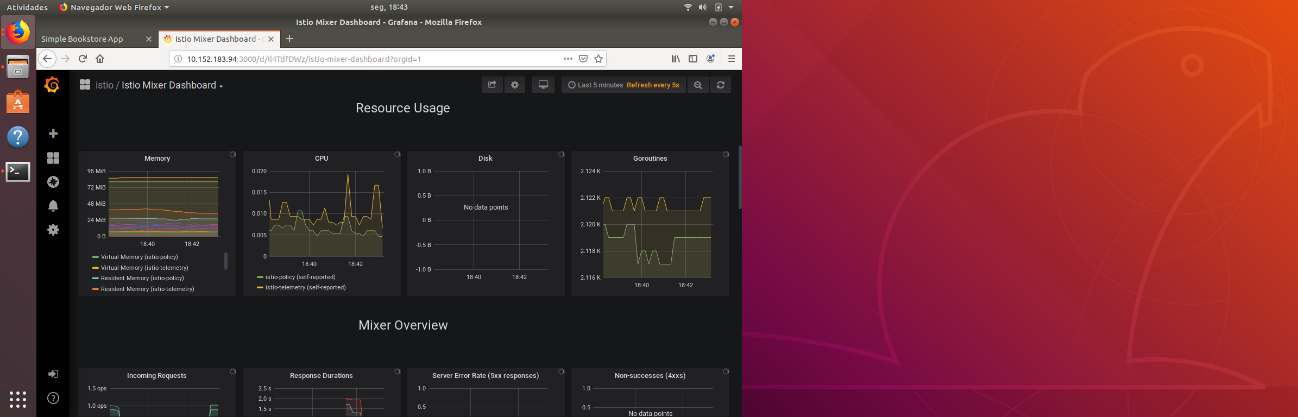
*aplicação rodando bookinfo (figura 1)*



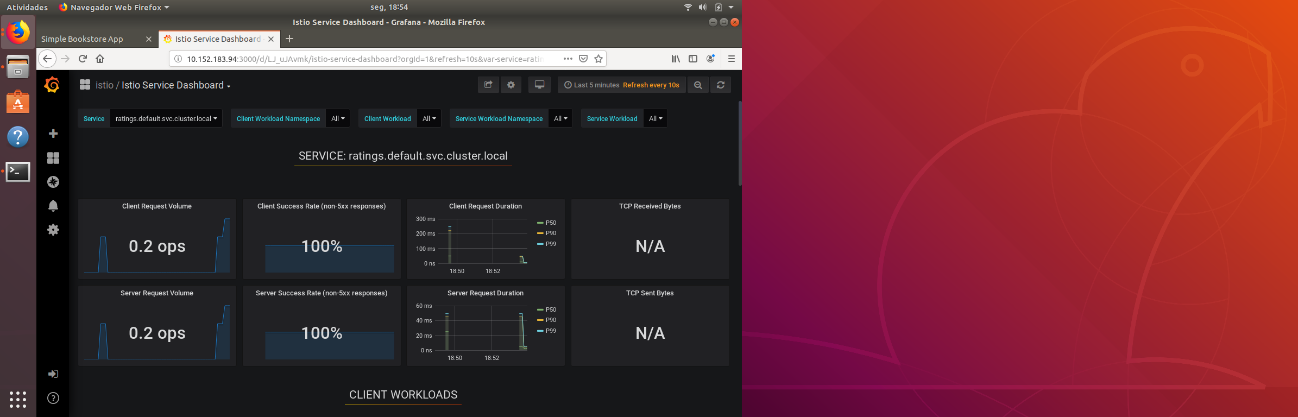
*aplicação rodando bookinfo (figura 2)*



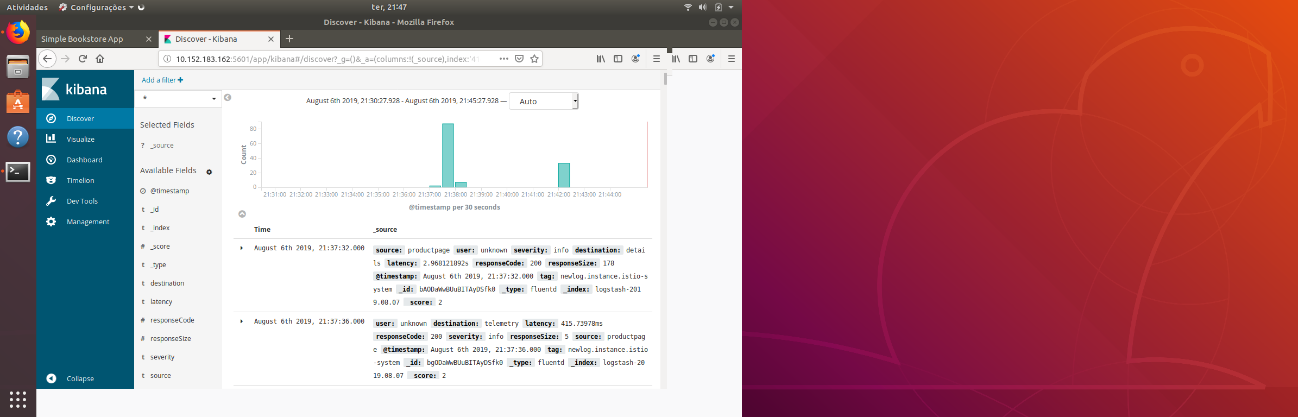
*aplicação de monitoramento grafana – Istio Mesh Dashboard (figura 3)*



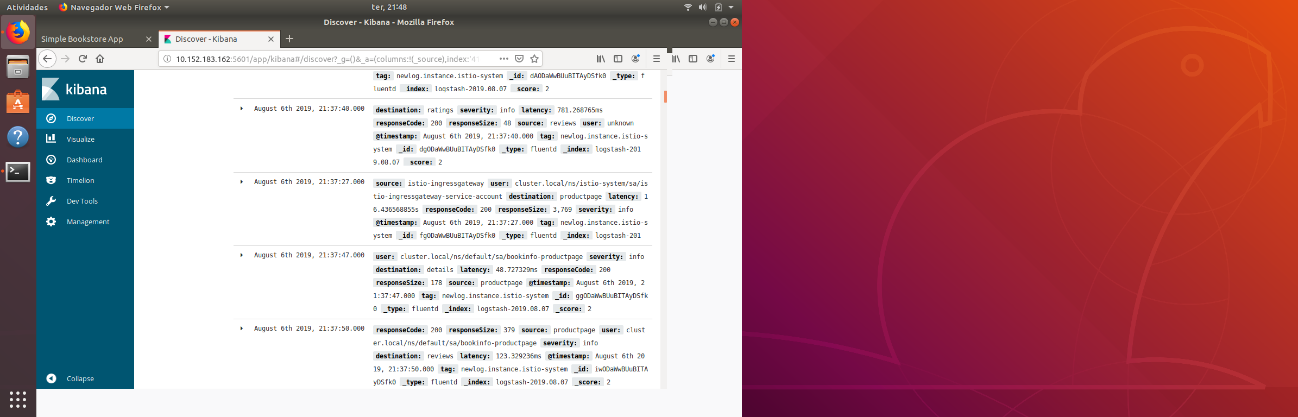
*aplicação de monitoramento grafana – Istio Mixer Dashboard (figura 4)*



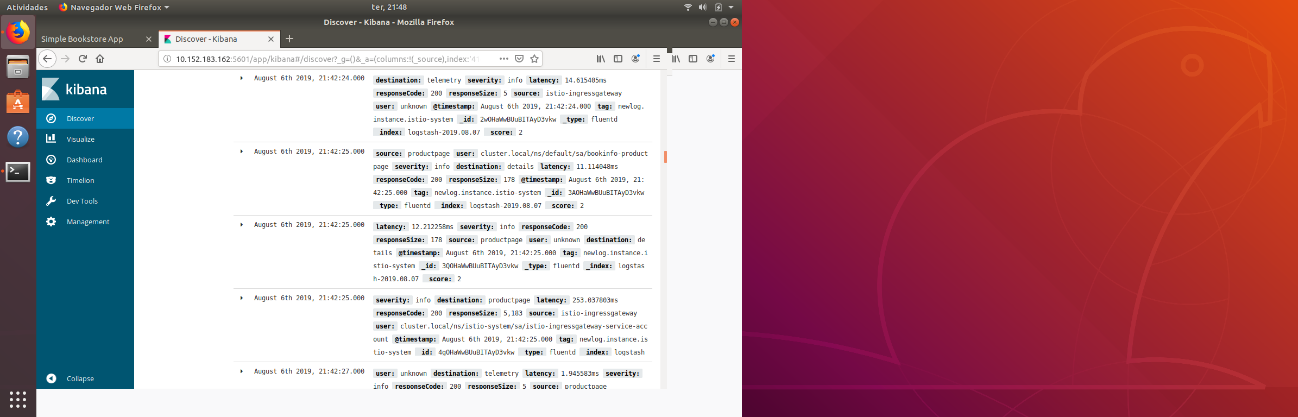
*aplicação de monitoramento grafana – Istio Service Dashboard – do serviço de rating (figura 5)*



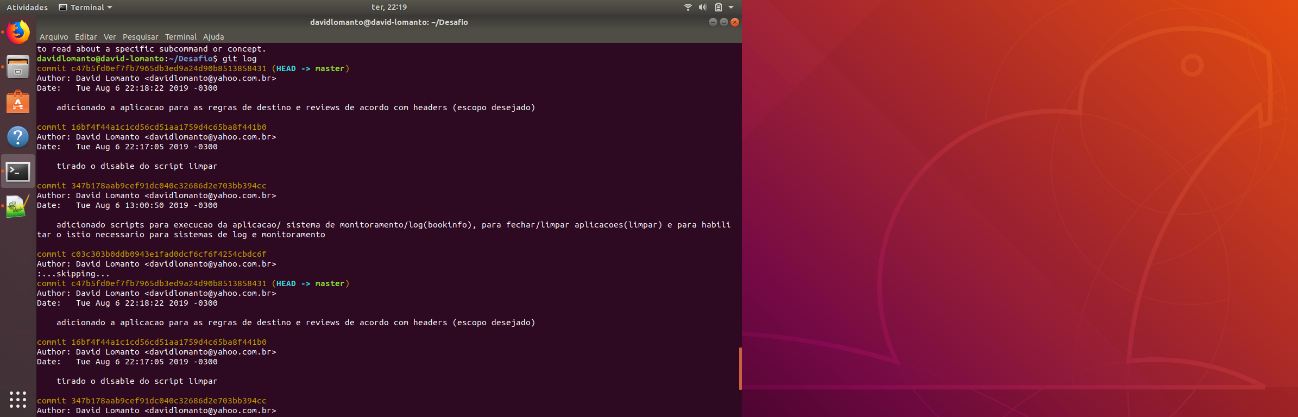
*aplicação de log kubana (figura 6)*



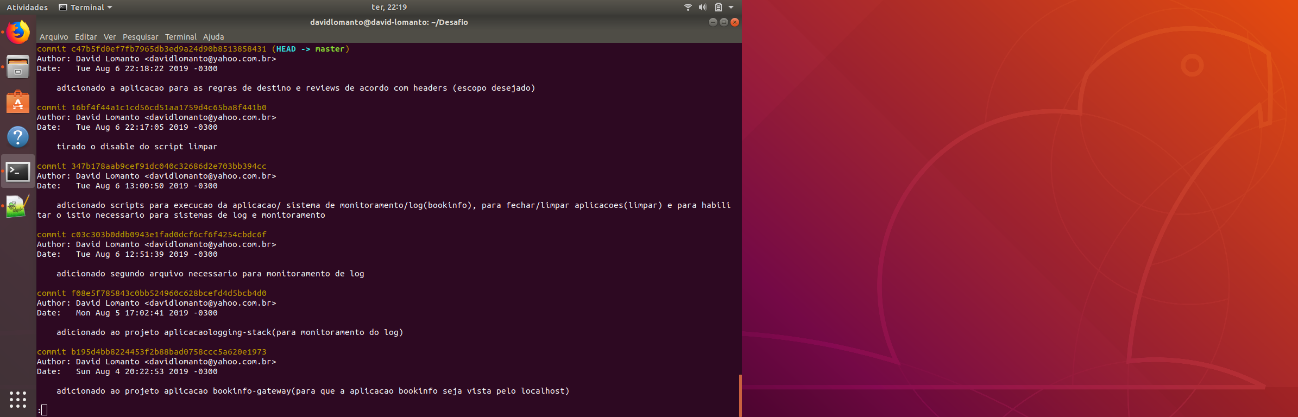
*aplicação de log kubana (figura 7)*



*aplicação de log kubana (figura 8)*



*git – commits do projeto desafio (figura 9)*



*Git – commits do projeto desafio (figura 10)*