Relatório Trabalho Final

Tópicos em Redes e Sistemas Distribuídos

Professor André Luiz Beltrami Rocha

Caio Peres
Guilherme Madureira

Arquitetura da solução:

VM1:

- Prometheus, Node Exporter, cAdvisor, MongoDB, App1, App2 e App3.

Em containers na VM1:

- cAdvisor, MongoDB, App1, App2 e App3.

VM2:

- Node Exporter, cAdvisor e App1.

Em containers na VM2:

- cAdvisor e App1.

Aplicação:

- **App1:**

  Aplicação feita utilizando C#, junto com o ambiente .NET Core. Para isso, utilizamos uma imagem pronta da Microsoft com o SDK configurado. Após configurado o container, foi necessário compilar o programa para linux, utilizando o seguinte comando:

  dotnet publish -c release -r ubuntu.16.04-x64 --self-contained

  No Dockerfile da aplicação foi necessário apenas rodar a aplicação e escolher a opção de servidor.

- **App2:**

  Aplicação feita utilizando Nodejs, porém ao invés de ser um webserver, ele apenas realiza requisições. Dessa forma, era mais fácil a manipulação de JSON que receberíamos do Prometheus.

- **App3:**

  Aplicação bem parecida com o App2, porém utilizando as métricas pedidas no projeto, especificado para as VMs. Também foi feito em Nodejs.

Decisões de Projeto:

Entender a arquitetura foi o maior desafio para as decisões, porém uma vez que entendemos o que estava sendo proposto, precisávamos apenas seguir o modelo.

Os passos para a realização do projeto foram:
Entender a arquitetura macro, definir uma VM principal, definir arquitetura do App1, testar no host e depois passar para a VM1 em container. Após isso, foi o momento de instalar o Prometheus, cAdvisor e Node Exporter, tudo na VM1, ao mesmo tempo em que testávamos o vagrantfile. Após muito troubleshooting, copiamos os mesmos comandos com exceção da instalação do prometheus para a VM2. Por fim definimos a arquitetura do App2, testamos no host e depois passamos a aplicação para container na VM1 e instalamos também o MongoDB na VM1.

Dificuldades Encontradas:

Entender a arquitetura foi a maior dificuldade, porém lidar com o Vagrant também é difícil, pois todos os comandos precisam estar funcionando e muitas vezes em ordem. Alguns comandos do linux mudam o diretório atual, e isso fez com que nós ficássemos sem entender o que estava acontecendo (necessitando adicionar alguns comandos repetidos para ter certeza que o diretório atual era o correto).

O Prometheus não ajuda muito na hora de filtrar os campos que nós precisávamos, tendo que fazer uma manipulação de JSON manualmente.
Uma coisa que não foi exatamente uma dificuldade, mas incomodou um pouco, foi a tempo para instanciar as máquinas e instalar todos os componentes pelo vagrant up, já que utilizando o C# e o sdk da microsoft, esse processo se tornava consideravelmente mais lento.

Figura macro da arquitetura da solução: