

Trabalho Prático 3: Serverless Computing

Cloud Computing - Lorenzo Carneiro Magalhães

Task 3: Serverless Runtime

Nessa terceira etapa, a tarefa era substituir a imagem de docker do runtime fornecida por um runtime próprio.

Implementação

Nessa tarefa desenvolvi duas classes: Context e ContextManager. A classe Context serviu apenas para passar como parâmetro da função entrypoint, enquanto a classe ContextManager fazia todo o processo lógico para o funcionamento do runtime.

Em primeiro momento, busquei deixar o funcionamento do runtime exatamente igual ao da imagem docker fornecida previamente, e após essa versão base estando funcionando, iria adicionar as funcionalidades extras. Assim, implementei todas as funcionalidades base.

O maior ponto de atenção na implementação foi a execução do módulo lido. Eu nunca havia mexido com a execução de código a partir de uma string a não ser por um eval simples, então tive de aprender a mexer com a biblioteca importlib, que é utilizada justamente nesse contexto de importação de módulos.

Após garantir o bom funcionamento das funcionalidades básicas, incluindo a passagem correta dos parâmetros input e context e a persistência de context.env, parti para as funcionalidades extras. As extensões de Redis Input Key, Redis Monitoring Period e EntryPoint name foram muito triviais, já que bastaram adicionar o recebimento de um parâmetro adicional via environment.

Agora o Complex Function Support requereu mais estudo, mas nada muito complexo: para aceitar o arquivo zip, o usuário deve gerar um ConfigMap similar ao gerado com o arquivo pyfile, mas nesse caso específico para o zip. Além disso, deve-se configurar dois parâmetros adicionais: `ZIP_TARGET_PATH` e `ZIP_ENTRY_RELATIVE_FILE_PATH`. O primeiro deve especificar onde o zip foi montado e o segundo especifica qual é o arquivo entrypoint, que deve conter também a função entrypoint escolhida. Caso esses parâmetros não forem preenchidos, o runtime não aceita zip e passa a aceitar um arquivo somente, que deve possuir caminho igual a `/opt/usermodule.py`, conforme estava na imagem docker fornecida.

Por fim, é importante falar que o arquivo de deployment fornecido pelo professor, mudando apenas o nome da imagem, continua sendo aceito pelo novo runtime.

Outro aspecto importante da implementação é que ela não chama o entrypoint se não tiver tido mudanças nas métricas, então caso o tempo inserido no Redis Monitoring Period for baixo, não quer dizer que o runtime vai chamar a função todas as vezes.