

Faça como eu fiz

Fizemos a configuração completa da nossa API dentro do Cluster. Agora é sua vez de configurar o próprio cluster e subir sua primeira aplicação no Kubernetes Engine!

Coloque o seguinte script em um arquivo chamado `api/apiDeSomas.js` :

```
const Koa = require('koa')
const processador = require('koa-bodyparser')
const aplicacao = new Koa()

const BigQuery = require('@google-cloud/bigquery').BigQuery
const instancia = new BigQuery()
const tabela = instancia.dataset('forumAlura').table('atividades')

async function executarQuery (query) {
  const opcoes = {
    query: query
  }

  const [trabalho] = await tabela.createQueryJob(opcoes)
  const [resultados] = await trabalho.getQueryResults()
  return resultados
}

aplicacao.use(processador())
aplicacao.use(async function (contexto) {
  const queryDeSomaPorCurso = 'SELECT count(*) as soma, nome_do_curso FROM atividades GROUP BY nome_do_curso'
  const queryDeSomaPorAula = 'SELECT count(*) as soma, nome_da_aula FROM atividades GROUP BY nome_da_aula'
  contexto.status = 200
  contexto.body = {
    soma_por_curso: await executarQuery(queryDeSomaPorCurso),
    soma_por_aula: await executarQuery(queryDeSomaPorAula)
  }
})

aplicacao.listen(3000)
console.log('A API está funcionando normalmente')
```

Use a seguinte configuração de imagem em um Dockerfile, para criar uma imagem do Docker, como abordamos na [atividade "Faça Como Eu Fiz" da aula 4](#) (<https://cursos.alura.com.br/course/google-gke/task/75743>):

```
FROM node:10
COPY . /api
EXPOSE 3000
CMD ["node", "api/apiDeSomas.js"]
```

Em seguida, configure seu cluster e crie um deploy para executar essa API! Não se esqueça de expor a API com um serviço.