

Fazendo sua primeira integração contínua com GoLang e Serverless

Júlio Lustosa

Engenheiro DevOps



Problema

Primeiros Passos com Bitbucket Pipeline

The screenshot displays the Bitbucket Pipelines interface for a repository named 'Alpha-team-app'. The left sidebar contains navigation options: Source, Commits, Branches, Pull requests, Pipelines (selected), Deployments, Downloads, Boards, and Settings. The main area shows 'Pipeline #14' which is 'Successful' and was pushed by Jonathon Lowe 26 seconds and 20 hours ago. The pipeline consists of five steps: 'build' (6s), 'test' (6s), 'Deploy to staging' (6s), and 'Deploy to production' (6s). The 'test' step is highlighted, and its logs are shown in a dark-themed window on the right. The logs indicate a failure in the 'jest' test suite due to a high-priority assertion error: 'FAIL ./calculator.test.js' with the message 'two high priority ops'. The error details show that the expected value was 5.75 and the received value was 4.25. A list of test results follows, showing 9 passed tests and 1 failed test.

Alpha-team-app Jonathon Lowe / alpha-team-app / Pipelines

Pipeline #14

Successful Rerun

26 sec • 20 hours ago

Push by Jonathon Lowe

Commit >

9deeach

Pipeline

View configuration

- build 6s
- test 6s
- Deploy to staging 6s
- Deploy to production 6s

Logs

Build + Add a service or database

```
+ npm test
npm info it worked if it ends with ok
npm info using npm@9.3.10.10
npm info using node@v6.9.4
npm info lifecycle @atlassian/calculator@1.0.0-0-pretest: @atlassian/calculator
npm info lifecycle @atlassian/calculator@1.0.0-0-test: @atlassian/calculator

> @atlassian/calculator@1.0.0 test /opt/atlassian/pipelines/agent/build
> jest

FAIL ./calculator.test.js
  • two high priority ops

  expect(received).toBe(expected)

  Expected value to be (using ===):
    5.75
  Received:
    4.25

  at Object.<anonymous>.test (calculator.test.js:38:38)
  at Promise.resolve.then.el (node_modules/p-map/index.js:46:16)
  at process._tickCallback (internal/process/next_tick.js:103:7)

  ✓ single addition (4ms)
  ✓ multiple addition (1ms)
  ✓ dangling addition (1ms)
  ✓ subtraction
  ✓ addition and subtraction (1ms)
  ✓ multiplication
  ✓ division (1ms)
  ✓ order of operations (1ms)
  ✓ multiple order of operations (1ms)
  ✗ two high priority ops (3ms)

Test Suites: 1 failed, 1 total
Tests:      1 failed, 9 passed, 10 total
Snapshots: 0 total
Time:       0.652s
```



bitbucket_serverless



Source



Commits



Branches



Pull requests



Pipelines



Deployments



Issues



Wiki



Downloads



Boards



Settings

PIPELINES

Settings

Deployments **NEW**

Repository variables

SSH keys

Integrations

Pipelines settings

Pipelines will build your repository on every push once you enable Pipelines and commit a valid bitbucket-pipelines.yml file in your repository.

Enable Pipelines



[View bitbucket-pipelines.yml](#)



Name



bitbucket-pipelines.yml

Arquivo de exemplo

```
! bitbucket-pipelines.yml bitbucket-pipelines.yml
```

```
image: golang:latest # Imagem Docker
```

```
pipelines:
```

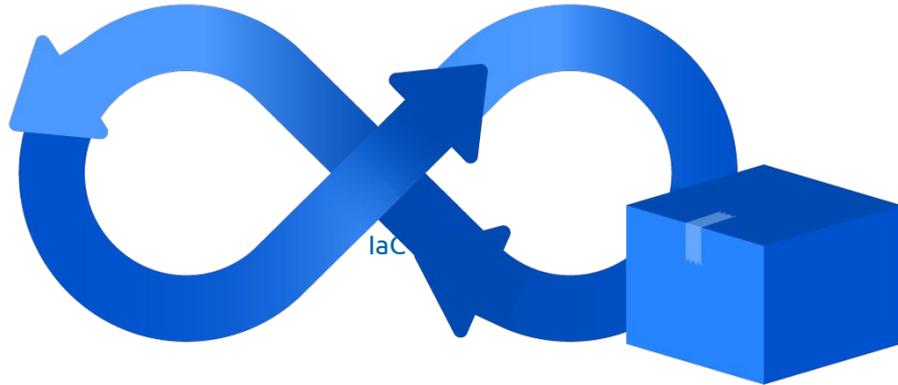
```
  default:
```

```
    - step:
```

```
      script:
```

```
        - echo "Meu primeiro pipeline ;)"
```

Criando um Artefato



```
image: golang:latest
```

```
pipelines:
```

```
  default:
```

```
    - step:
```

```
      name: "Criando o Artefato"
```

```
      script:
```

- source bitbucket-pipelines-go.sh
- cd \${IMPORT_PATH}
- go build -o bin/hello hello/main.go

```
      artifacts:
```

- bin/**

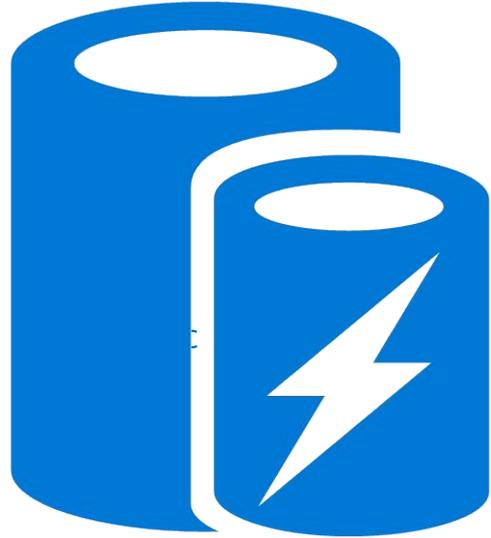
```
    - step:
```

```
      name: "Executando o Artefato"
```

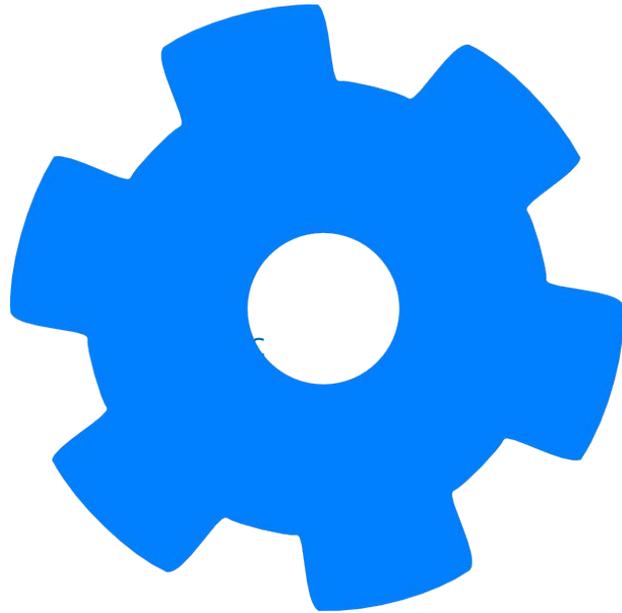
```
      script:
```

- source bitbucket-pipelines-go.sh
- cd \${IMPORT_PATH}/bin
- ./hello

Trabalhando com Cache



Criando etapa personalizada



Primeiros passos com Serverless Framework



serverless

```
$ npm install -g serverless
```

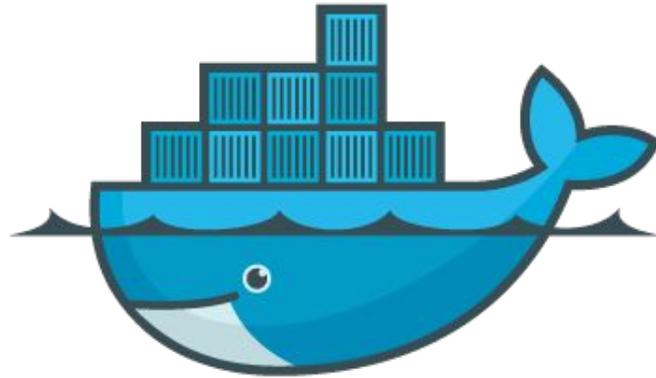
Fazendo deploy com Serverless Framework



serverless

```
$ serverless deploy --stage dev
```

Criando imagens docker para CI e CD



docker

```
FROM node:8-alpine
```

```
RUN apk update && apk add --update --no-cache \  
    git \  
    bash \  
    curl \  
    openssh \  
    py-pip
```

```
RUN npm config set unsafe-perm true
```

```
RUN npm update -g
```

```
RUN pip install --upgrade pip && \  
    pip install --upgrade awscli
```

```
# Instala Serverless Framework
```

```
RUN yarn global add serverless
```

```
RUN mkdir /app
```

```
WORKDIR /app
```

Dockerfile

Estamos contratando

**globo
.com**

talentos.globo.com

Exemplos Práticos



Todos os códigos fontes em
bitbucket.org/juliolustosa/bitbucket_serverless

Obrigado !
Júlio Lustosa



[linkedin.com/in/juliolustosa](https://www.linkedin.com/in/juliolustosa)