**Web Academy CI/CD**

**Laboratório - Aula 02**

**Objetivo do Laboratório**: Criando workflows via projeto local associado a conta no github, treino com checkout e configuração de ambientes.

**Pré-Requisitos:**

* Ter uma conta no GitHub
* Máquina com Linux / Windows instalado e Git já configurado com a conta do GitHub
* Arquivos disponíveis no Colabweb para o Laboratório (caso necessário)

**Exercício 01 - Associando um projeto simples a um repositório com um workflow básico (checkout, compilar)**

1 - Criar um diretório para o seu projeto "lab02"

2 - Nele, crie um diretório .github/workflows para armazenar o seu arquivo build.yml

Inicie o seu arquivo build.yml com o seguinte conteudo

name: Simple Local

on: push

jobs:

first-job:

runs-on: ubuntu-latest

steps:

- name: Diga oi

run: echo "Hello World!"

3 - Crie um arquivo .gitignore para configurar o que o git não deve rastrear

Inicie o seu arquivo .gitignore com o seguinte conteudo

node\_modules

4 - Crie um arquivo package.json para configurarmos o nosso projeto simples

Inicie o seu arquivo package.json com o seguinte conteudo

{

"name": "exemplo",

"main": "app.js"

}

5 - Crie um arquivo app.js para o nosso aplicativo simples de teste.

console.log("Exemplo Simples");

var moment = require('moment');

var date = moment().format('LL');

console.log(date);

6 - Crie um repositório público em sua conta

7 - Vá ao seu projeto local e associe ele ao seu repositório recém-criado

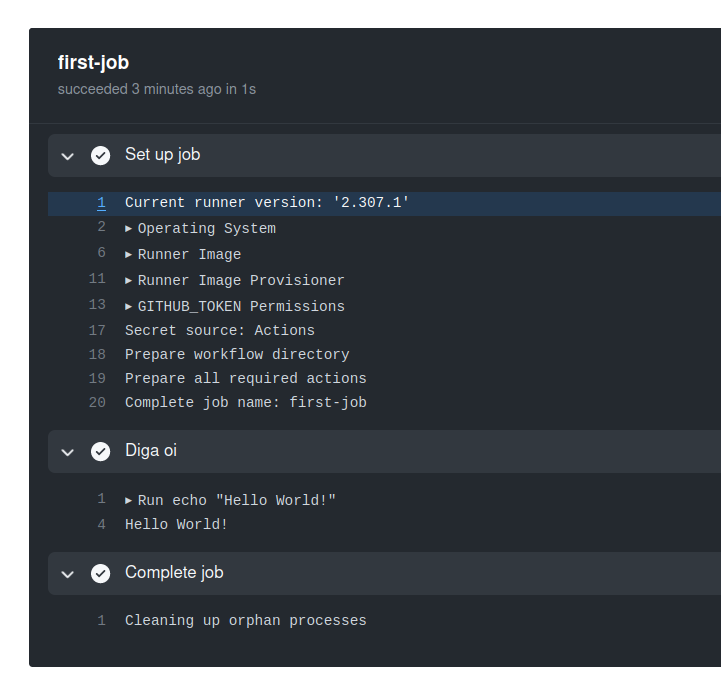
git init

git add README.md

git commit -m "first commit"

git remote add origin <link do seu repositorio>

git push --set-upstream origin master

8 - Observe em sua conta que o repositório está ok e o workflow de início rodou

9 - Vamos atualizar agora o nosso build.yml para o seu novo conteudo!

Agora vamos fazer o checkout do nosso código para as máquinas do github e rodar o nosso exemplo.

Atualize o seu build.yml para o seguinte conteudo

name: Simple Local

on: push

jobs:

build-node:

runs-on: ubuntu-latest

container: node:18

steps:

- run: node --version

- run: npm --version

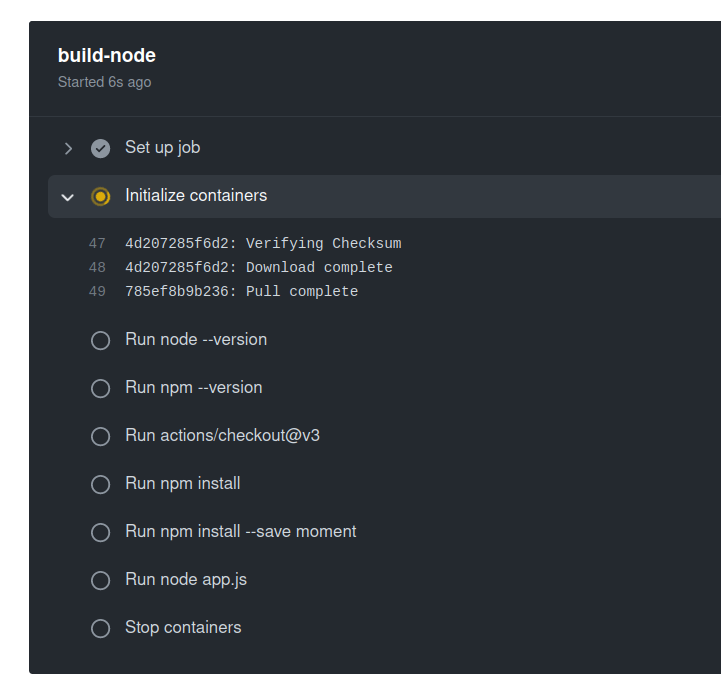
- uses: actions/checkout@v3

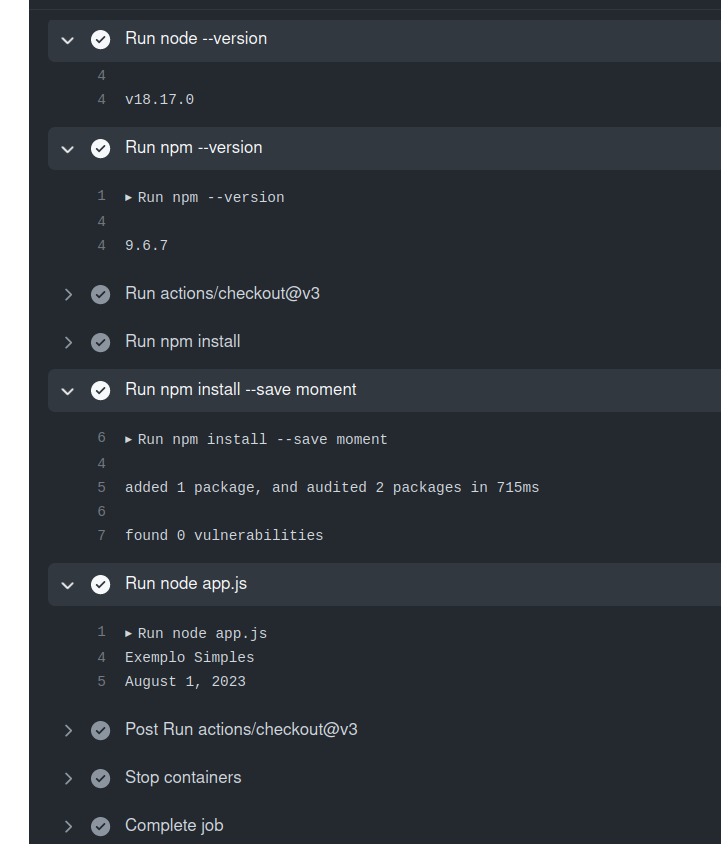
- run: npm install

- run: npm install --save moment

- run: node app.js

10. Observe na sua conta do github, na tab Actions , o seu workflow sendo executado.



11 - Ao final, o seu projeto foi executado com sucesso, através do checkout do seu código e posterior configuração e execução!

**Parabéns!! Fim do exercício!**

**Exercício 02 - Criando um workflow com mais passos (e controlando alternativas de execução)**

01 -Agora, vamos partir de um projeto já pronto e um pouco mais complexo para o nosso exemplo.

02 - Baixe o arquivo aula02\_lab02.zip e descomprima o arquivo no seu diretório de trabalho.

03 - Compile o projeto com o comando "npm install"

04 - Rode o projeto com o comando "npm run dev" e abra um navegador no link que é informado após a execução.

Com tudo correto, vamos agora associar o projeto a uma conta no github

05 - Crie um repositório público e associe o projeto local ao seu respectivo repositório

git remote add origin <link do seu repositorio>

git push --set-upstream origin master

Perceba que não temos workflows configurados e a aba Actions está na sua configuração padrão

06 - Vamos adicionar a primeira versão do nosso arquivo .yml - crie o arquivo .github/workflows/build01.yml e copie o seguinte conteudo

name: Build V1

on: push

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Get code

uses: actions/checkout@v3

- name: Install dependencies

run: npm ci

- name: Lint

run: npm run lint

- name: Test code

run: npm run test

- name: Build code

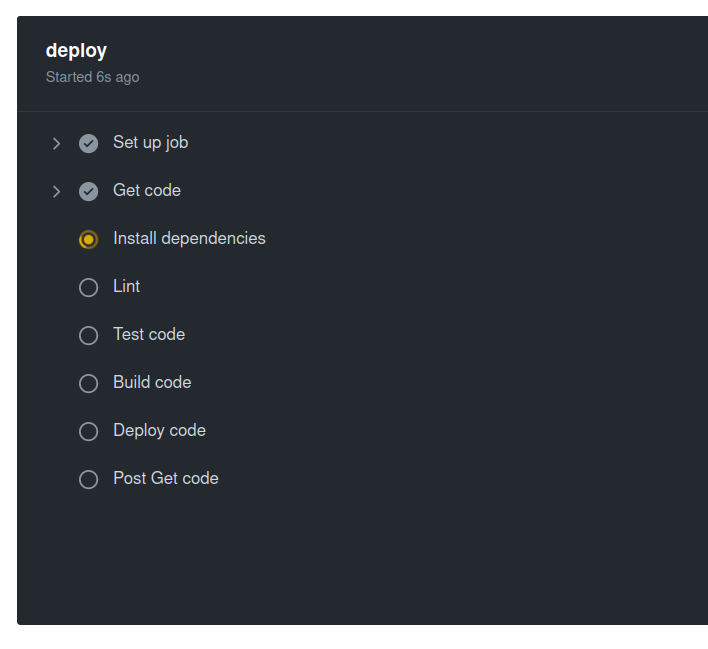
run: npm run build

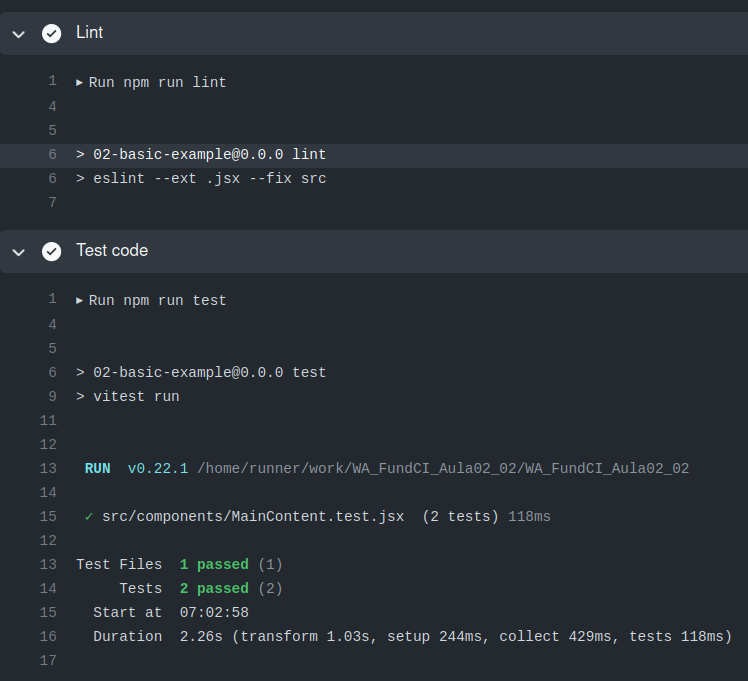
- name: Deploy code

run: echo "Deploying..."

07 - Adicione o arquivo ao controle do git, commit e faça o push

08 - No seu repositório remoto, vá na aba Actions para ver o seu workflow funcionando.



09 - Veja como os passos do lint e do teste passam com sucesso 

10 - Vamos agora adicionar uma segunda alternativa de arquivo de build, dessa vez, com vários passos ao invés de um só.

11 - Crie o arquivo build02.yml no diretorio .github/workflows e coloque o seguinte conteudo

name: Build V2

on: push

jobs:

lint:

runs-on: ubuntu-latest

steps:

- name: Get code

uses: actions/checkout@v3

- name: Install dependencies

run: npm ci

- name: Lint

run: npm run lint

test:

needs: lint

runs-on: ubuntu-latest

steps:

- name: Get code

uses: actions/checkout@v3

- name: Install dependencies

run: npm ci

- name: Test code

run: npm run test

deploy:

needs: test

runs-on: ubuntu-latest

steps:

- name: Get code

uses: actions/checkout@v3

- name: Install dependencies

run: npm ci

- name: Build code

run: npm run build

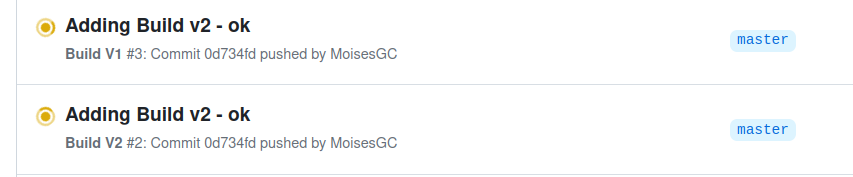
- name: Deploy code

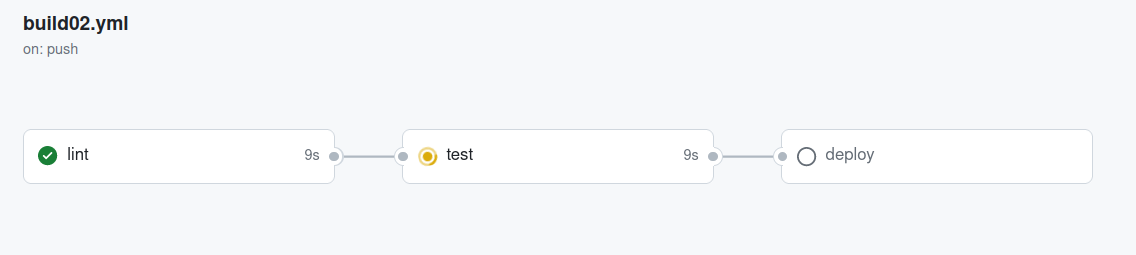
run: echo "Deploying..."

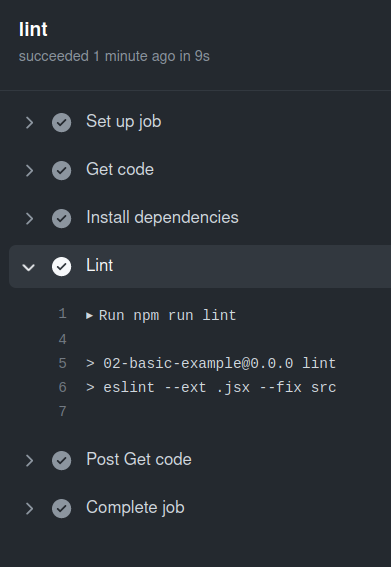
12 - Adicione o arquivo ao controle do git, comite e faça o push

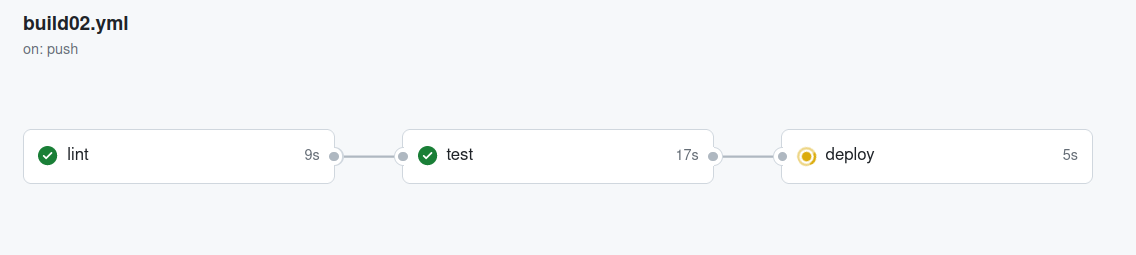
13 - No seu repositório remoto, vá na aba Actions para ver o seu workflow funcionando - perceba que o workflow anterior continua funcionando e vai ser acionado por este - ambos são acionados por push!

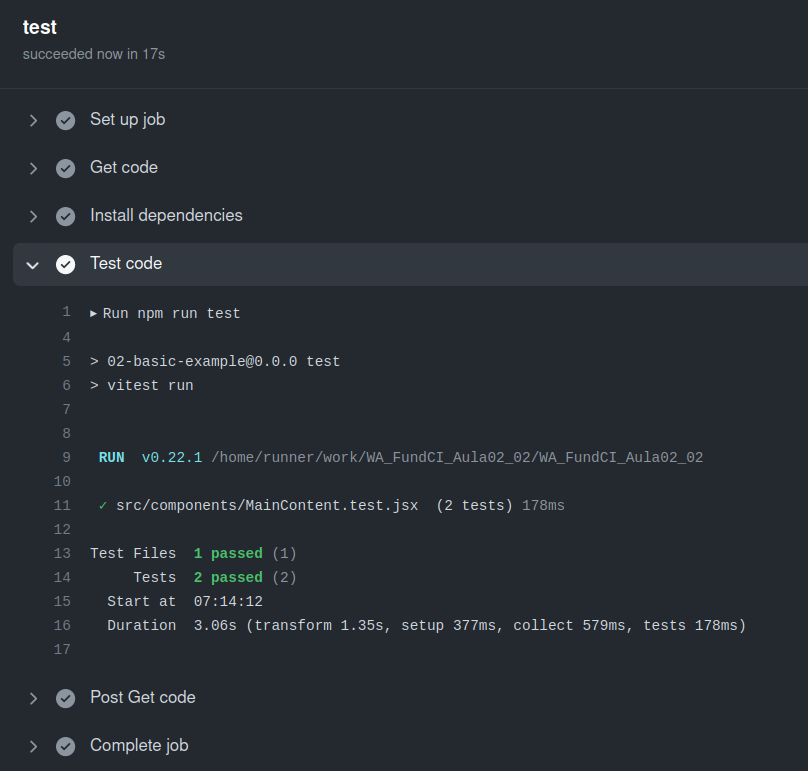
Note que agora as etapas estão encadeadas, dado a segunda estratégia implementada!

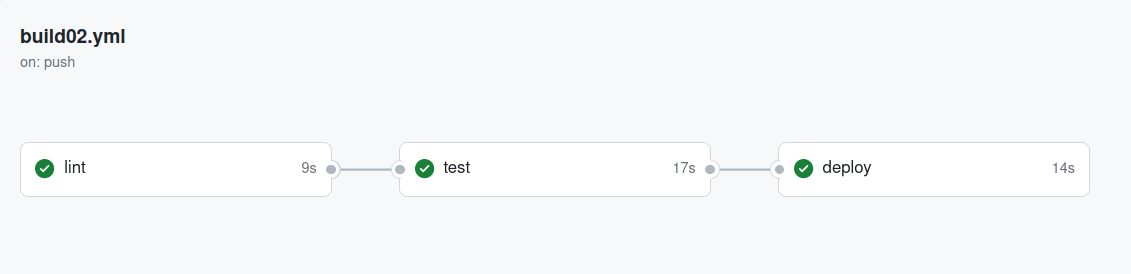


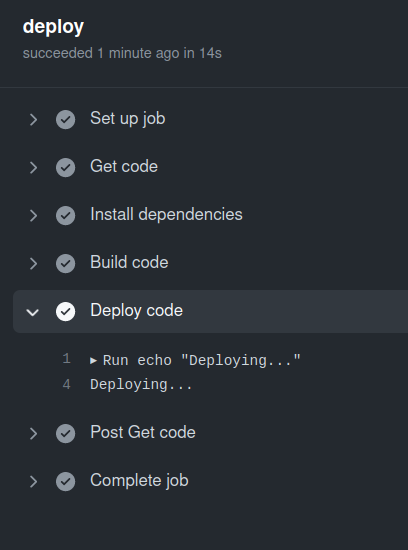












**14 - Parabéns, você terminou o exercício!!**