**Web Academy CI/CD**

**Laboratório - Aula 04**

**Objetivo do Laboratório**: Consolidar o uso de pipelines através da configuração de pipelines que utilizam condicionais e deploy em infra-estrutura em nuvem!

**Pré-Requisitos:**

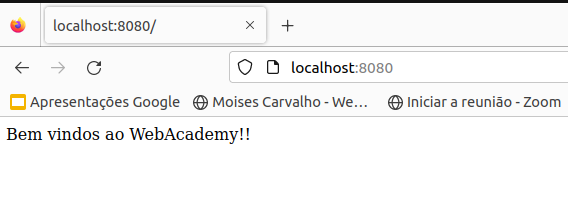
* Ter uma conta no GitHub
* Máquina com Linux / Windows instalado e Git já configurado com a conta do GitHub
* Ter uma conta no Docker Hub
* Ter uma conta no Google Cloud com Google Cloud Services
* Arquivos disponíveis no Colabweb para o Laboratório (caso necessário)

**Exercício - Utilizando condições para deploy no Google Cloud**

OBS: Caso não tenha o Docker configurado, siga as instruções para a sua versao de SO: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04

1 - Baixe o arquivo lab04.tgz do Colabweb e descompacte no seu diretório de trabalho.

2 - Rode *npm install* e depois *npm run start.* Abra um navegador na porta 8080 para confirmar que o projeto está rodando



3 - Com o seu Docker configurado, faça o build da aplicação

docker build . --tag web\_app

4 - Na sequência, rode a imagem criada no passo anterior - se tudo estiver certo, deve funcionar do mesmo jeito que a aplicação local - use o comando abaixo para rodar a imagem docker

docker run -it -p 8080:8080 web\_app

5 - Crie um arquivo .gitignore no seu projeto e configure para ignorar node\_modules

6 - Note que o arquivo .github/workflows/wa-google-cloudrun-docker.yml ja está disponivel, mas vamos fazer algumas modificações nele em passos mais a frente.

7 - Crie um repositório público em sua conta no gitHub (como fizemos no lab anterior)

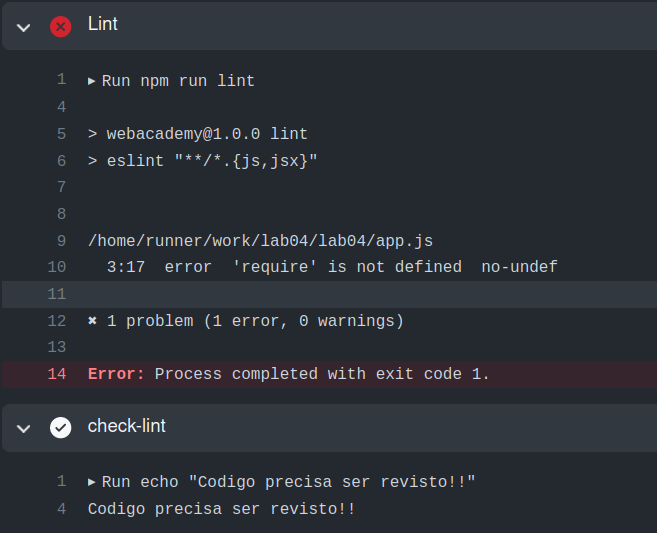
8 - Inicialize o seu projeto com o git, adicione todos os arquivos ao controle e faça o primeiro commit.

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

git remote add origin <link do seu repositório>

git push --set-upstream origin master

10 - O seu projeto vai subir no repositório recém criado, porém, no passo lint, um erro vai ser detectado! Note que o passo seguinte acontece dado o uso do condicional de observação para a falha que está configurado no pipeline.

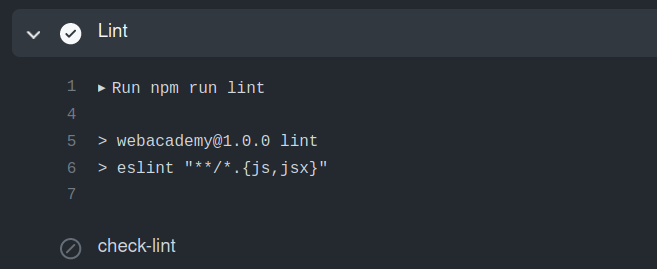


12 - Adicione a seguinte linha no início do arquivo app.js para a correção:

/\* eslint-env node \*/

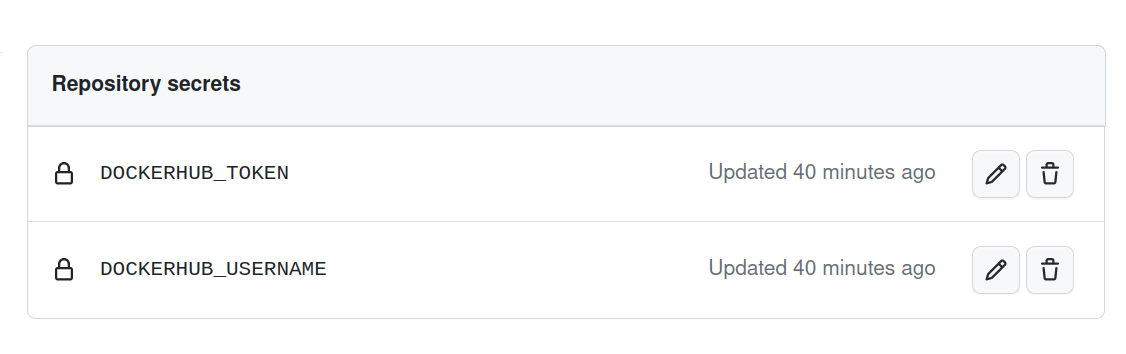
commite e suba novamente!

Dessa vez, o lint vai passar sem erros e note que o passo check-lint não é mais executado!



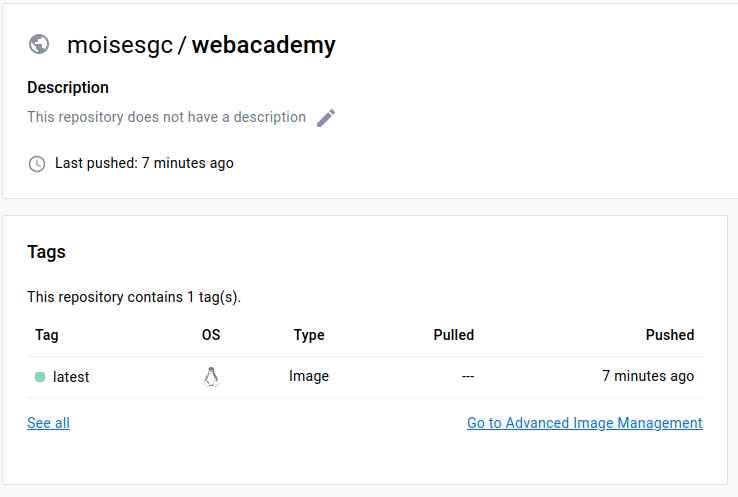
Como não configuramos nenhum segredo até agora, somente o passo do lint está funcionando.

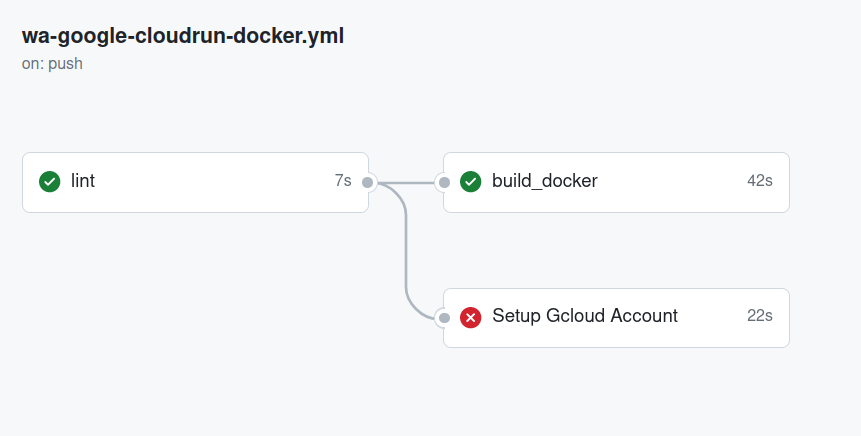
13 - Vá na seção de configuração do seu repositório e adicione as suas credenciais do DockerHub (se não tiver, vá em <https://hub.docker.com/> e crie a sua conta e gere a chave)



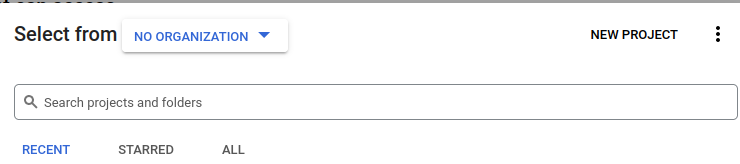
\* com a sua conta do DockerHub criada, atualize o arquivo docker-compose.yml e o arquivo wa-google-cloudrun-docker.yml com o nome do seu usuário do DockerHub nas configurações da tag da imagem.

14 - Rode novamente o pipeline e veja que o build e deploy do docker no DockerHub acontece sem problemas… mas ainda temos que ajustar as configurações necessárias no google cloud.

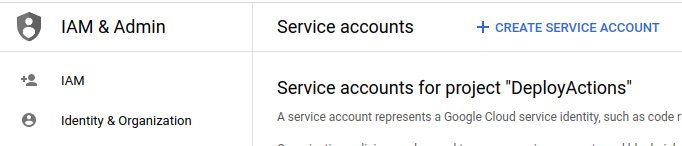




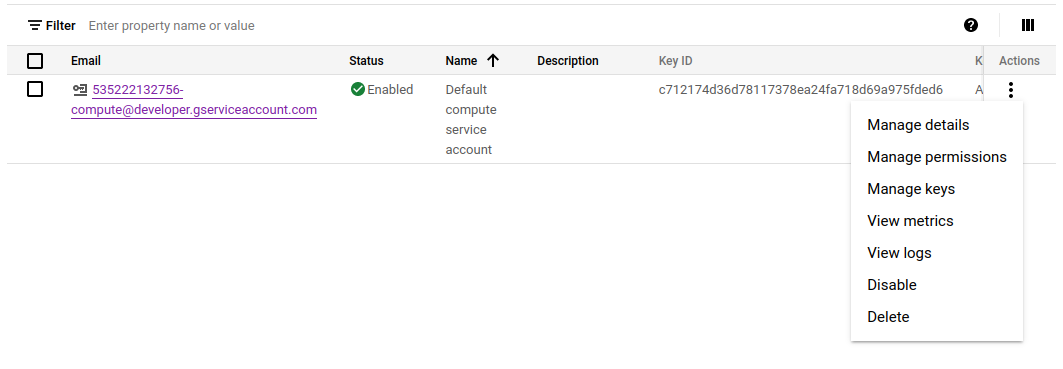
15 - Assumindo que você já tem uma conta no google cloud, entre no seu console e crie um novo projeto



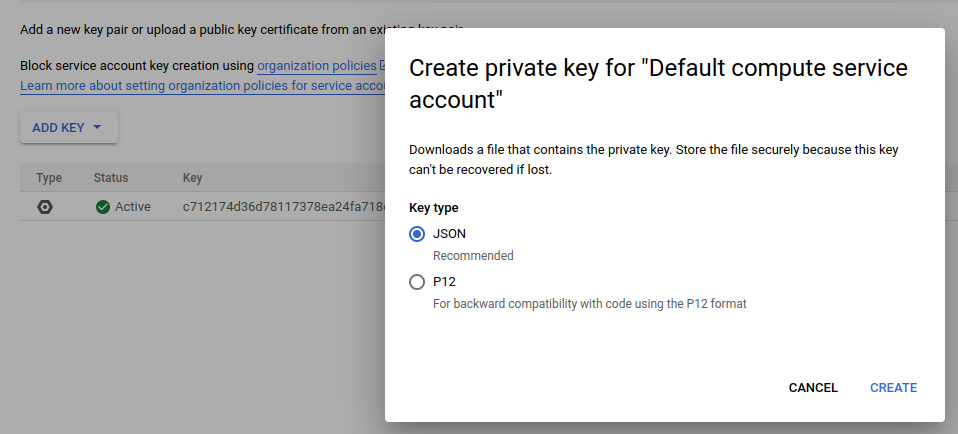
16 - Na aba do IAM, crie uma conta de serviço para o seu projeto



17 - Na conta recem-criada, vá em Manage key para criar uma chave de acesso ao seu Cloud Run

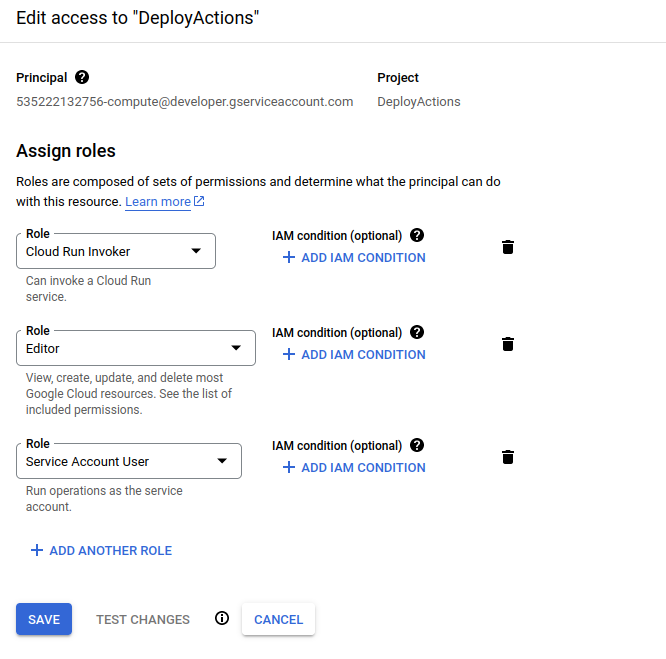


e crie uma chave JSON para o seu acesso



18 - Baixe a chave JSON , ela vai ser utilizada para configurar o acesso do gitHub Actions.

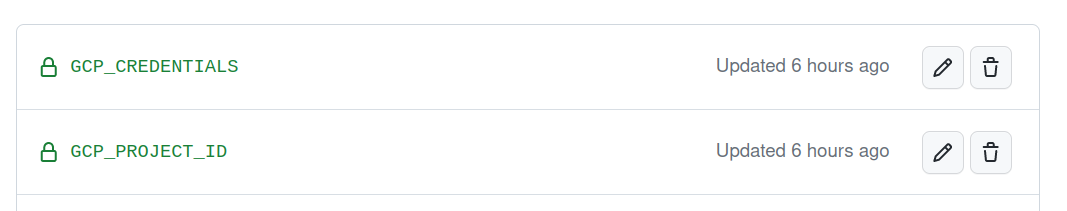
19 - Ainda na configuração da conta de serviço, adicione os seguintes roles para o seu funcionamento



20 - Se não estiver habilitado, ative o Cloud Run e o Artifact Registry!

21 - No GitHub Actions, vá em Settings -> Environments e crie um environment chamado wagcp\_deploy

22 - Neste Environment, crie 2 segredos



No primeiro, coloque no conteúdo a chave JSON que foi baixada no passo 18. No segundo segredo, basta colocar no id do projeto que você criou no passo 15.

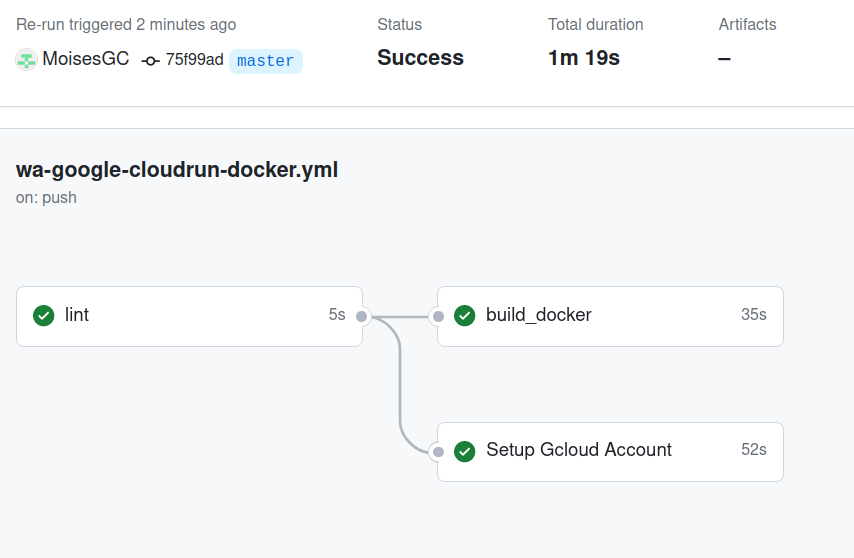
É importante que que você alinhe as configurações de região que você habilitou no seu Cloud Run com a última linha de comando do arquivo .yml que gerencia o pipeline - no arquivo enviado, está us-west1

# deploy image

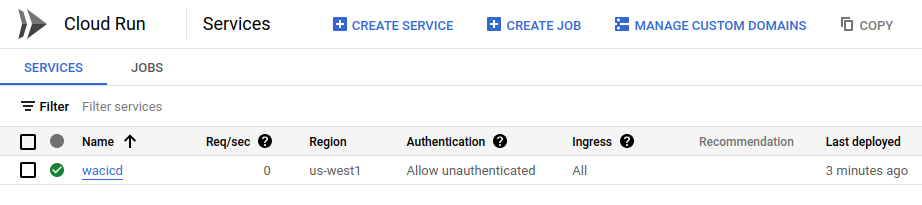
- name: Deploy Docker image

run: gcloud run deploy wacicd --image $IMAGE\_NAME --region us-west1 --memory 128Mi --min-instances 0 --max-instances 1 --platform managed --port 8080 --allow-unauthenticated

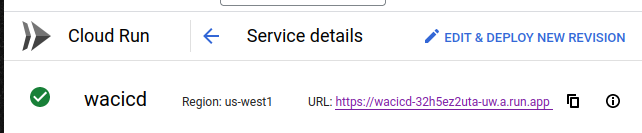
23 - Rode novamente o pipeline e tudo deve funcionar em erros

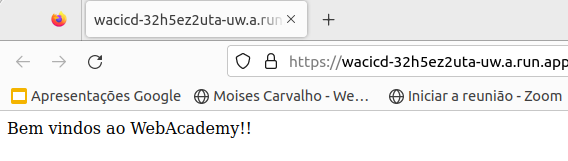


24 - Volte na aba Cloud Run do Google Console e clique no seu serviço recém deployed!



25 - Na pagina do serviço, clique na URL para acessar o serviço na infraestrutura do google!!





Parabéns, você fez deploy automatizado na infraestrutura do Google!!

PS: Se você receber um alerta de que a URL nao tem autorização para ser acessada externamente, volte ao console do Cloud Run e habilite as seguintes permissões!