Isabella Sofia Martins - 10420398

Jennifer Tondade - 10420574

- Link do repositório no Github: ProjetoNuvem
- Link para o vídeo de explicação: Vídeo no Youtube

(Parte 1) Documentação do desenvolvimento do Front e Back-end

O seguinte documento apresenta as ferramentas utilizadas e passo-a-passo de como desenvolver e consumir a aplicação web interativa que organiza informações essenciais sobre cuidados com bebês.

A aplicação possui o frontend em HTML/CSS/JS, enquanto o backend em Node.js/Express, que consome uma API de checklist de itens para bebês. A conteinerização foi feita separadamente através de docker files presentes em cada pasta.

Dokerfile front-end (configurações):

Usar Nginx para servir a página

• FROM nginx:alpine

Copiar todos os arquivos do frontend para o Nginx

• COPY . /usr/share/nginx/html

Expor porta 8080

EXPOSE 8080

Dockerfile back-end (configurações):

Base Node.js

• FROM node:18

Diretório de trabalho

WORKDIR /app

Copiar package.json e package-lock.json

COPY package*.json ./

Instalar dependências (vai instalar cors se estiver no package.json)

RUN npm install

Copiar o restante do código

• COPY..

Expor porta 25000

EXPOSE 25000

Comando para rodar o servidor

• CMD ["node", "index.js"]

Passo a passo de como excecutar a aplicação:

Backend:

- wsl --install
- wsl --set-default-version 2
- docker version
- docker info
- docker build -t backend-bebe.
- docker run -p 25000:25000 backend-bebe

Frontend:

- cd frontend-bebe
- docker build -t frontend-bebe.
- docker run -d -p 8080:80 frontend-bebe

Front e back:

- pasta raiz ProjetoNuvem
- docker-compose up --build
- back: http://localhost:25000/api/data
- front: http://localhost:8080

(Parte 2) Documentação da Configuração no ambiente da AWS

1. Criação da VPC

O primeiro passo após a realização da conteinerização das pastas contendo o front e o backend da aplicação foi criar a VPC no ambiente AWS com as seguintes configurações:

VPC → Criar VPC

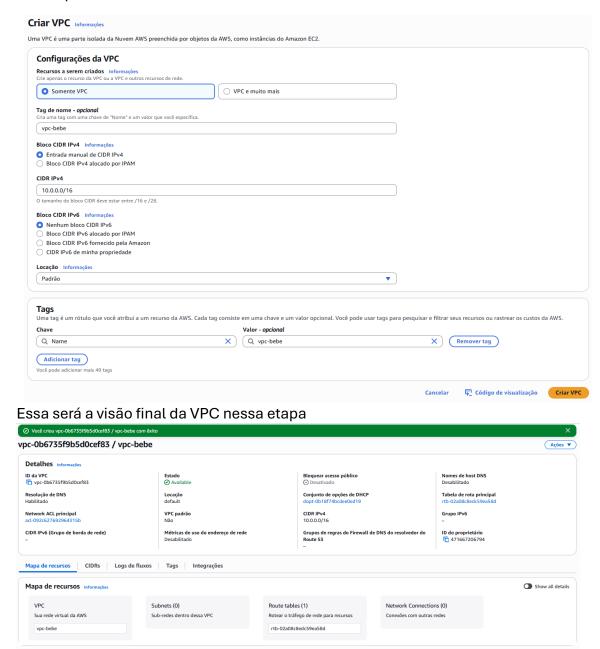
Configurações:

• Nome: vpc-bebe

• IPv4 CIDR: 10.0.0.0/16

• Locação: padrão

Clique em criar VPC



2. Criação das Sub-redes

Após criar a VPC, é necessário criar duas sub-redes e relaciona-las a VPC. Uma pública para o frontend da aplicação (poderá ser visualizada por qualquer um) e uma privada contendo o backend, em que só poderá ser acessada pela VM relacionada a ele.

Para cria-las, é necessário:

- Ir até VPC → Sub-redes → Criar Sub-rede
- Relacionar sua sub-rede com a VPC criada (vpc-bebe)



Configuração da Sub-rede 1:

Nome: subnet-front-publica

Zona de disponibilidade: use1-az6 (us-east-1a)

Bloco CIDR IPv4 da sub-rede: 10.0.1.0/24

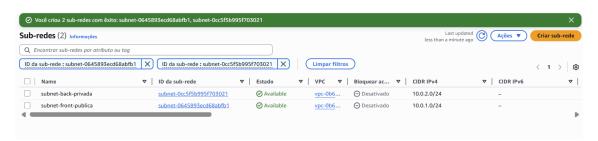
Configuração da Sub-rede 2:

Nome: subnet-back-privada

• Zona de disponibilidade: use1-az1 (us-east-1b)

Bloco CIDR IPv4 da sub-rede: 10.0.2.0/24

Após criadas, será possível observá-las como na imagem abaixo, sendo elas as selecionadas.



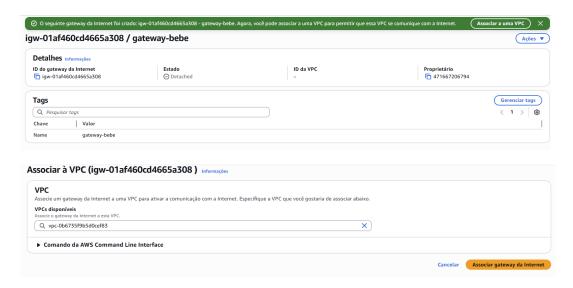
3. Criação da Internet Gateway

Temos que criar um gateway para que seja possível acessar a internet através da VPC.

VPC → Gateways da Internet → criar gateway da internet

Configurações

- Tag de nome: gateway-bebe → criar gateway
- Em seguida, clique em associar a uma VPC e escolha a vpc-bebe



4. Criação das tabelas de rota

• Tabelas de rota -> criar tabela de rota

Configurações Route Table 1:

• Nome: tab-rota-bebe-publica

VPC: vpc-bebe

Create

Configurações Route Table 2:

• Nome: tab-rota-bebe-privada

VPC: vpc-bebe

Create

4.1. Associar com sub-rede publica:

- Selecione a tabela de rota criada
- Clique em Ações → editar associações de sub-rede → selecionar a sub-rede "subnet-front-publica" → salvar associações.
- Clique em Ações → editar rotas → Destino: "0.0. 0.0/0" → Alvo: Gateway da Internet → selecione "gateway-bebe" → Salvar alterações;



4.2. Associar com sub-rede privada:

- Selecione a table de rota criada
- Clique em Acoes → Editar associação de sub-redes → selecionar a sub-rede "subnet-back-privada" → salvar associações.
- Para a rota privada não é necessário configurar a rota para a internet.

Visao final das rotas criadas:



Visão final da VPC (workflow):



5. Criação das instâncias EC2

Para esta aplicação iremos criar duas instancias EC2 (uma para acessar o front publicamente, e outra para acessar o backend de maneira privada e protegida).

Não se esqueça de salvar o aquivo da sua chave, pois será utilizada nos próximos passos.

Criação da instancia publica - front-end:

- EC2 → Instancias → Lançar instancia
- Configurações:

Nome: ec2-front-bebe

o AMI: Amazon Linux 2

o Instancia: t2.micro

o Par de chaves: default - vockey.pem

Configuracoes de rede:

o VPC: vpc-bebe

o Sub-rede: subnet-front-publica

o Atribuir IP público automaticamente: habilitar

o Nome do grupo de segurança: gs-bebe-front

Regras do grupo de segurança de entrada:

Tipo: Todo o trafego

o Intervalo de portas: 8080

o Tipo de origem: qualquer lugar

Execute a instancia;

Criação da instancia privada - back-end:

- EC2 → Instancias → Lançar instancia
- Configurações:

Nome: ec2-back-bebe

o AMI: Amazon Linux 2

o Instancia: t2.micro

o Par de chaves: default - vockey

Configuracoes de rede:

o VPC: vpc-bebe

Sub-rede: subnet-back-privada

Atribuir IP público automaticamente: desabilitar

o Nome do grupo de segurança: gs-back-bebe

Regras do grupo de segurança de entrada:

Tipo: TCP personalizado

o Intervalo de portas: 25000

o Tipo de origem: personalizado

Origem 10.0.1.25/32 (IpV4 privado da instância do front-end)



Execute a instância;



- Conectar com a instancia publica (ec2-front-bebe)
- Clique na instancia → Conectar → Conecte-se usando um IP publico →
 Conectar

onexão de instância do EC2	Gerenciador de sessões	Cliente SSH	Console d	e série do EC2
da instância				
i-0461a387d63340b8c (ec2-fron	t-bebe)			
po de conexão				
 Conecte-se usando um IP público Conecte-se usando um endereço IPv4 ou IPv6 público 				Conecte-se usando um IP privado Conecte-se usando um endereço IP privado e um endpoint de VPC
Endereço IPv4 público				
13.217.33.60				
Endereço IPv6				
me de usuário	enda nava iniciar a inetfineia. En uneŝ ni	So definiu um nomo de un	cufula nacranallis	rado, use o nome de usuário padrão, ec2-user.
Q. ec2-user	X	so dennia am nome de a	suario personana	aud, use o nume de usuano paurad, ecz-user.
Q ecz-user	^			
 Observação: na maioria dos ca AMI padrão. 	isos, o nome de usuario padrão, e	c2-user, està correto.	No entanto, le	ia as instruções de uso da AMI para verificar se o proprietário da AMI alterou o nome de usuário da

Sua instancia terá essa imagem:



O primeiro passo para a conexão é verificar se possui Docker na VM:

Docker - - version

Se não houver, precisamos fazer a instalação digitando os códigos:

- sudo yum update -y
- sudo yum install -y docker
- sudo systemctl enable docker
- sudo systemctl start docker
- sudo usermod -ag docker ec2-user

6. Subir pasta com o front na VM

Agora, é necessário acessar o repositório com o front e o backend. Para isso, iremos clonar nosso repositório que está hospedado no github, mas também é possível conectar com a pasta localmente.

Para isso, primeiro é necessário verificar se temos git instalado:

- sudo yum update -y
- sudo yum install -y git
- git --version

Para clonar o repositório:

- cd ~
- git clone https://github.com/JenniferTondade/ProjetoNuvem.git

DICA: ao acessar a pasta desejada – neste caso, a que contém o frontend – sempre consulte o que há dentro do diretório antes com o comando "dir", pois pode haver uma segunda pasta. Veja:

```
Complete!

[ec2-user@ip-10-0-1-25 ~]$ git --version

git version 2.50.1

[ec2-user@ip-10-0-1-25 ~]$ cd ~

[ec2-user@ip-10-0-1-25 ~]$ git clone https://github.com/JenniferTondade/ProjetoNuvem.git

Cloning into 'ProjetoNuvem'...

remote: Enumerating objects: 100% (1071/1071), done.

remote: Counting objects: 100% (1071/1071), done.

remote: Compressing objects: 100% (829/829), done.

remote: Total 1071 (delta 173), reused 1064 (delta 171), pack-reused 0 (from 0)

Receiving objects: 100% (1071/1071), 9.64 MiB | 32.92 MiB/s, done.

Resolving deltas: 100% (173/173), done.

[ec2-user@ip-10-0-1-25 ~]$ dir

ProjetoNuvem

[ec2-user@ip-10-0-1-25 ~]$ cd ProjetoNuvem

[ec2-user@ip-10-0-1-25 ProjetoNuvem]$ dir

README.md backend-bebe docker-compose.yml frontend-bebe node_modules package-lock.json package.json

[ec2-user@ip-10-0-1-25 ProjetoNuvem]$ cd frontend-bebe

[ec2-user@ip-10-0-1-25 ProjetoNuvem]$ cd frontend-bebe
```

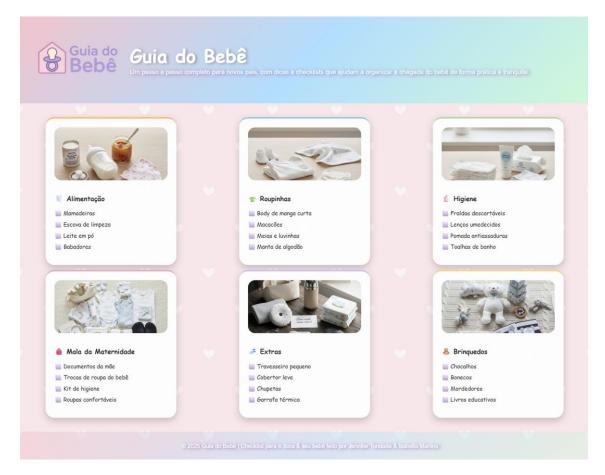
Ao chegar na pasta frontend-bebe, através dos comandos

- cd ProjetoNuvem
- cd frontend-bebe

Iremos rodar agora a imagem do Docker que esta dentro do Docker file no frontend.

- 7. Rodar e testar o container
- sudo docker build -t frontend-bebe.
- sudo docker run -d -p 8080:80 frontend-bebe
- no bash: curl http://localhost:8080
- no navegador: http://<IP-PUBLICO-DA-VM>:8080

Você deverá conseguir acessar o site na web:



8. Testar o acesso ao backend

Para dar acesso do backend apenas para quem tem o grupo de segurança, precisamos fazer um jumper com SSH para acessá-lo através da instancia publica, já que o backend não possui gateway e sua uma conexão de entrada é o ipv4 da instancia pública.

No CMD do seu computador:

Execute o comando abaixo substituindo o caminho para o que leva ao seu arquivo que contém a chave de segurança. O comando abaixo irá copiar sua chave no ambiente local e passar para o ambiente em nuvem (VM) para que seja possível acessar a instancia privada.

 scp -i "C:\Users\Isabella Sofia\Downloads\vockey.pem" "C:\Users\Isabella Sofia\Downloads\vockey.pem" ec2-user@<IP-PUBLICO-FRONTEND>:/home/ec2-user/

Na VM da AWS:

- ssh -i ~/vockey.pem ec2-user@10.0.1.239 (ip publico da VM publica)
 pode ter um erro por excesso de permissions, então digite:
- chmod 400 ~/vockey.pem

 ssh -i ~/vockey.pem ec2-user@10.0.2.4 //endereco privado da instancia publica

conseguimos entrar na instancia privada:

Para rodar o backend com nossa API de checklist:

- git clone https://github.com/JenniferTondade/ProjetoNuvem.git backend
- cd backend
- cd backend-bebe
- sudo docker build -t backend.
- sudo docker run -d -p 25000:25000 backend
- curl http://10.0.1.239:25000/api/data (IPv4 privado da VM pública)

```
[ec2-user@ip-10-0-1-239 backend-bebe]$ curl http://10.0.1.239:25000
c(IDCTYPE html>
chead>
cmeta charset="utf-8">
c(itle>Error</title>
c/head>
cmeta charset="utf-8">
c(itle>Error</title>
c/head>
cody>
cyro>Cannot GET /
c/head>
c/hoady
c/head>
c/hoady
c/head>
c/hea
```

Repare que ao tentar acessar a aplicação pela internet através do HTTP, não irá ter retorno - esse backend está em uma instancia conectada a sub-redes sem um gateway de internet e unicamente atrelado com a rede privada da instância

pública, então só é possível acessar o conteúdo da API através da conexão com a instancia do front-end primeiro.

- ponto de atenção ao executar: garanta que ao rodar a aplicação em suas instâncias, e front e o back não estejam rodando nas portas direcionadas, pois pode provocar um erro ao tentar conectar utilizando o Docker. Nestes casos, realize os seguintes comandos no CMD:
 - o Docker ps
 - O Docker stop < container id >