

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 containerização foi feita separadamente através de docker files presentes em cada pasta.

Dockerfile 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 executar 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 containerização 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
- Localização: padrão

- Clique em criar VPC

Criar VPC

Informações

Uma VPC é uma parte isolada da Nuvem AWS preenchida por objetos da AWS, como instâncias do Amazon EC2.

Configurações da VPC

Recursos a serem criados

Informações

Crie apenas o recurso da VPC ou a VPC e outros recursos de rede.

☒ Somente VPC

☐ VPC e muito mais

Tag de nome - opcional

Cria uma tag com uma chave de "Nome" e um valor que você especifica.

vpc-bebe

Bloco CIDR IPv4

Informações

☒ Entrada manual de CIDR IPv4

☐ Bloco CIDR IPv4 alocado por IPAM

CIDR IPv4

10.0.0.0/16

O tamanho do bloco CIDR deve estar entre /16 e /28.

Bloco CIDR IPv6

Informações

☒ Nenhum bloco CIDR IPv6

☐ Bloco CIDR IPv6 alocado por IPAM

☐ Bloco CIDR IPv6 fornecido pela Amazon

☐ CIDR IPv6 de minha propriedade

Localização

Informações

Padrão

Tags

Uma tag é um rótulo que você atribui a um recurso da AWS. Cada tag consiste em uma chave e um valor opcional. Você pode usar tags para pesquisar e filtrar seus recursos ou rastrear os custos da AWS.

Chave

Valor - opcional

X

X

Remover tag

Adicionar tag

Você pode adicionar mais 49 tags

Cancelar

Código de visualização

Criar VPC

Essa será a visão final da VPC nessa etapa

Você criou vpc-0b6735f9b5d0cef83 / vpc-bebe com êxito

vpc-0b6735f9b5d0cef83 / vpc-bebe

Ações

Detalhes

Informações

ID da VPC

vpc-0b6735f9b5d0cef83

Resolução de DNS

Habilitado

Network ACL principal

acl-092c627692964315b

CIDR IPv6 (Grupo de borda de rede)

–

Estado

Available

Localção

default

VPC padrão

Não

Métricas de uso do endereço de rede

Desabilitado

Bloquear acesso público

Desativado

Conjunto de opções de DHCP

dhcp-0b18f74bcdce0ed19

CIDR IPv4

10.0.0.0/16

Grupos de regras do Firewall de DNS do resolvidor do Route 53

–

Nomes de host DNS

Desabilitado

Tabela de rota principal

rtb-02a08c8edc59ea58d

Grupo IPv6

–

ID do proprietário

471667206794

Mapa de recursos

CIDRs

Logs de fluxos

Tags

Integrações

Mapa de recursos

Informações

Show all details

VPC

Sua rede virtual da AWS

vpc-bebe

Subnets (0)

Sub-redes dentro dessa VPC

Route tables (1)

Rotear o tráfego de rede para recursos

rtb-02a08c8edc59ea58d

Network Connections (0)

Conexões com outras redes

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)

Criar sub-rede [Informações](#)

VPC

ID da VPC
Crie sub-redes nessa VPC.

Selecionar uma VPC

Q |

vpc-064852cf03ce79f91 172.31.0.0/16	(padrão)
vpc-050b9faee2ac621c1 (vpc-bebe) 10.0.0.0/16	
vpc-0b4c68c5e42e928d7 (Work VPC) 10.0.0.0/16	

Adicionar nova sub-rede

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

Configuracao 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.

✓ Você criou 2 sub-redes com êxito: subnet-0645893ecd68abfb1, subnet-0cc5f5b995f703021

Sub-redes (2) [Informações](#)

Encontrar sub-redes por atributo ou tag

ID da sub-rede: subnet-0645893ecd68abfb1 X ID da sub-rede: subnet-0cc5f5b995f703021 X Limpar filtros

<input type="checkbox"/>	Name	ID da sub-rede	Estado	VPC	Bloquear ac...	CIDR IPv4	CIDR IPv6
<input type="checkbox"/>	subnet-back-privada	subnet-0cc5f5b995f703021	Available	vpc-0b6...	Desativado	10.0.2.0/24	-
<input type="checkbox"/>	subnet-front-publica	subnet-0645893ecd68abfb1	Available	vpc-0b6...	Desativado	10.0.1.0/24	-

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**

🔔 O seguinte gateway da Internet foi criado: igw-01af460cd4665a308 - gateway-bebe. Agora, você pode associar a uma VPC para permitir que essa VPC se comunique com a Internet. [Associar a uma VPC](#) ✕

igw-01af460cd4665a308 / gateway-bebe [Ações](#)

Detalhes [Informações](#)
ID do gateway da Internet
igw-01af460cd4665a308

Estado
Detached

ID da VPC
-

Proprietário
471667206794

Tags [Gerenciar tags](#)

Chave	Valor
Name	gateway-bebe

Associar à VPC (igw-01af460cd4665a308) [Informações](#)

VPC
Associe um gateway da Internet a uma VPC para ativar a comunicação com a Internet. Especifique a VPC que você gostaria de associar abaixo.

VPCs disponíveis
Associe o gateway da Internet a esta VPC.

 ✕

► Comando da AWS Command Line Interface

[Cancelar](#) [Associar gateway da Internet](#)

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;

Editar rotas

Destino 10.0.0.0/16	Alvo local	Status Ativo	Propagado Não	Route Origin CreateRouteTable
<input type="text" value="0.0.0.0"/>	<input type="text" value="local"/>	-	Não	CreateRoute
	<input type="text" value="igw-01af460cd4665a308"/>			

[Adicionar rota](#)

[Cancelar](#) [Visualização](#) [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:

Tabelas de rotas (2/6) Informações

Q. Encontrar tabelas de rotas por atributo ou tag

Last updated 3 minutes ago Ações Criar tabela de rotas

	Name	ID da tabela de rotas	Associações explícitas de sub-rede	Associações de ...	Principal	VPC
<input type="checkbox"/>	Work Public Route Table	rtb-075f57384121b1f4	subnet-0478c52d587f730da / Work Public Subnet	-	Não	vpc-093354dad0343ee79 Work V
<input type="checkbox"/>	-	rtb-026e27904c66b41c5	-	-	Sim	vpc-0589a23422f780300
<input type="checkbox"/>	-	rtb-0e2095fc999e9f695	-	-	Sim	vpc-093354dad0343ee79 Work V
<input type="checkbox"/>	-	rtb-02a08c8edc59ea58d	-	-	Sim	vpc-0b6735f9b5d0cef83 vpc-bebe
<input checked="" type="checkbox"/>	tab-rota-bebe-publica	rtb-0106c2a5bda22c785	-	-	Não	vpc-0b6735f9b5d0cef83 vpc-bebe
<input checked="" type="checkbox"/>	tab-rota-bebe-privada	rtb-0ada8d13f0b064f3d	subnet-0cc5f5b995f703021 / subnet-back-privada	-	Não	vpc-0b6735f9b5d0cef83 vpc-bebe

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 arquivo da sua chave, pois será utilizada nos próximos passos.

Criação da instancia publica – front-end:

- EC2 → Instancias → Lançar instancia
- Configurações:
 - Nome: ec2-front-bebe
 - AMI: Amazon Linux 2
 - Instancia: t2.micro
 - Par de chaves: default – vockey.pem

Configuracoes de rede:

- VPC: vpc-bebe
- Sub-rede: subnet-front-publica
- Atribuir IP público automaticamente: habilitar
- Nome do grupo de segurança: gs-bebe-front

Regras do grupo de segurança de entrada:

- Tipo: Todo o trafego
- Intervalo de portas: 8080
- Tipo de origem: qualquer lugar

- **Execute a instancia;**

Criação da instancia privada – back-end:

- EC2 → Instancias → Lançar instancia
- **Configurações:**
 - **Nome: ec2-back-bebe**
 - **AMI: Amazon Linux 2**
 - **Instancia: t2.micro**
 - **Par de chaves: default – vockey**

Configuracoes de rede:

- **VPC: vpc-bebe**
- **Sub-rede: subnet-back-privada**
- **Atribuir IP público automaticamente: desabilitar**
- **Nome do grupo de segurança: gs-back-bebe**

Regras do grupo de segurança de entrada:

- **Tipo: TCP personalizado**
- **Intervalo de portas: 25000**
- **Tipo de origem: personalizado**
- **Origem 10.0.1.25/32 (IPv4 privado da instância do front-end)**

Regras do grupo de segurança de entrada

▼ Regra de grupo de segurança 1 (TCP, 25000, 10.0.1.24/32)

Remover

Tipo | Informações

TCP personalizado

Protocolo | Informações

TCP

Intervalo de portas | Informações

25000

Tipo de origem | Informações

Personalizado

Origem | Informações

Q Adicionar CIDR, lista de prefixos ou grupo de s

10.0.1.24/32 X

Descrição (opcional) | Informações

p. ex. SSH para a área de trabalho do administrac

- **Execute a instância;**

Instâncias (2/3) Informações									
Q Localizar instância por atributo ou tag (case-sensitive)									
Todos os ...									
Name	ID da instância	Estado da inst...	Tipo de inst...	Verificação de star	Status do alarm	Zona de dispon...	DNS IPv4 público	Endereço IP...	IP elástico
ec2-back-bebe	i-0128de8fc7857b45b	Executando	t2.micro	–	Exibir alarmes +	us-east-1b	–	–	–
Bastion Host	i-080b5b72c870074db	Executando	t2.micro	2/2 verificações a	Exibir alarmes +	us-east-1a	ec2-54-224-216-167.co...	54.224.216.167	–
ec2-front-bebe	i-04815f2da78678d27	Executando	t2.micro	2/2 verificações a	Exibir alarmes +	us-east-1a	–	54.229.244.171	–

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

Conectar Informações

Conecte-se a uma instância usando o cliente baseado em navegador.

Conexão de instância do EC2 | Gerenciador de sessões | Cliente SSH | Console de série do EC2

ID da instância
 i-0461a387d63340b8c (ec2-front-bebe)

Tipo 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
 -

Nome de usuário
 Insira o nome de usuário definido na AMI usada para iniciar a instância. Se você não definiu um nome de usuário personalizado, use o nome de usuário padrão, ec2-user.

ec2-user

Observação: na maioria dos casos, o nome de usuário padrão, ec2-user, está correto. No entanto, leia as instruções de uso da AMI para verificar se o proprietário da AMI alterou o nome de usuário da AMI padrão.

[Cancelar](#) [Conectar](#)

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: 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 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:



Guia do Bebê

Um passo a passo completo para novos pais, com dicas e checklists que ajudam a organizar a chegada do bebê de forma prática e tranquila.



Alimentação

- Mamadeiras
- Escova de limpeza
- Leite em pó
- Babadores



Roupinhas

- Body de manga curta
- Macacões
- Meias e luvinhas
- Manta de algodão



Higiene

- Fraldas descartáveis
- Lenços umedecidos
- Pomada antiassaduras
- Toalhas de banho



Mala da Maternidade

- Documentos da mãe
- Trocas de roupa do bebê
- Kit de higiene
- Roupas confortáveis



Extras

- Travesseiro pequeno
- Cobertor leve
- Chupetas
- Garrafa térmica



Brinquedos

- Chocalhos
- Bonecos
- Mordedores
- Livros educativos

© 2025 Guia do Bebê | Checklist para o Joca & seu bebê feito por Jennifer Tondade & Isabella Martins

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 //endereço privado da instancia publica`

```
this key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.239' (ED25519) to the list of known hosts.
ec2-user@10.0.1.239: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-239 frontend-bebe]$ ssh -i ~/vockey.pem ec2-user@10.0.1.239
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for '/home/ec2-user/vockey.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/ec2-user/vockey.pem": bad permissions
ec2-user@10.0.1.239: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-239 frontend-bebe]$ chmod 400 ~/vockey.pem
[ec2-user@ip-10-0-1-239 frontend-bebe]$ ssh -i ~/vockey.pem ec2-user@10.0.1.239

      #
    ~\_ #####          Amazon Linux 2023
    ~~\_#####\
    ~~ \####|
    ~~  \##/
    ~~~ \#/           https://aws.amazon.com/linux/amazon-linux-2023
        v~' '->
         ~~~
          ~~-.-.-
            /  _/
             m/'
```

Last login: Fri Sep 26 15:23:52 2025 from 177.81.73.222
[ec2-user@ip-10-0-1-239 ~]\$ █

- **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)**

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:
 - **Docker ps**
 - **Docker stop <container id>**