## Guia de Instalação e Deployment do Sistema de Precificação e CRM em Oracle Cloud

Este guia detalha as etapas para instalar e implantar o sistema de precificação e CRM na Oracle Cloud Free Tier, utilizando uma arquitetura de contêineres com Docker.

### 1. Visão Geral da Arquitetura

O sistema é composto por dois componentes principais:

* **Backend (Flask):** Uma API RESTful que gerencia a lógica de precificação, dados de materiais, fatores de dificuldade, gestão de clientes, projetos e integrações com IA.
* **Frontend (React):** Uma aplicação web que fornece a interface do usuário para franqueados, permitindo a interação com a calculadora de preços, CRM e funcionalidades de IA.

Para a implantação na Oracle Cloud, utilizaremos a seguinte arquitetura:

* **Computação:** Uma instância de VM ARM-based (Ampere A1) na Always Free Tier, que oferece recursos generosos (4 OCPUs e 24 GB de RAM) para executar nossos contêineres Docker.
* **Banco de Dados:** Oracle Autonomous Database (PostgreSQL) na Always Free Tier, que oferece um banco de dados gerenciado e de alto desempenho.
* **Rede:** Uma Virtual Cloud Network (VCN) com sub-redes públicas e privadas para garantir a segurança e a conectividade dos nossos serviços.
* **Contêineres:** Docker para empacotar e executar o backend e o frontend da aplicação.

### 2. Configuração do Ambiente Oracle Cloud

Antes de iniciar a implantação, é necessário configurar o ambiente na Oracle Cloud.

#### 2.1. Criar uma Conta Oracle Cloud Free Tier

1. Acesse o site da [Oracle Cloud Free Tier](https://www.oracle.com/cloud/free/) e crie uma nova conta.
2. Siga as instruções para verificar sua identidade e configurar sua conta.

#### 2.2. Criar um Compartimento

1. No console da Oracle Cloud, navegue até **Identity & Security** > **Compartments**.
2. Clique em **Create Compartment** e crie um novo compartimento para organizar os recursos do projeto (ex: crm\_precificacao\_compartment).

#### 2.3. Gerar um Par de Chaves SSH

1. Gere um par de chaves SSH para acessar a VM. Você pode usar o comando ssh-keygen no Linux/macOS ou o PuTTYgen no Windows.

ssh-keygen -t rsa -b 2048 -f ~/.ssh/oracle\_cloud\_key

1. Guarde a chave privada (oracle\_cloud\_key) em um local seguro e anote o conteúdo da chave pública (oracle\_cloud\_key.pub).

### 3. Implantação do Backend (Flask com Docker)

Nesta etapa, vamos provisionar a VM, instalar o Docker e implantar o backend da aplicação.

#### 3.1. Criar uma Instância de VM ARM-based

1. No console da Oracle Cloud, navegue até **Compute** > **Instances**.
2. Clique em **Create Instance** e configure a VM com as seguintes opções:
   * **Name:** backend-vm
   * **Compartment:** Selecione o compartimento criado anteriormente.
   * **Image and shape:**
     + **Image:** Ubuntu
     + **Shape:** VM.Standard.A1.Flex (ARM-based)
     + **OCPU count:** 4
     + **Amount of memory (GB):** 24
   * **Networking:**
     + Selecione uma VCN e uma sub-rede pública.
     + Marque a opção **Assign a public IPv4 address**.
   * **Add SSH keys:** Cole o conteúdo da sua chave pública SSH.
3. Clique em **Create** para provisionar a VM.

#### 3.2. Configurar Regras de Segurança (Ingress Rules)

1. Navegue até a VCN associada à sua VM e clique em **Security Lists**.
2. Selecione a lista de segurança padrão e adicione as seguintes Ingress Rules:
   * **Source CIDR:** 0.0.0.0/0
   * **IP Protocol:** TCP
   * **Destination Port Range:** 22 (para SSH), 80 (para HTTP), 443 (para HTTPS), 5000 (para o backend Flask).

#### 3.3. Instalar Docker e Docker Compose

1. Conecte-se à VM via SSH usando a chave privada e o IP público da instância:

ssh -i ~/.ssh/oracle\_cloud\_key ubuntu@<IP\_PUBLICO\_DA\_VM>

1. Instale o Docker:

sudo apt-get update

sudo apt-get install -y docker.io

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -aG docker ubuntu

1. Instale o Docker Compose:

sudo apt-get install -y docker-compose

1. Faça logout e login novamente para aplicar as alterações de grupo do Docker.

#### 3.4. Implantar o Backend

1. Clone o repositório do backend na VM:

git clone <URL\_DO\_REPOSITORIO\_BACKEND> franchise\_pricing\_crm

cd franchise\_pricing\_crm

1. Crie um arquivo .env na raiz do projeto com as variáveis de ambiente necessárias:

FLASK\_APP=src/main.py

FLASK\_ENV=production

DATABASE\_URL=<URL\_DO\_BANCO\_DE\_DADOS\_AUTONOMOUS>

OPENAI\_API\_KEY=<SUA\_CHAVE\_DA\_OPENAI>

1. Construa e execute o contêiner Docker:

docker-compose up --build -d

### 4. Implantação do Frontend (React)

Para o frontend, vamos construir a aplicação React e servi-la com o Nginx em um contêiner Docker.

#### 4.1. Configurar o Nginx

1. Crie um arquivo nginx.conf na raiz do projeto do frontend com a seguinte configuração:

server {

listen 80;

server\_name <IP\_PUBLICO\_DA\_VM>;

location / {

root /usr/share/nginx/html;

index index.html index.htm;

try\_files $uri $uri/ /index.html;

}

location /api/ {

proxy\_pass http://<IP\_PRIVADO\_DA\_VM\_BACKEND>:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

1. Substitua <IP\_PUBLICO\_DA\_VM> e <IP\_PRIVADO\_DA\_VM\_BACKEND> pelos IPs corretos.

#### 4.2. Criar um Dockerfile para o Frontend

1. Crie um Dockerfile na raiz do projeto do frontend:

# Build stage

FROM node:18-alpine as build

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . ./

RUN npm run build

# Production stage

FROM nginx:stable-alpine

COPY --from=build /app/dist /usr/share/nginx/html

COPY nginx.conf /etc/nginx/conf.d/default.conf

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

#### 4.3. Implantar o Frontend

1. Clone o repositório do frontend na mesma VM (ou em outra, se preferir):

git clone <URL\_DO\_REPOSITORIO\_FRONTEND> pricing-frontend

cd pricing-frontend

1. Construa e execute o contêiner Docker do frontend:

docker build -t pricing-frontend .

docker run -d -p 80:80 pricing-frontend

### 5. Configuração do Banco de Dados (Oracle Autonomous Database)

1. No console da Oracle Cloud, navegue até **Oracle Database** > **Autonomous Database**.
2. Clique em **Create Autonomous Database** e configure o banco de dados:
   * **Workload type:** Transaction Processing
   * **Deployment type:** Serverless
   * **Database version:** 19c
   * **OCPU count:** 1
   * **Storage (TB):** 0.02 (20 GB)
   * **Administrator credentials:** Defina um nome de usuário e senha para o administrador do banco de dados.
3. Após a criação, acesse a página de detalhes do banco de dados e clique em **DB Connection** para obter a URL de conexão.
4. Use a URL de conexão no arquivo .env do backend.

### 6. Teste e Verificação

1. Acesse o IP público da sua VM em um navegador. Você deve ver a interface do frontend.
2. Teste as funcionalidades da aplicação, como a calculadora de preços e o CRM, para garantir que a comunicação com o backend e o banco de dados está funcionando corretamente.

### 7. Referências

* [Oracle Cloud Free Tier](https://www.oracle.com/cloud/free/)
* [Documentação do Oracle Cloud Infrastructure](https://docs.oracle.com/en-us/iaas/)
* [Run Always Free Docker Container on Oracle Cloud Infrastructure](https://medium.com/oracledevs/run-always-free-docker-container-on-oracle-cloud-infrastructure-c88e36b65610)