

## **1. Orquestração de Contêineres com Kubernetes**

- Por quê? Kubernetes é uma escolha robusta para gerenciar containers, permitindo escalabilidade e resiliência. Ele pode automatizar o provisionamento, a gestão, a escala e a recuperação de falhas dos contêineres que executam os bancos de dados e os serviços do seu sistema DBaaS.

## **2. Contêineres Docker para Bancos de Dados e Aplicações**

- Por quê? Docker permite empacotar o servidor de banco de dados e as aplicações em contêineres isolados, facilitando o deployment e a portabilidade. Cada "flavor" de banco de dados pode ser preparado como uma imagem Docker que inclui o SO e o servidor de banco de dados pré-configurado.

## **3. API Gateway**

- Por quê? Um API Gateway atua como um ponto de entrada unificado para as APIs do seu sistema, simplificando a interação entre os clientes (CLI e Web Portal) e os serviços internos. Ele também pode lidar com preocupações transversais como autenticação, registro de logs e limitação de taxa.

## **4. Serviço de Compute XaaS (por exemplo, AWS EC2, Google Compute Engine)**

- Por quê? Você mencionou "Compute XaaS" para provisionamento de VMs. Usar um provedor de nuvem como AWS EC2 ou Google Compute Engine permite aproveitar APIs REST para programaticamente criar e gerenciar instâncias de VM que hospedarão os bancos de dados.

## **5. Terraform para IaC (Infraestrutura como Código)**

- Por quê? Terraform pode ser usado para automatizar o provisionamento de recursos na nuvem (incluindo VMs, redes, balanceadores de carga), garantindo que sua infraestrutura seja facilmente replicável e gerenciável como código.

## **6. Linguagens de Programação e Frameworks**

- Para o backend (DBaaS Manager), linguagens como Go, Python (com Flask ou Django) ou Node.js (com Express) são excelentes opções devido à sua eficiência, suporte a concorrência e rica biblioteca de pacotes para desenvolvimento de APIs REST.
- Para o CLI Tool, você pode usar Python (com Click ou argparse) ou Node.js, que permitem a criação de ferramentas de linha de comando robustas e portáteis.
- Para o Web Portal, tecnologias como React, Vue.js ou Angular podem ser usadas para criar uma interface de usuário rica e interativa, comunicando-se com o backend via REST API.

## **7. Banco de Dados Relacional**

- Por quê? Embora a engine específica do banco de dados não seja relevante, você precisará escolher um sistema de gerenciamento de banco de dados relacional (RDBMS) que será padronizado em todas as VMs. PostgreSQL e MySQL são escolhas populares, livres e de código aberto que podem ser facilmente containerizadas.

### **Fluxo de Funcionamento Proposto:**

Usuário interage com o Portal Web ou CLI para solicitar a criação de um banco de dados, especificando o flavor e o espaço de armazenamento.

Portal/CLI envia a solicitação para o DBaaS Manager, que valida a solicitação.

DBaaS Manager usa a API do Compute XaaS para provisionar uma VM com as especificações requisitadas.

Uma vez que a VM está pronta, o DBaaS Manager configura o servidor de banco de dados na VM, utilizando as imagens Docker pré-configuradas para o flavor escolhido.

O usuário é notificado de que o banco de dados está pronto para uso.