**O que são containers e por que usamos Docker para desenvolvimento?**

Containers são ambientes isolados que empacotam uma aplicação e suas dependências, garantindo que se tenha o mesmo resultado em diferentes ambientes.

O Docker facilita a criação, distribuição e execução desses containers em ambientes de desenvolvimento, onde é fácil rodar a aplicação seja no computador ou na nuvem.

**Qual a função do PHP-FPM e por que ele é utilizado junto ao Nginx?**

O PHP-FPM executa os scripts de PHP de forma mais eficiente, basicamente usada para gerenciar os processos que executam o código PHP.

O PHP-FPM é usado junto ao Nginx porque ele é leve, eficiente e ele lida com múltiplas requisições simultâneas de forma eficiente, ajudando o servidor a escalar.

**Qual a diferença entre um servidor Apache e um servidor Nginx?**

Enquanto o Apache é um servidor web tradicional e flexível, com muitos módulos que permitem personalizar o comportamento do servidor de várias formas, o Nginx é mais rápido e leve, usando uma arquitetura assíncrona para gerenciar múltiplas conexões ao mesmo tempo sem criar um novo processo para cada uma delas.

Vale ressaltar que o Apache tende a consumir mais recursos, especialmente em alta carga, porque cada requisição é tratada por um processo separado.

Por isso, o Nginx é super popular quando a prioridade é desempenho e alta escalabilidade, especialmente em sites de alto tráfego.

**Como o Docker organiza imagens e containers? Explique o conceito de multi-stage builds.**

O Docker usa imagens como templates para criar containers. Uma imagem é um arquivo que contém tudo o que a aplicação precisa para rodar (código, dependências, etc.), e o container é a instância em execução dessa imagem.

O processo de multi-stage builds no Docker torna o processo de criação de imagens mais eficiente, dividindo o processo de construção da imagem em vários stages dentro do mesmo Dockerfile. Cada stage pode ter seu próprio conjunto de instruções.