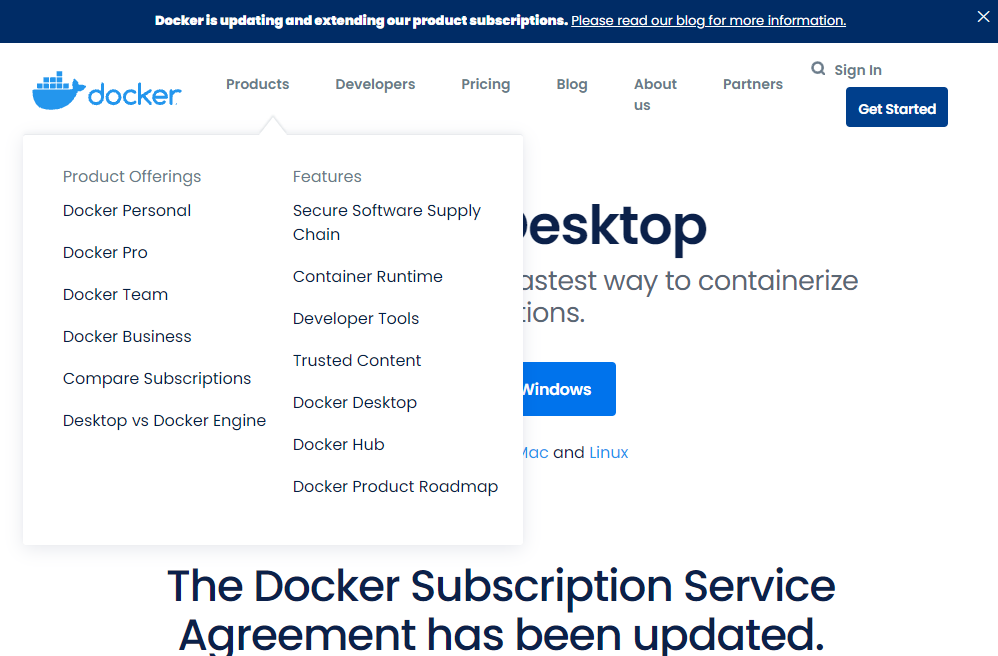
**Docker é um conjunto de produtos de plataforma como serviço que usam virtualização de nível de sistema operacional para entregar software em pacotes chamados contêineres. Os contêineres são isolados uns dos outros e agrupam seus próprios softwares, bibliotecas e arquivos de configuração.**

**A necessidade de transferir aplicações de um ambiente computacional para outro com segurança e rapidez tornou-se crítica para os negócios e times de desenvolvimento e operações. E foi materializada pelo conceito de contêiner, que tira mais esse problema da frente, encapsulando as aplicações em uma estrutura isolada e portátil.**

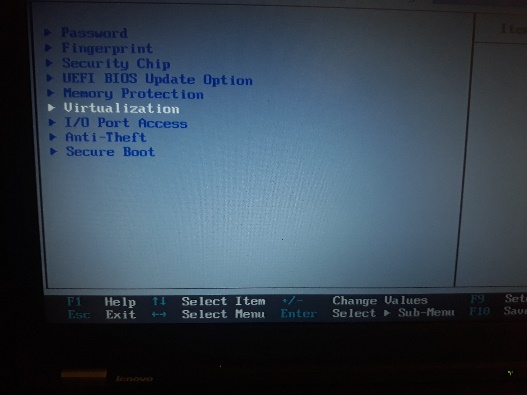
**Embora seja uma das ferramentas de conteinerização mais usadas e conhecidas, o Docker não está sozinho. Há outros fornecedores no ecossistema: ConteinerD, CoreOS, Canonical, LXC Linux Conteiners, CRIO-D e Mesos Containerizer.**

**Embora seja uma das ferramentas de conteinerização mais usadas e conhecidas, o Docker não está sozinho. Há outros fornecedores no ecossistema: ConteinerD, CoreOS, Canonical, LXC Linux Conteiners, CRIO-D e Mesos Containerizer.**

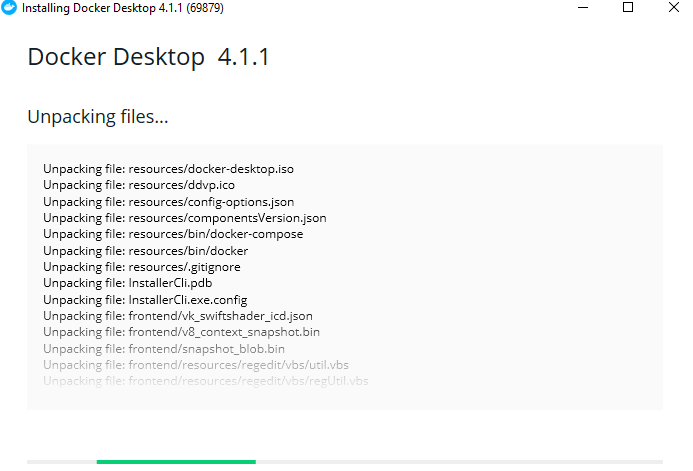
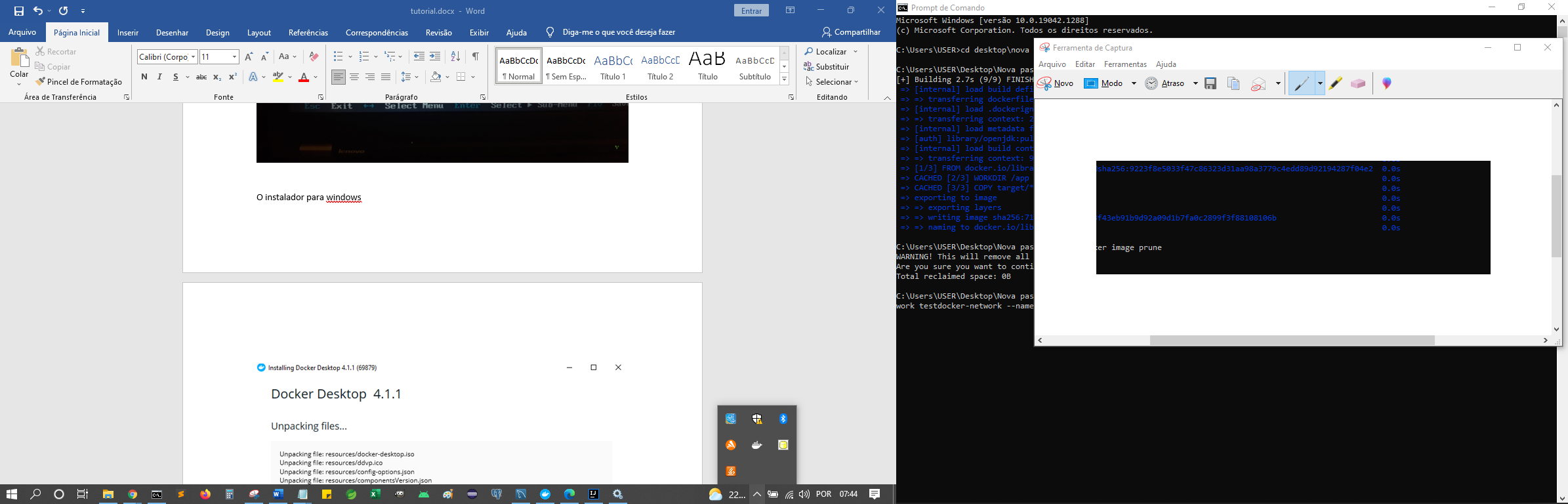
**O primeiro passo e descarregar Docker para Windows ele foi desenhado inicialmente para ambiente Linux mas com o tempo saíram versões para mac e Windows 10(se não for 10 não vai funcionar**

****

**Acessando na bios ao iniciar o sistema e apertando f4 ativamos a virtualização(o q e) necessária para poder trabalhar com Docker.**

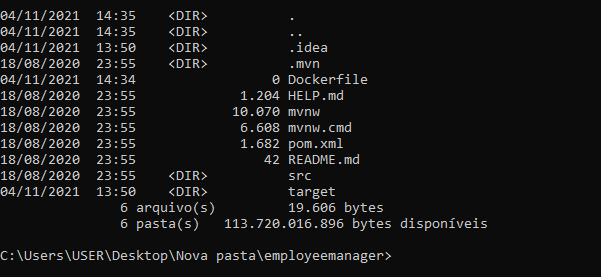
****

**O instalador para Windows faz toda a configuração previa**

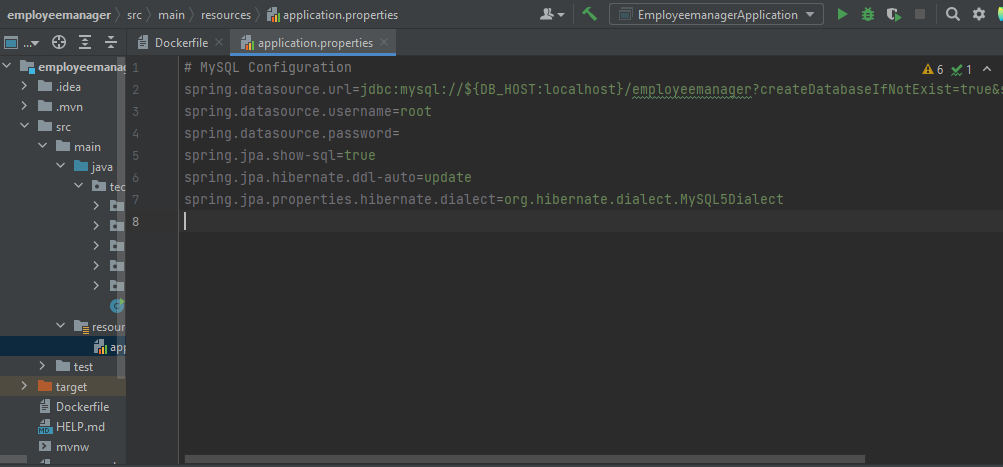
** **

**Para este exemplo e importante ter configurado as variáveis de ambientes do Maven já que precisa compilar a API**

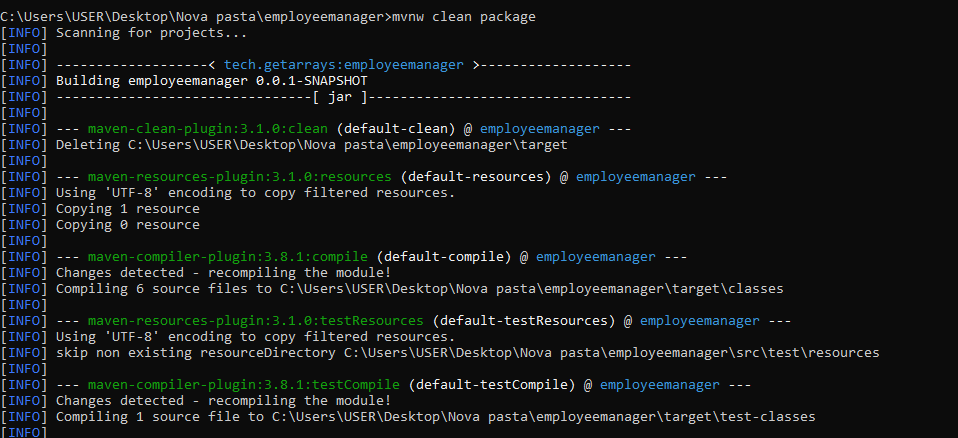
**Acedemos na pasta do arquivo**

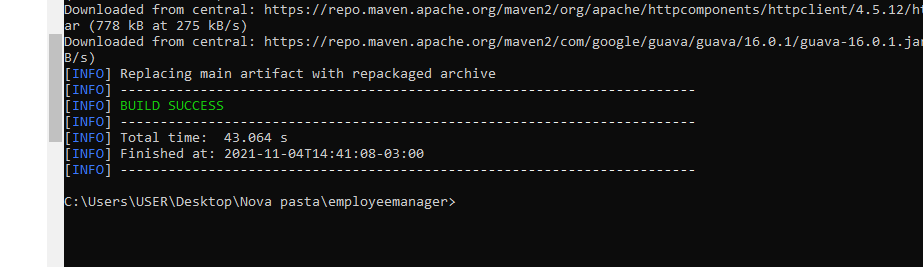
****

**E no arquivo application.properties dentro do Spring configuramos a base de dados com uma variável para que possa ser interpretado pelo Docker**

****

**Compilamos o arquivo posteriormente ter feito a configuração**

****

****

**O arquivo compilado se encontra na pasta target <nome\_do\_arquivo>.jar**

**E geramos um arquivo dentro da API Spring com o nome Dockerfile que contém as instruções para o Docker**

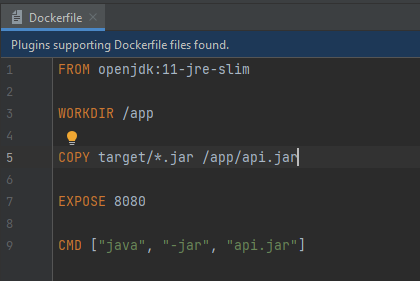
**FROM: o arquivo necessário a descarregar de dockerhub**

**WORKDIR: o nome da pasta do arquivo compilado no Docker**

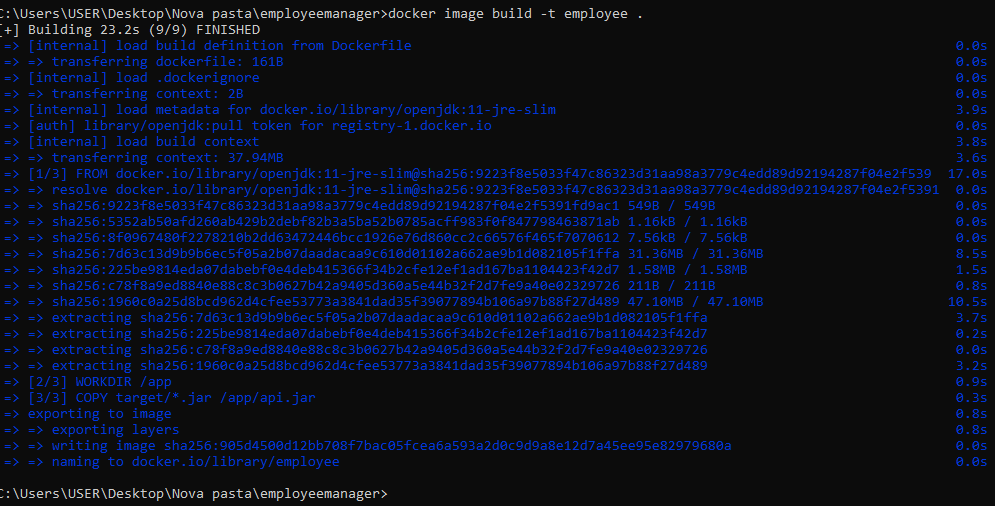
**COPY: faz uma cópia do arquivo compilado original (indicado pelo \* para ler qualquer nome )na pasta target feito previamente, na pasta app criada com o nome api.jar**

**EXPOSE: indica o porto que vai utilizar no Docker**

**CMD: mediante um array indica o nome para executar a cópia compilada pelo Docker no caso api.jar**

****

**No cmd criamos a imagem da API Spring**

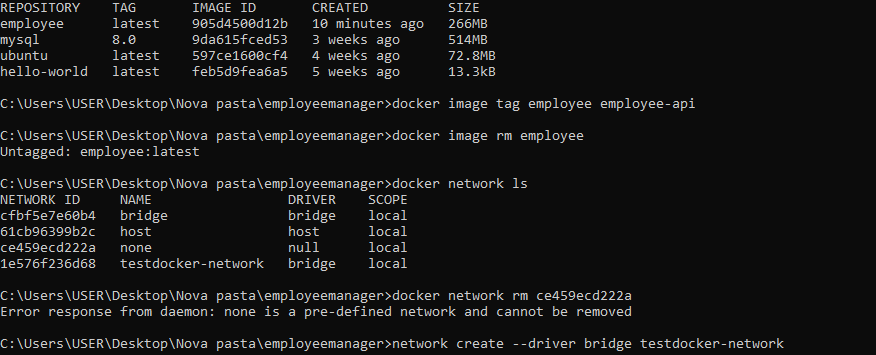
****

**Docker image tag: renomea uma imagem criada com um outro nome**

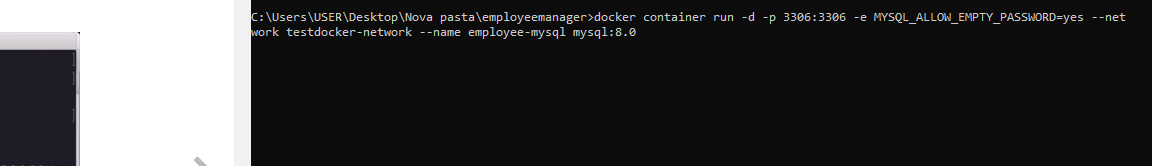
**Docker image rm: remove uma imagem selecionada**

**Docker image ls: lista as imagens no Docker**

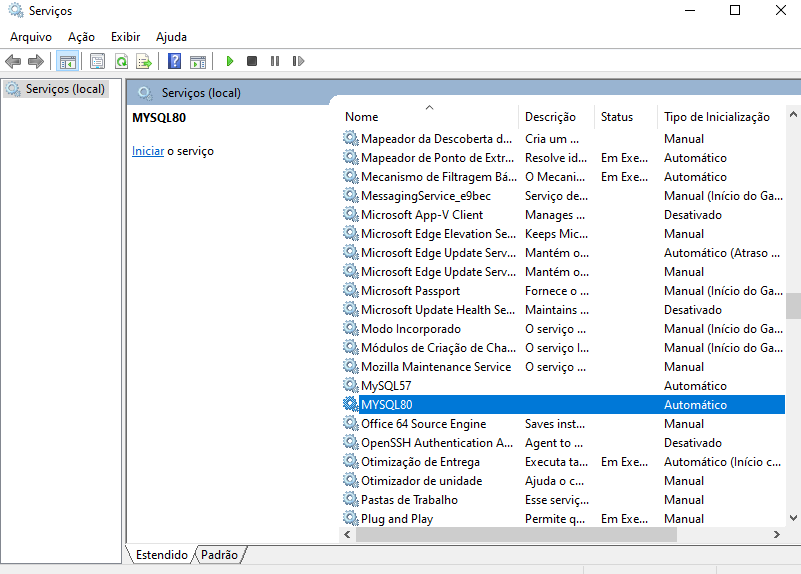
**Docker network ls: lista a redes default e criadas no Docker**

****

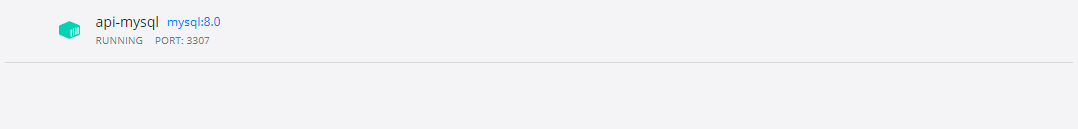
**Vamos a criar uma rede contendo MySQL, temos que indicar o porto onde se va executar, se permite senha vazia ou não, o nome da rede e o nome da imagem da base de dados**

****

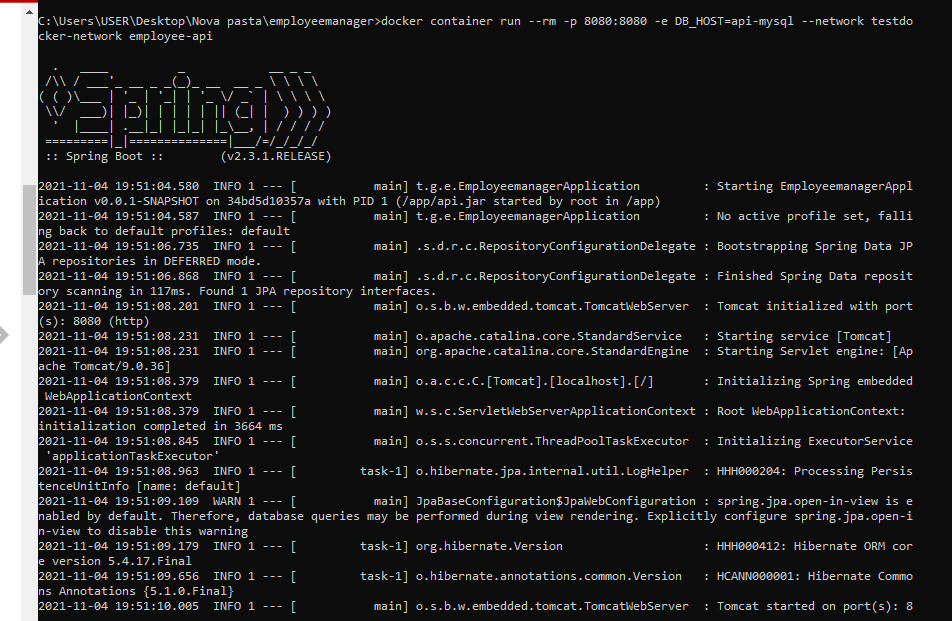
**Antes de iniciar o serviço Docker se deve deter o porto no sistema com o comando cmd *killstat* o em services -> mysql 80 ou 57**

****

**Dentro do Docker ativamos a porto onde vai se executar a API**

****

**E executamos no cmd a API com o seguinte comando**

****