

Alexandre Nuernberg

Esse documento atende a parte da questão 2, que pede a criação de uma imagem Docker e a disponibilização do Dockerfile.

Dados disponíveis no GitHub:

https://github.com/alexandreberg/Prova_SENAI_02302_2024_Questao2

Na pasta: host_server

Questão 2: **(COM GIT)**

O desenvolvimento de sistemas robóticos vem se tornando cada vez mais abordado em soluções industriais, comerciais e de serviços. Essa busca é acompanhada pela complexidade dos processos e problemas encontrados.

No âmbito de sistemas embarcados, a aplicação de ROS (Robot Operating System), agrega funcionalidades, bibliotecas e ferramentas para construção e programação de sistemas robóticos [2]. Para auxiliar no desenvolvimento de soluções com o ROS, muitos desenvolvedores de software embarcado aproveitam a versatilidade da plataforma Docker, projetada para facilitar compilações, execuções e compartilhamento de aplicações [3].

Diante do que foi exposto anteriormente, crie uma imagem Docker para a utilização dos pacotes ROS2 versão Humble. A imagem deve incluir as dependências necessárias para construir pacotes ROS, suas bibliotecas de comunicação, pacotes de mensagens, e ferramentas de linha de comando. O arquivo **Dockerfile deve ser entregue juntamente com instruções claras para construir a imagem Docker e executar containers a partir dela.**

Passo-a-passo para instalar o Docker no Ubuntu Linux

Rodar esse passo na máquina host que hospedará os containers Docker

Antes de rodar a imagem do ros2, é necessário que o Docker esteja instalado na máquina host. Seguem os passos de instalação.

Obs: comandos abaixo devem ser executados num terminal com shell por exemplo (*bash*) pode ser usado o *xterm*.

Passo 1: Atualizar o sistema

```
sudo apt update
```

Este comando atualiza a lista de pacotes disponíveis nos repositórios do Ubuntu.

Passo 2: Instalar pacotes necessários

```
sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release
```

Este comando instala os pacotes necessários para adicionar o repositório do Docker e baixar sua chave GPG.

Passo 3: Adicionar a chave GPG do Docker

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Este comando baixa a chave GPG do repositório do Docker e a adiciona ao seu sistema.

Passo 4: Adicionar o repositório do Docker

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \n $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Este comando adiciona o repositório oficial do Docker à lista de repositórios do seu sistema.

Passo 5: Atualizar a lista de pacotes

```
sudo apt update
```

Este comando atualiza a lista de pacotes para incluir os pacotes do repositório do Docker.

Passo 6: Instalar o Docker Engine

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Este comando instala o Docker Engine, a CLI do Docker, o containerd (runtime do container) e o plugin do Docker Compose.

Passo 7: Adicionar seu usuário ao grupo docker

```
sudo usermod -aG docker $USER
```

Este comando adiciona seu usuário ao grupo "docker", permitindo que você execute comandos do Docker sem sudo. **Você precisará fazer logout e login novamente para que essa alteração tenha efeito.**

Passo 8: Iniciar o Docker

```
sudo systemctl start docker
```

Este comando inicia o serviço do Docker.

Passo 9: Verificar a instalação

```
sudo docker run hello-world
```

Este comando baixa e executa uma imagem de teste do Docker, verificando se a instalação foi bem-sucedida.

O Docker agora está instalado no seu sistema Ubuntu. Você pode começar a usar o `docker` comando para baixar imagens, executar containers e gerenciar suas aplicações.

Passo-a-passo para criar a imagem Docker à partir de um Dockerfile

1. Salve o Dockerfile:

Crie um arquivo chamado *Dockerfile* com o conteúdo fornecido.

```
alerta@china:~$ cd
alerta@china:~$ mkdir docker
alerta@china:~$ cd docker
alerta@china:~/docker$ vi Dockerfile
```

Dockerfile com comentários:

Dockerfile

```
# Define a imagem base como Ubuntu 22.04
FROM ubuntu:22.04

# Configura o frontend do apt para não interativo (modo silencioso)
ENV DEBIAN_FRONTEND=noninteractive
# Define o fuso horário para São Paulo
ENV TZ=America/Sao_Paulo

# Instala pacotes necessários
RUN apt update && \
    apt install -y --no-install-recommends \
        software-properties-common \ # Permite adicionar repositórios PPA
        build-essential \           # Ferramentas essenciais para compilar software
        cmake \                     # Sistema de build para o ROS
        git \                       # Sistema de controle de versão
        python3-pip \               # Gerenciador de pacotes Python
        wget \                      # Utilitário para download de arquivos da web
        curl \                     # Utilitário para download de arquivos da web
        gnupg2 \                   # Ferramenta para verificação de assinaturas digitais
        lsb-release \              # Utilitário para obter informações sobre a
distribuição Linux
        locales \                  # Suporte a localização
        sudo \                    # Utilitário para executar comandos como superusuário
        tzdata && \                 # Dados de fuso horário
        rm -rf /var/lib/apt/lists/* # Limpa o cache do apt para reduzir o tamanho da imagem

# Configura o locale para inglês americano
RUN locale-gen en_US en_US.UTF-8 && \
    update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8

# Adiciona o repositório ROS 2
RUN curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg && \
    echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" | tee
/etc/apt/sources.list.d/ros2.list > /dev/null

# Instala o ROS 2 Humble versão Desktop
RUN apt update && \
    apt install -y --no-install-recommends ros-humble-desktop && \
    rm -rf /var/lib/apt/lists/*

# Configura o ambiente ROS 2
```

```
ENV ROS_DISTRO humble
RUN echo "source /opt/ros/$ROS_DISTRO/setup.bash" >> /root/.bashrc

# Cria um diretório para o workspace ROS 2
RUN mkdir -p /ws_ros2/src
WORKDIR /ws_ros2

# Comando para iniciar o container (bash)
CMD ["/bin/bash"]
```

O arquivo Dockerfile é necessário para a criação da imagem. Ele contém as instruções para o Docker sobre como construir a imagem, incluindo a instalação do sistema operacional, pacotes, dependências e configurações. Este Dockerfile instala o pacote ros-humble-desktop, que inclui ferramentas como RViz e Gazebo.

2. Construa a imagem:

Abra o terminal na pasta onde você salvou o Dockerfile e execute o seguinte comando:

```
docker build -t ros2-humble .
```

Este comando irá construir a imagem Docker com base nas instruções do Dockerfile.

- -t ros2-humble: Define o nome da imagem como "ros2-humble".
- .: Indica que o Dockerfile está no diretório atual.

3. Execute um container a partir da imagem:

Após a imagem ser criada, você pode executar um container com o seguinte comando:

```
docker run -it ros2-humble
```

Este comando irá iniciar um container interativo (-it) com base na imagem "ros2-humble".

Rodando o container

Uma vez o container criado no Linux-host siga os passos para executá-lo.

Verifique os containers que estão disponíveis e rodando

```
alerta@china:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
95974631fb73	ros2-humble	"/bin/bash"	10 hours ago	Up 2 seconds		ros2_02

```
alerta@china:~$
```

Caso não esteja, inicie o container do ros2:

```
alerta@china:~$ docker start ros2_02
```

Loga no container e criar um usuário *rosuser*, *ajusta a senha e adiciona ao grupo sudo*:

```
alerta@china:~$ docker exec -it ros2_02 bash
root@95974631fb73:/ws_ros2#
root@95974631fb73:/ws_ros2# useradd -m -d /home/rosuser -s /bin/bash rosuser
root@95974631fb73:/ws_ros2# passwd rosuser
root@95974631fb73:/ws_ros2# usermod -aG sudo rosuser
root@95974631fb73:/ws_ros2# su - rosuser
```

À partir daqui o sistema está pronto para responder as perguntas da questão 2.