

# **Relatório referente a implementação do Docker em Debian Linux**

## Sumário

<b>I – Requisitos do sistema (Debian):</b> .....	3
<b>II – Método de instalação do Docker:</b> .....	3
<b>III – Configuração inicial:</b> .....	4
<b>IV – Organização do Docker:</b> .....	4
<b>V – Hospedagem de sites através do Docker:</b> .....	5
<b>VI - Conclusão e sugestões:</b> .....	6

## I – Requisitos do sistema (Debian):

1. **Sistema operacional:** Debian 10 (Buster) ou superior;
2. **Processador:** Processador de 64 bits;
3. **Memória RAM:** 4gb de memória RAM;
4. **Espaço em disco:** 2GB de espaço livre HD/SSD.
5. **Conexão WI-FI:** necessária

## II – Método de instalação do Docker:

Existem diferentes maneiras de instalar o Docker no Debian, a adotada nesse método e a que se utiliza do CMD do próprio Debian (12) para instalar o Docker sem necessidade de fontes externas.

1. Inicie a máquina onde o Linux já se encontra instalado (Debian 10 ou mais recente);
2. Abra a central de aplicativos;
3. Abra o aplicativo denominado “Terminal”;
4. Dentro do Terminal digite “su” para entrar em modo root;
5. Digite sua senha de administrador;
6. No terminal digite “*sudo apt-get update*” para atualizar todos os softwares do Debian;
7. Depois digite “*sudo apt-get upgrade*” para ter certeza de que tudo foi atualizado;
8. Então digite “*sudo apt-get install \ apt-transport-https \ ca-certificates \ curl \ gnupg-agent \ software-properties-common*” para baixar softwares extras que serão necessários;
9. Vá até o site Docker Hub;
10. Na página responsável pelo Debian retire os códigos necessários;
11. Então digite “*for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove \$pkg; done*” para desinstalar pacotes que poderiam ser conflitantes com o docker;
12. Após isso, digite:  
“*sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /usr/share/keyrings/docker-archive-keyring.gpg*” isso irá instalar o repositório do Docker;
13. E logo após digite “*sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin*” para baixar a versão mais recente do Docker;
14. Após todos os passos de instalação serem concluídos ainda no terminal digite “*sudo docker run hello-world*” para verificar se o Docker foi corretamente instalado.

E importe que nenhum desses passos seja seguido caso o Docker já tenha sido instalado previamente na máquina, pois isso pode gerar um conflito no sistema do Docker. Caso alguma das etapas falhar consulte sua versão do Debian para saber se ela é compatível com tal funcionalidade, se a versão do Debian estiver correta consulte o site Docker Hub para ter mais especificações sobre a instalação.

### III – Configuração inicial:

- **Criação de imagens:**

Para criar as imagens no Docker apenas são necessários comandos pré-configurados pelo próprio Docker que esclareçam o caminho até a o diretório desejado, como por exemplo “`docker build -t nome-da-imagem`”;

- **Criação de contêineres:**

A criação de contêineres no docker exige uma complexidade maior, já que primeiramente será necessário criar um Dockerfile que se trata de um arquivo de texto redigido através de uma linguagem de programação que contém as instruções para criar uma imagem Docker. Após isso se usa o comando “*docker build*” para que o docker construa a imagem a partir da Dockerfile. Após todos esses passos iniciais já é possível criar e iniciar um contêiner Docker utilizando-se do comando “`docker run`” e especificando os parâmetros requeridos dentro do comando. Também é possível checar se o contêiner está rodando corretamente através do comando “*docker ps*”.

- **Interação com o contêiner:**

Para acessar o contêiner após sua criação use o código “*docker exec -it meu-container /bin/bash*”. Para parar seu contêiner aplique o código “*docker stop meu-container*” já para removê-lo digite “`docker rm meu-container`”

Todos os comandos citados acima funcionarão apenas se o Docker for configurado corretamente desde de o início. Além disso precisam ser minuciosamente redigidos para que a máquina entenda de maneira satisfatória.

### IV – Organização do Docker:

Para a organização e gerenciamento dos contêineres Docker existe uma infinidade de ferramentas com o Portainer, Docker swarm, Rancher entre diversas outras, mas a que utilizaremos é o Proxmox já que entre todas as ferramentas citadas ela é a que mais foi útil por sua simplicidade, número de ferramentas e transparência no momento de gerenciar os contêineres.

Uma alternativa para gerenciamento manual dos contêineres é a criação de códigos através de ferramentas como o GitHub actions que automatizem os trabalhos manuais utilizando-se de scripts e pipelines CI/CD para reduzir a necessidade de alguém manuseando manualmente os contêineres, porém ainda sendo necessário um supervisor para garantir a integridade dos códigos e de suas atividades.

Além disso é de grande importância a documentação minuciosa de cada passo tomado desde instalação a do Docker até as automações através do GitHub actions, já que inevitavelmente todos os contêineres precisaram de manutenção leve ou pesada para continuarem operando, e sem uma documentação extensa do processo de criação do container não é possível fazer uma manutenção efetiva.

## V – Hospedagem de sites através do Docker:

Configurando o servidor Web: A configuração dos servidores dos contêineres docker se inicia nesse caso através do Nginx, sendo necessário instala-lo através do seguintes passos:

1. Vá ate o site Nginx proxy manager e retire o código  
*“version: '3.8'”*  
*services:*  
*app:*  
*image: 'jc21/nginx-proxy-manager:latest'*  
*restart: unless-stopped*  
*ports:*
  - *'80:80'*
  - *'81:81'*
  - *'443:443'**volumes:*
  - *./data:/data*
  - *./letsencrypt:/etc/letsencrypt*”
2. Especifique a versão do seu docker compose na primeira linha do código;
3. Va até o site de download do VS Code e o instale no Linux;
4. Instale o VS Code em seu Linux;
5. No VS Code crie um arquivo chamado *“docker compose.yaml”* o nome do arquivo pode mudar dependendo da sua versão do docker compose;
6. Copie o código do Nginx no VS Code;
7. Abra o terminal do Linux (Debian) e coloque o nome do arquivo *“docker compose.yaml”* se digitado de maneira errônea o processo falhará;
8. Copie novamente o script do site Nginx no terminal que agora estará no modo docker;
9. Após copiar o comando aperte *“Ctrl + O”* para salvar, *“ENTER”* para confirmar e *“Ctrl + x”* para sair do modo docker;
10. Após isso digite *“docker compose up”* para iniciar o serviço do container.

Após todos esses comandos feitos o seu container já estará no ar apesar de ainda não terem sido definidos a imagem que será usada ou o as especificações do container docker.

## **VI - Conclusão e sugestões:**

Após esses passos serem feitos o seu container docker já estará no ar e funcionando podendo inclusive ser configurado e monitorado a partir do proxmox.

E recomendado que essas ações sejam feitas na ordem exata que foram descritas e não alteradas de nenhuma forma, para garantir o bom funcionamento e segurança do seus contêineres docker.