

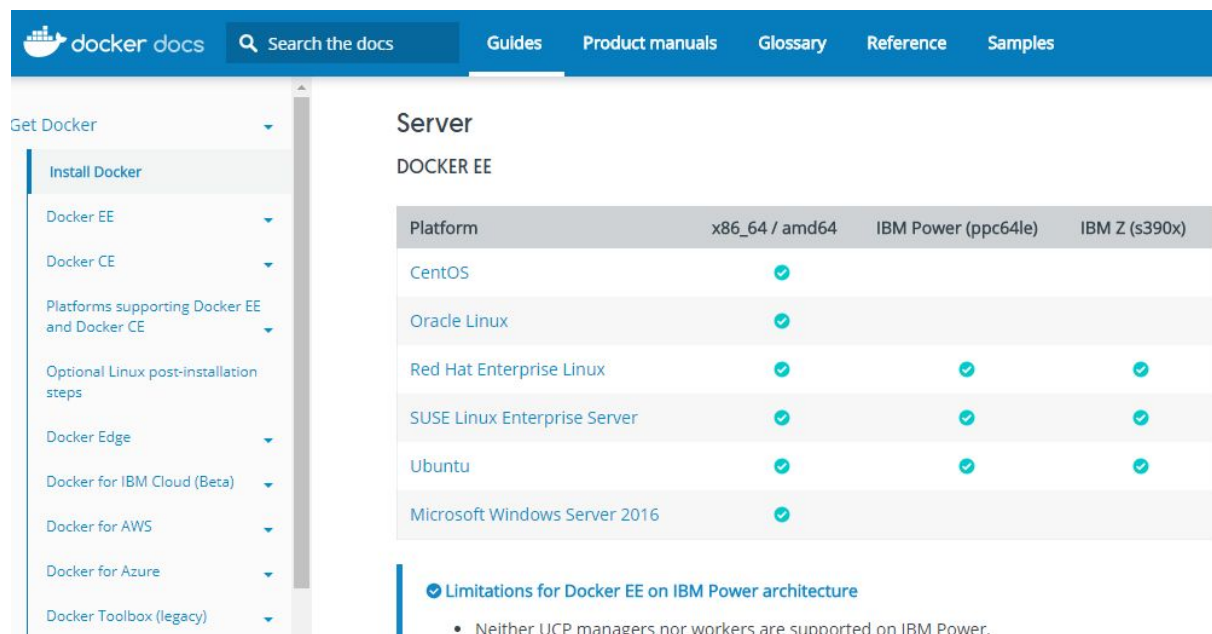
Pré instalações antes de instalar a DOJOT

Todo o processo de instalação pode ser encontrado no Git Hub

<https://github.com/dojot/docker-compose>

DOCKER COMPOSE

Como pré-requisito é feito a instalação do Docker, através do site, ele fornece um tutorial de instalação que é a mesma que será seguida aqui, utilizamos a distribuição ubuntu 16.04 para instalar a plataforma



The screenshot shows the Docker Docs website. The left sidebar has a 'Get Docker' menu with 'Install Docker' selected. The main content area is titled 'Server' and 'DOCKER EE'. It contains a table with columns for 'Platform', 'x86_64 / amd64', 'IBM Power (ppc64le)', and 'IBM Z (s390x)'. The table lists several operating systems with checkmarks indicating support. Below the table, there is a section titled 'Limitations for Docker EE on IBM Power architecture' with a bullet point stating that neither UCP managers nor workers are supported on IBM Power.

Platform	x86_64 / amd64	IBM Power (ppc64le)	IBM Z (s390x)
CentOS	✓		
Oracle Linux	✓		
Red Hat Enterprise Linux	✓	✓	✓
SUSE Linux Enterprise Server	✓	✓	✓
Ubuntu	✓	✓	✓
Microsoft Windows Server 2016	✓		

✓ Limitations for Docker EE on IBM Power architecture

- Neither UCP managers nor workers are supported on IBM Power.

<https://docs.docker.com/install/#server>

A instalação dos Componentes do Docker será feita utilizando o repositório do Git

Primeiro é feita a atualização dos pacotes do Ubuntu

- `sudo apt-get update`

Logo em seguida fazemos a instalação de pacotes para permitir o uso de um repositório através de HTTPS

- `sudo apt-get install \`
 `apt-transport-https \`
 `ca-certificates \`
 `curl \`
 `software-properties-common`

Adicionando uma chave GPG oficial do Docker, usadas para baixar as versões mais recentes:

- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

Após baixar a chave, faremos a verificação através do comando:

- `apt-key fingerprint 0EBFCD88`

Adicionando repositórios estáveis:

```
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

```
sudo add-apt-repository \
"deb [arch=armhf] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

```
sudo add-apt-repository \
"deb [arch=s390x] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

Instalando DOCKER CE

- `sudo apt-get update`

Instale a versão mais recente do Docker CE, ou vá para o próximo passo para instalar uma versão específica. Qualquer instalação existente do Docker é substituída.

- `sudo apt-get install docker-ce`

Após isso, vamos testar o docker para verificar se a instalação ocorreu de forma correta

- `sudo docker run hello-world`

Este comando baixará uma imagem de teste e a executa em um contêiner. Quando o contêiner é executado, ele imprime uma mensagem informativa e sai.

Após isso voltamos para o git e acessamos o Docker Compose

- https://github.com/docker/compose/blob/master/docs/install_guide.rst

e entramos na guia Docker Compose

<https://docs.docker.com/compose/install/>

No Linux, você pode baixar o binário do Docker Compose com a versão mais recente, Execute este comando para baixar a versão mais recente do Docker Compose :

```
curl -L
```

```
https://github.com/docker/compose/releases/download/1.19.0/docker-compose-`uname  
-s`-`uname -m` -o /usr/local/bin/docker-compose
```

Aplicando permissões para executáveis do binário

```
chmod +x /usr/local/bin/docker-compose
```

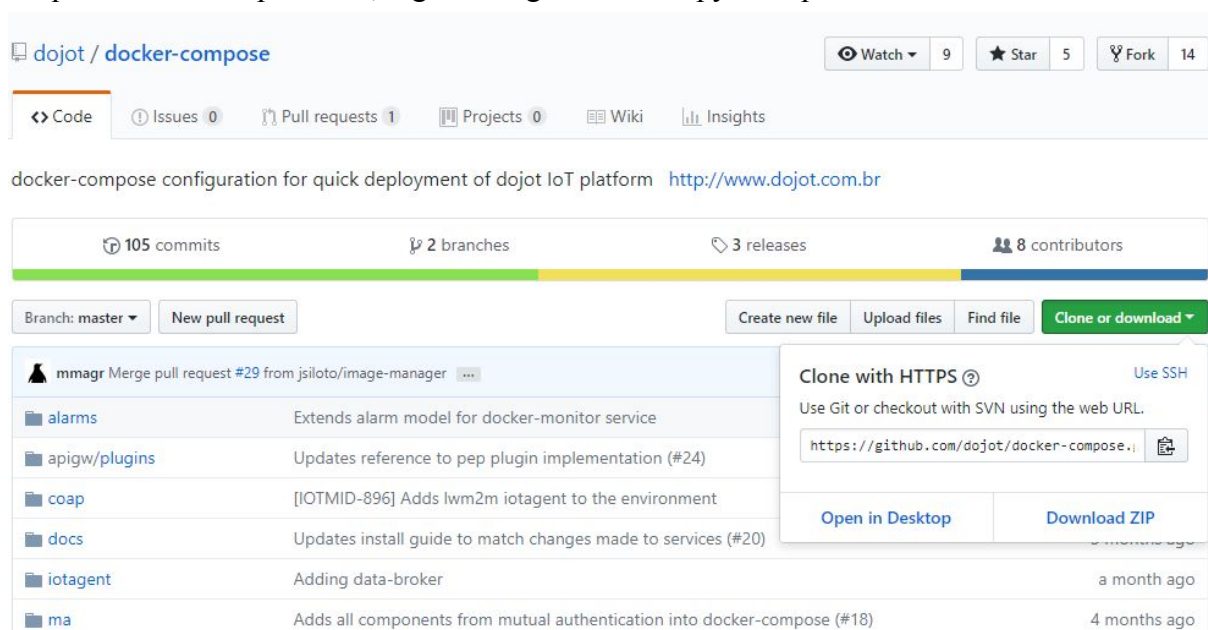
Teste de instalação , se a instalação ocorreu corretamente, deve ser exibido a ultima versão do Docker-Compose

- `docker-compose --version`
- `docker-compose version 1.19.0, build 1719ceb`

Instalação da Plataforma Dojot através do link

<https://github.com/dojot/docker-compose>

Clique em clonar repositório, logo em seguida em “copy to clipboard”



Para configurar o ambiente, simplesmente clone o repositório de implantação e execute os comandos abaixo. `git clone https://github.com/dojot/docker-compose.git`

Os scripts de implantação e o repositório de configuração habilitados pelo docker-compose podem ser encontrados em:

- `git clone git@github.com:dojot/docker-compose.git`

Após realizar o clone do repositório, entre em Docker-compose no meu caso está em `home/usuario/docker-compose`

- `cd docker-compose`

Depois que o repositório é clonado, ainda existem alguns módulos externos que devem ser baixados antes de usar a plataforma. Esses módulos podem ser baixados executando o seguinte comando:

```
git submodule update --init --recursive
```

Após isso atualizamos a lista de módulos baixados com o comando:

```
docker-compose up -d
```

Para listar os contêineres atualmente em execução, usamos o comando

```
docker ps
```

Para garantir o mapeamento adequado da API nos serviços de processamento, o gateway da API deve ser configurado. Para fazer isso, execute `kong_config.sh`, presente na raiz do repositório.

```
./kong.config.sh
```

A terminal window with a dark purple background. The prompt is root@UFCQXABL2LRCXXX:~/home/Luan/docker-compose#. The command ls is entered, showing a list of files: certa, coap, create.user.sh, cygnus, docker-compose.yml, docs, lotagent, kong.config.sh, mqtt, node-red-translator, policy.conf.sh, public, README.rst, test. The command ./kong.config.sh is then entered, and the prompt changes to root@UFCQXABL2LRCXXX:~/home/Luan/docker-compose# ./kong.config.sh. The rest of the terminal is obscured by a large black rectangle.

```
root@UFCQXABL2LRCXXX:~/home/Luan/docker-compose# ls
certa  coap  create.user.sh  cygnus  docker-compose.yml  docs  lotagent  kong.config.sh  mqtt  node-red-translator  policy.conf.sh  public  README.rst  test
root@UFCQXABL2LRCXXX:~/home/Luan/docker-compose# ./kong.config.sh
```

Por padrão a plataforma já cria um usuário `admin` para criar mais usuarios, execute o `create.user.sh` alterando o nome e a senha como também os privilégios

```
GNU nano 2.5.3 Arquivo: create.user.sh
#!/bin/bash -e

getip() {
  name=$(docker ps -f name-$1 --format "{{.Names}}")
  if [[ ! -z "$name" ]]; then
    docker inspect --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" $name
  fi
}

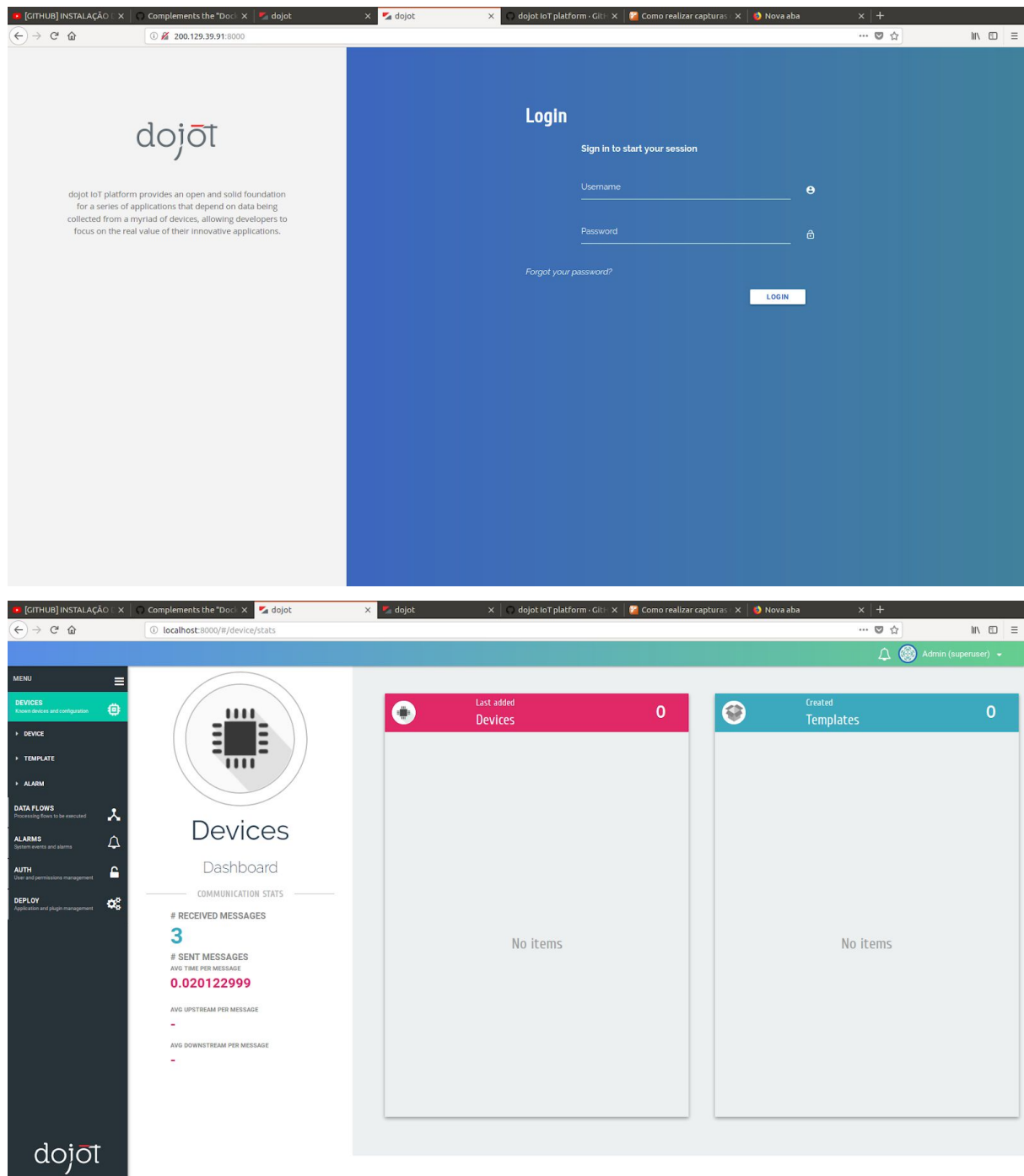
curl -sS $(getip auth):5000/user \
  -H 'content-type: application/json' \
  -d '{"username": "admin", "passwd": "admn", "service": "admn", "email": "admn@noemail.com", "name": "Admin (superuser)", "profile": "admn"}' |
  python -n json.tool
```

13 linhas lidas

Obter Ajuda Gravar Onde está? Recort txt Justificar Pos atual Pág anter Prim linha Onde tá próx. Marcar txt Recuar texto Desfazer
Sair Ler o arq Substituir Colar txt Lintar Ir p/ linha Próx pag Últ linha Parênteses Copiar texto Desalinhar txt Refazer

```
root@UFCQXABL2LRCXX:/home/luan/docker-compose# ls
certs coap create.user.sh cygnus docker-compose.yml docs lotagent kong.config.sh mqtt node-red-translator policy.conf.sh public README.rst test
root@UFCQXABL2LRCXX:/home/luan/docker-compose# nano create.user.sh
```

Se tudo ocorreu de forma certa, você pode acessar a plataforma através do endereço localhost:8000 ou utilizando o endereço ip caso o mesmo seja válido como no meu caso



Outras dependências é a biblioteca MQTT que pode ser facilmente instalado através do comando:

```
npm install mqtt --save
```

