**Esse documento foi Copiado do projeto do Saymon:** <https://github.com/saymowan/docker-mariadb-seleniumgrid-IC-POM>

**Algumas das configurações do arquivo não estão no meu projeto por meu projeto ser na linguagem “Java”.**

# Configurações para o container docker do Mantis, Selenium Grid e MariaDB.

**Criar o arquivo docker-compose.yml no diretório C:\mantis:**

db:

environment:

- MYSQL\_ROOT\_PASSWORD=root

- MYSQL\_DATABASE=bugtracker

- MYSQL\_USER=mantisbt

- MYSQL\_PASSWORD=mantisbt

image: mariadb

ports:

- "3306:3306"

restart: always

mantisbt:

image: "vimagick/mantisbt:latest"

links:

- db

ports:

- "8989:80"

restart: always

selenium-hub:

image: selenium/hub

container\_name: selenium-hub

ports:

- "4444:4444"

environment:

- GRID\_TIMEOUT=15

- GRID\_BROWSER\_TIMEOUT=15

- GRID\_MAX\_SESSION=2

chrome:

image: selenium/node-chrome

links:

- selenium-hub

environment:

- HUB\_HOST=selenium-hub

- HUB\_PORT=4444

- SCREEN\_WIDTH=1920

- SCREEN\_HEIGHT=1080

- SCREEN\_DEPTH=24

firefox:

image: selenium/node-firefox

links:

- selenium-hub

environment:

- HUB\_HOST=selenium-hub

- HUB\_PORT=4444

- SCREEN\_WIDTH=1920

- SCREEN\_HEIGHT=1080

- SCREEN\_DEPTH=24 - "8989:80"

**1. Preparação do ambiente Mantis**

Serão necessárias as seguinte configurações para iniciar o projeto:

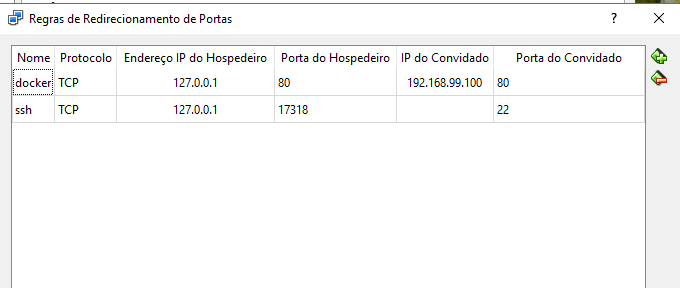
**Docker-compose:** no projeto, é possível encontrar um arquivo chamado "docker-compose.yml", este arquivo tem as devidas configurações da aplicação que foi automatizada, seu banco de dados e o framework de execução remota de testes automatizados. Cole o mesmo no diretório C:\mantis:

**1.1 Preparação Mantis + MariaDB**

1. Instalar [Docker Toolbox](https://docs.docker.com/toolbox/toolbox_install_windows/)
2. Executar o software *Docker Quickstart Terminal*
3. Acessar o diretório "C:/mantis" através do terminal aberto no passo anterior
4. No diretório haverá o arquivo **docker-compose.yml**
5. Executar o comando> docker-compose.exe up -d
6. Após o processamento, para validar execute o comando docker ps -a e os contêineres estarão disponíveis.

**1.2 Configuração máquina remota**

1. Com o tópico 1.1 já realizado, executar no *Docker Quickstart Terminal*, o comando docker-machine ip e coletar a informação
2. Abrir o software VirtualBox (última versão deverá estar instalada)
3. Encontrar a imagem referente ao docker
4. Acessar "Configurações"
5. Acessar "Redes"
6. Acessar "Avançado"
7. Acessar "Redirecionamento de Portas"
8. A configuração para funcionar no docker toolbox deverá estar dessa maneira:

[](https://camo.githubusercontent.com/e5e91a786f46f4df59f1f668215d8a7764b5500e/68747470733a2f2f692e696d6775722e636f6d2f6e554b547372322e706e67)

1. Incluir linha conforme nome "docker"
2. Protocolo: TCP
3. Endereço de Hospedeiro: 127.0.0.1
4. Porta de Hospedeiro: 80
5. IP Convidado preenchido com o valor recebido do docker (docker-machine ip default): 192.168.99.100
6. Porta do convidado: 80

O ambiente Mantis deverá estar disponível em: [http://192.168.99.100:8989](http://192.168.99.100:8989/)

**1.3 Configuração inicial Mantis**

Para o primeiro acesso ao Mantis, será necessário configurar o banco de dados conforme tabela abaixo:

| Variável | Valor |

|-----|------|

| Type of Database | MySQL Improved |

| Hostname (for Database Server) | mantis\_db\_1 |

| Username (for Database) | mantisbt |

| Password (for Database) | mantisbt |

| Database name (for Database) | bugtracker |

| Admin Username (to create Database if required) | root |

| Admin Password (to create Database if required) | root |

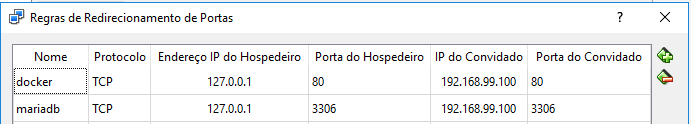
Após preencher, clicar em **Login/Continue** e aguardar o processamento (demora cerca de 5 minutos).

O primeiro acesso deverá ser feito utilizando as credenciais *administrator/root*. Redefinir a senha para o valor *administrator*.

**1.4 Configuração inicial MariaDB**

Abrir o software VirtualBox:

1. Encontrar a imagem referente ao docker
2. Acessar "Configurações"
3. Acessar "Redes"
4. Acessar "Avançado"
5. Acessar "Redirecionamento de Portas"
6. A configuração para funcionar no docker toolbox deverá estar dessa maneira:

[](https://camo.githubusercontent.com/a9ad4ff585d4026143e5b64653b08f3d0a4bd829/68747470733a2f2f692e696d6775722e636f6d2f30396c4a787a562e706e67)

1. Incluir linha conforme nome "docker"
2. Protocolo: TCP
3. Endereço de Hospedeiro: 127.0.0.1
4. Porta de Hospedeiro: 3306
5. IP Convidado preenchido com o valor recebido do docker (docker-machine ip default): 192.168.99.100
6. Porta do convidado: 3306

Para acessar o banco de dados, utilize algum SGBD MySQL (exemplo: Heidi SQL) com as credenciais listadas acima.

## 4. Selenium Grid

Para a execução remota dos testes automatizados, via selenum grid, serão utilizados os seguintes passos:

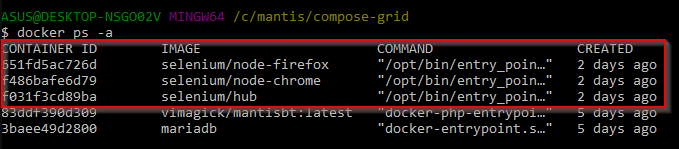
* Configuração dos contêineres hub, node chrome e node mozilla
* Verificação do console
* Configurações do projeto para execução: Local ou Remota
* Configurações do projeto para execução em um nó

**4.1 Configuração dos contêineres hub, node chrome e node mozilla**

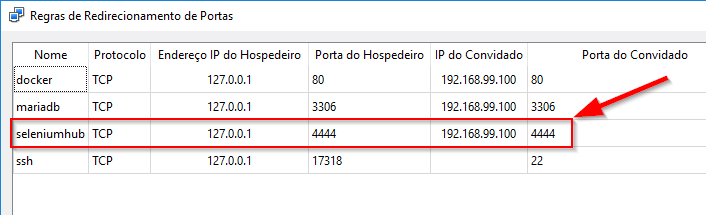
* Executar o software Docker Quickstart Terminal
* Após o processamento, para validar execute o comando docker ps -a e os contêineres estarão disponíveis:
* selenium/node-firefox
* selenium/node-chrome
* selenium/hub

**4.2 Verificação do console**

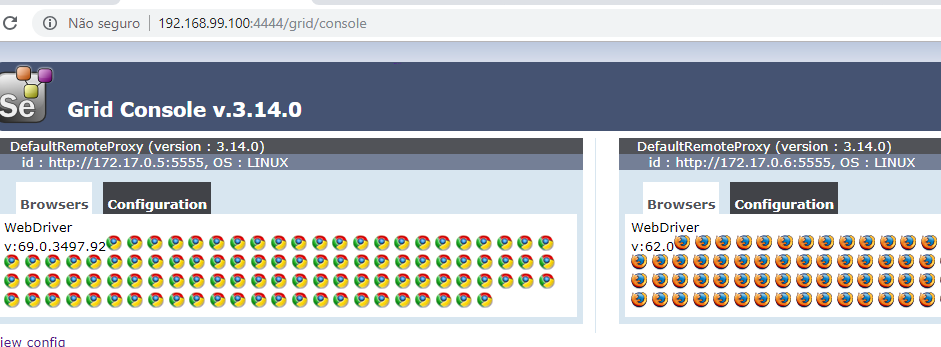
Após o processamento, as seguintes imagens estarão disponíveis em execução:

[](https://camo.githubusercontent.com/a108ec6a9cf8b36bcb3fef97d9035b6cacdf5557/68747470733a2f2f692e696d6775722e636f6d2f337669417642472e706e67)

Faça a configuração de porta no VirtualBox conforme passos anteriores informados:

[](https://camo.githubusercontent.com/5e6e2cfe541578e55e9a85aab939be1b47122bbd/68747470733a2f2f692e696d6775722e636f6d2f384d34484639382e706e67)

Ao executar o comando no navegador http://192.168.99.100:4444/grid/console também é possível verificar o console rodando corretamente com seus nós:

[](https://camo.githubusercontent.com/baf686fd18b39344ea66512addd2d005f3412e33/68747470733a2f2f692e696d6775722e636f6d2f7869447164356b2e706e67)

Quanto browser :P

**4.3 Configurações do projeto para execução: Local ou Remota**

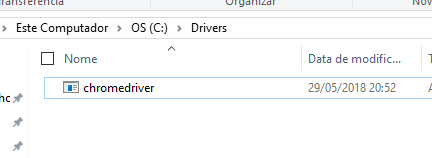
Para a execução local ou via docker, foi criada uma variável no App.Config chamada de <add key="Local" value="false"/>. Muito intuitiva.

* Caso seja remota, colocar o valor true e configurar corretamente o

selenium hub e nós.

* Caso seja local, iremos sempre utilizar o Google Chrome, para isso coloque o valor false e coloque um binário do chrome driver no

diretório C:\Drivers com o nome "chromedriver". Exemplo:

[](https://camo.githubusercontent.com/e262f30d543c29c12d9330346e91132830071b4c/68747470733a2f2f692e696d6775722e636f6d2f784347364642312e706e67)

**4.4 Configurações do projeto para execução em um navegador remoto**

Para a execução remota em um determinado nó (chrome ou mozilla), configure os seguintes parâmetros no App.config, o hub deverá estar rodando com os nós.

* Variável: <add key="Local" value="true"/>
* Variável que indica o navegador (chrome ou firefox): <add key="NavegadorDefault" value="chrome"/>

As configurações de tamanho, dimensionamento e outros argumentos por browser estão na classe Browser.cs.