

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:

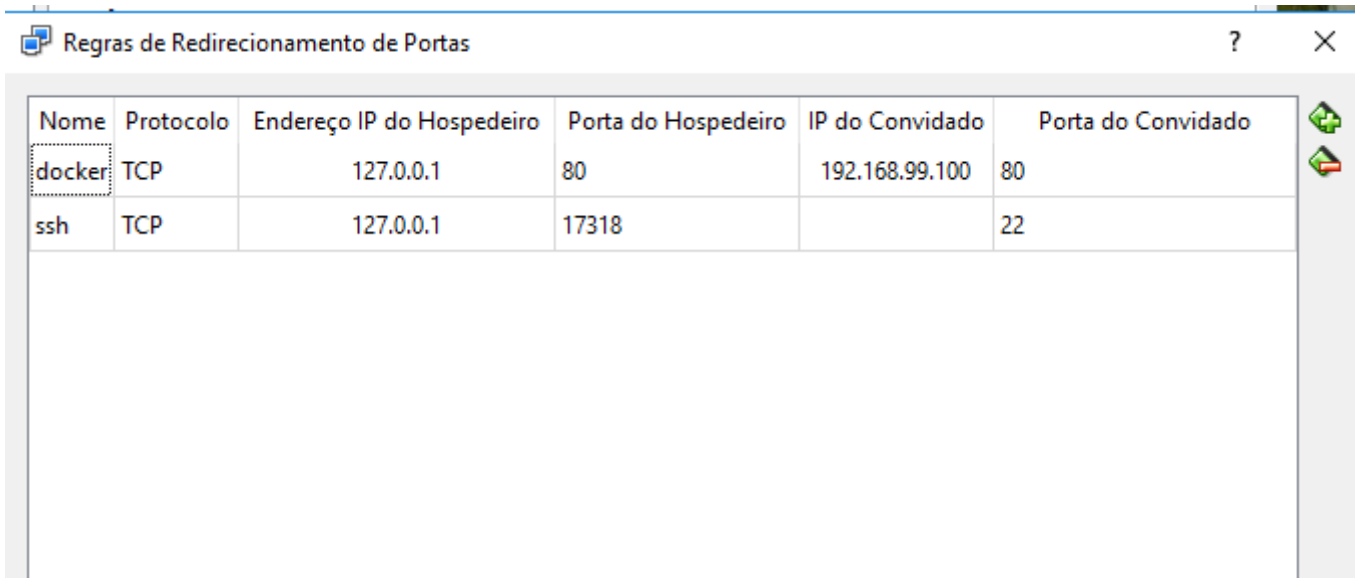
1.1 Preparação Mantis + MariaDB

1. Instalar [Docker Toolbox](#)
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:



Nome	Protocolo	Endereço IP do Hospedeiro	Porta do Hospedeiro	IP do Convidado	Porta do Convidado
docker	TCP	127.0.0.1	80	192.168.99.100	80
ssh	TCP	127.0.0.1	17318		22

9. Incluir linha conforme nome "docker"
10. Protocolo: TCP
11. Endereço de Hospedeiro: 127.0.0.1
12. Porta de Hospedeiro: 80
13. IP Convidado preenchido com o valor recebido do docker (docker-machine ip default): 192.168.99.100
14. Porta do convidado: 80

O ambiente Mantis deverá estar disponível em: <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 |

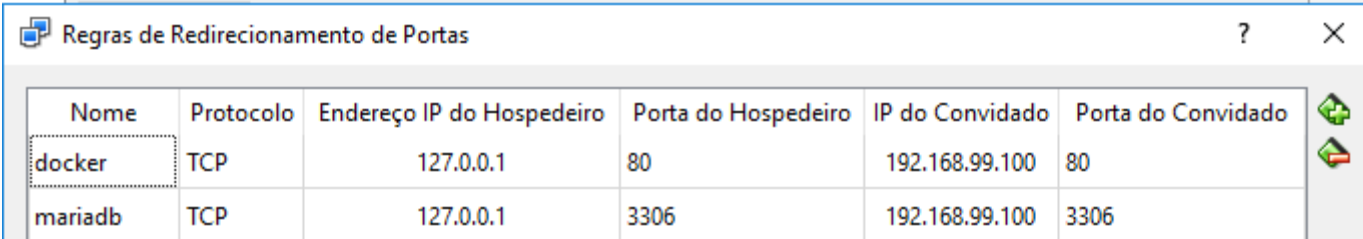
Após preencher, clicar em **Login/Continue** e aguardar o processamento (demora cerca de 5 minutos).

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:



Nome	Protocolo	Endereço IP do Hospedeiro	Porta do Hospedeiro	IP do Convidado	Porta do Convidado
docker	TCP	127.0.0.1	80	192.168.99.100	80
mariadb	TCP	127.0.0.1	3306	192.168.99.100	3306

7. Incluir linha conforme nome "docker"

8. Protocolo: TCP
9. Endereço de Hospedeiro: 127.0.0.1
10. Porta de Hospedeiro: 3306
11. IP Convidado preenchido com o valor recebido do docker (docker-machine ip default): 192.168.99.100
12. 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 selenium 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:

```

ASUS@DESKTOP-NSG002V MINGW64 /c/mantis/compose-grid
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
651fd5ac726d       selenium/node-firefox "/opt/bin/entry_poi..." 2 days ago
f486baf6d79        selenium/node-chrome "/opt/bin/entry_poi..." 2 days ago
f031f3cd89ba       selenium/hub        "/opt/bin/entry_poi..." 2 days ago
83dd+390d309       vimagick/mantisbt:latest "docker-php-entrypoi..." 5 days ago
3baee49d2800       mariadb             "docker-entrypoint.s..." 5 days ago

```

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

Nome	Protocolo	Endereço IP do Hospedeiro	Porta do Hospedeiro	IP do Convidado	Porta do Convidado
docker	TCP	127.0.0.1	80	192.168.99.100	80
mariadb	TCP	127.0.0.1	3306	192.168.99.100	3306
seleniumhub	TCP	127.0.0.1	4444	192.168.99.100	4444
ssh	TCP	127.0.0.1	17318		22

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:

Não seguro | 192.168.99.100:4444/grid/console

Grid Console v.3.14.0

DefaultRemoteProxy (version : 3.14.0)
 id : http://172.17.0.5:5555, OS : LINUX

Browsers

Configuration

WebDriver
 v:69.0.3497.92

iew confia

Quanto browser :P

4.3 Configurações do projeto para execução: Local ou Remota

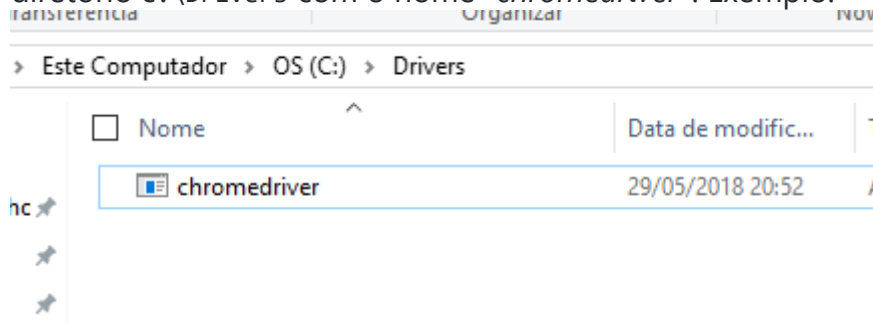
Para a execução local ou via docker, foi criada uma variável no App.Config chamada de `<add key="Local" value="false"/>`. Muito intuitiva.

- 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:



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`.