Selenium Grid Setup

Prerequisites

- Java 11 or higher installed
- Browser(s) installed

1) Standalone Mode

Standalone combines all Grid components seamlessly into one.
Running a Grid in Standalone mode gives you a fully functional Grid with a single command, within a single process. Standalone can only run on a single machine.

Step1: Download the Selenium Server jar file from the latest release

https://github.com/SeleniumHQ/selenium/releases

Step2:

Open terminal and navigate to the directory where you have placed the downloaded JAR file

Run the following command to start standalone server

```
java -jar selenium-server-<version>.jar standalone
```

```
java -jar selenium-server-4.11.0.jar standalone
```

After starting successfully the Grid in Standalone mode, point your WebDriver tests to *http://localhost:4444.*

2) Hub and Node (Using jar file)

A Hub is composed by the following components: Router, Distributor, Session Map, New Session Queue, and Event Bus.

Start Hub in the machine using

```
java -jar selenium-server-<version>.jar hub
```

The command below assumes the Node is running on the same machine where the Hub is running.

```
java -jar selenium-server-<version>.jar node
```

More than one Node on the same machine

```
java -jar selenium-server-<version>.jar node --port 5555
```

Node in different machine

To successfully register a Node to a Hub, it is important to expose the Event Bus ports (4442 and 4443 by default) on the Hub machine. This also applies for the Node port. With that, both Hub and Node will be able to communicate.

If the Hub is using the default ports, the --hub flag can be used to register the Node

```
java -jar selenium-server-<version>.jar node --hub http://<hub-ip>:4444
```

When the Hub is not using the default ports, the --publish-events and --subscribe-events flags are needed.

```
java -jar selenium-server-<version>.jar hub --publish-events tcp://<hub-ip>:8886 --subscribe-events tcp://<hub-ip>:8887 --port 8888
```

The Node needs to use those ports to register successfully

```
java -jar selenium-server-<version>.jar node --publish-events tcp://<hub-ip>:8886 --subscribe-events tcp://<hub-ip>:8887
```

3) Selenium Grid Using Docker

The Hub and Nodes will be created in the same network and they will recognize each other by their container name. A Docker network needs to be created as a first step.

Create docker network

```
docker network create grid
```

Create container for hub

```
docker run -d -p 4442-4444:4442-4444 --net grid --name selenium-hub selenium/hub:4.11.0-20230801
```

To Create Node

1.
```
docker run -d --net grid -e SE_EVENT_BUS_HOST=selenium-hub --shm-size="2g" -e SE_EVENT_BUS_PUBLISH_PORT=4442 -e SE_EVENT_BUS_SUBSCRIBE_PORT=4443 selenium/node-chrome:4.11.0-20230801
```

2.
```
docker run -d --net grid -e SE_EVENT_BUS_HOST=selenium-hub --shm-size="2g" -e SE_EVENT_BUS_PUBLISH_PORT=4442 -e SE_EVENT_BUS_SUBSCRIBE_PORT=4443 selenium/node-edge:4.11.0-20230801
```

Fore more session use "-e SE_NODE_MAX_SESSION=5"

```
docker run -d --net grid -e SE_EVENT_BUS_HOST=selenium-hub --shm-size="2g" -e SE_EVENT_BUS_PUBLISH_PORT=4442 -e SE_EVENT_BUS_SUBSCRIBE_PORT=4443 -e SE_NODE_MAX_SESSION=5 selenium/node-firefox:4.11.0-20230801
```