

# Selenium Grid для запуска в Jenkins

Отдел тестирования и внедрения

Exported on 02/21/2018

## Table of Contents

|  |    |
|--|----|
| 1 Как вы думаете, каким образом автотесты запускаются на вашем компьютере? .....   | 3  |
| 2 А как это происходит в Jenkins? .....  | 4  |
| 3 Так каким же образом сделать так, чтобы тесты запускались вдалеке от вас и не занимали рабочее время и пространство? ..... | 5  |
| 4 Шаг 0. Виртуальный сервер .....  | 6  |
| 5 Шаг 1. О файлах .....  | 7  |
| 6 Шаг 2. Подготовительные действия (да, в шаге 2) .....  | 8  |
| 7 Шаг 3. Запускаем! .....  | 10 |
| 8 Шаг 4. А как же тесты? .....   | 11 |
| 9 А если у вас не один сервер, а несколько, то скопируйте папку на другие и просто запустите ноды (startNode.bat) .....      | 12 |

## **1 Как вы думаете, каким образом автотесты запускаются на вашем компьютере?**

Без вашей помощи Cucumber обращается к установленному Ruby, который уже управляет браузером без вашей помощи, ориентируясь на ваш код.

## 2 *А как это происходит в Jenkins?*

Немного теории:

**Jenkins** - это система непрерывного развертывания, которая, в свою очередь, развернута на каком-то сервере и находится по определенному сетевому адресу.

**Selenium Grid** - тут теорией не ограничиться, скажу просто, что это технология, которая позволяет тестам удачно интегрироваться в Continuous Integration, основанную на Jenkins.

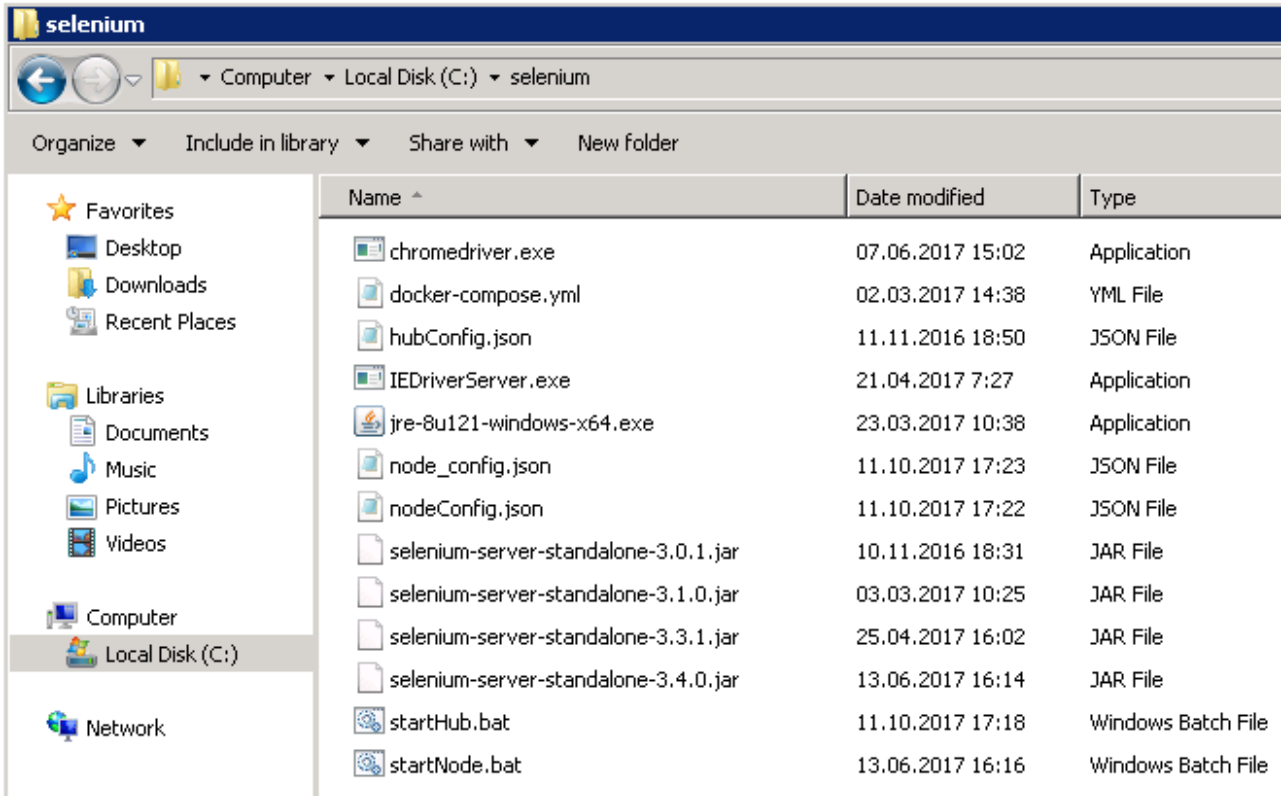
### ***3 Так каким же образом сделать так, чтобы тесты запускались вдалеке от вас и не занимали рабочее время и пространство?***

## 4 Шаг 0. Виртуальный сервер

Предположим, что он у вас есть. А если его у вас нет - нужно обращаться к тому, кто может вам его создать. Пусть для простоты понимания он будет **один !**

## 5 Шаг 1. О файлах

Создадим на виртуальном сервере папку. Ее название не важно, расположение, в целом, тоже. Но пока все делают папку **selenium** на диске **C**.



**В папке должны лежать:**

- драйвер для браузера, в котором должны запускаться автотесты - на картинке chromedriver.exe и iedriverserver.exe для Chrome и IE соответственно;
- hubConfig.json - настроечный файл для хаба - сервера - Selenium Grid;
- node\_config.json и nodeConfig.json - настроечные файлы для нод - узлов - Selenium Grid;
- selenium-server-standalone.jar - желательно посвежее) На картинке целых четыре варианта;
- startHub.bat и startNode.bat - файлы для запуска хаба и ноды;
- инсталлер для java - может не понадобится, но пусть будет.

## 6 Шаг 2. Подготовительные действия (да, в шаге 2)

Первым делом **установим Java**. Или узнаем, установлена ли она до нас.

Далее, посмотрим на файл **hubConfig.json**. Указанный ниже код не особо нуждается в пояснении, его мне не приходилось менять ни разу. **Запомните порт: 4444**.

### hubConfig.json

```
{ "host": null, "port": 4444, "newSessionWaitTimeout": -1, "servlets" : [], "prioritizer": null,
  "capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",
  "throwOnCapabilityNotPresent": true, "nodePolling": 5000, "cleanUpCycle": 5000, "browserTimeout": 0,
  "jettyMaxThreads": -1, "maxSession": 10 }
```

Теперь обратимся к файлу **node\_config.json**.

**Для каждого браузера свой блок**, в котором указаны название, максимальное количество сессий, протокол (не меняем) и путь к исполняемому файлу (именно в таком формате, потому что у нас все лежит в одной папке, начиная с шага 0).

А для всех - прокси и порт (не меняем), адрес хаба и его порт (запомнили же из hubConfig), подключено или нет (true) и максимальное количество сессий.

**Адресом хаба здесь является адрес виртуального сервера, на котором будет запущен bat-файл startHub** (вспомним, что изначально у нас один сервер, поэтому укажем тут его IP и порт 4444)

А теперь скопируем все, что есть в этом файле, в файл nodeConfig.json - они должны быть одинаковыми.

### node\_config.json

```
{
  "capabilities":
  [
    {
      "browserName": "chrome",
      "maxInstances": 10,
      "seleniumProtocol": "WebDriver",
      "webdriver.chrome.driver": "chromedriver.exe"
    },
    {
      "browserName": "internet explorer",
      "maxInstances": 10,
      "seleniumProtocol": "WebDriver",
      "webdriver.ie.driver": "IEDriverServer.exe"
    }
  ],
  "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
  "port": 5555,
  "hub": "http://autotest.ltcloud.alfaintra.net:4445",
  "register": true,
  "maxSession": 50
}
```



Посмотрим на **bat-файлы**. Стоит заметить, что **здесь мы ссылаемся на jar-файл selenium-server**, который изначально мы договорились взять посвежее - поэтому может понадобится заменить версию 😊

Из **startHub** будет запускаться сам сервер Selenium Grid

**startHub.bat**

```
REM %~dp0

java ^
-jar %~dp0\selenium-server-standalone-3.4.0.jar ^
-role hub ^
-hubConfig %~dp0\hubConfig.json
```

А там, где будет запущен **startNode**, будут запускаться тесты.

**startNode.bat**

```
java ^
-Dwebdriver.chrome.driver=chromedriver.exe ^
-Dwebdriver.ie.driver=IEDriverServer.exe ^
-jar selenium-server-standalone-3.4.0.jar ^
-role node ^
-nodeConfig node_config.json
```

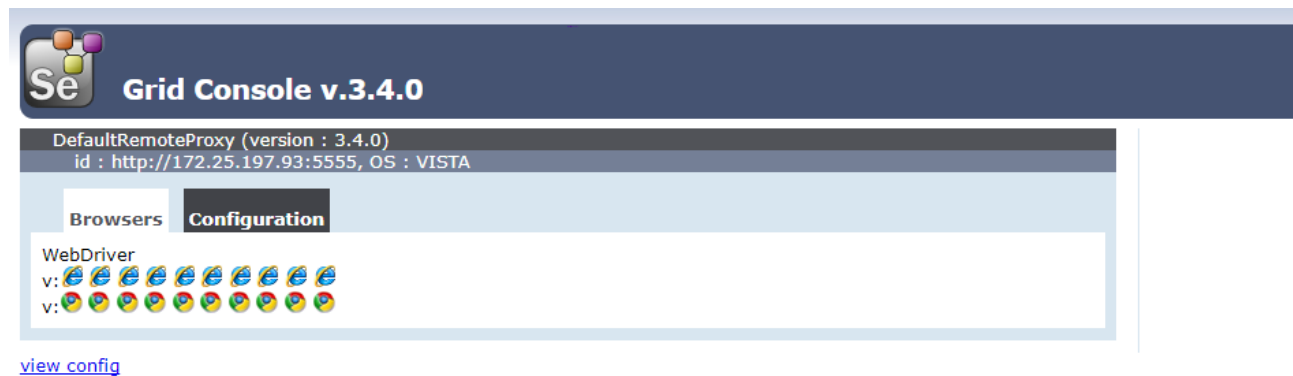
## 7 Шаг 3. Запускаем!

Для начала запустим **startHub.bat**

Потом запустим **startNode.bat**

Ведь сервер у нас только **один**.

Перейдем по ссылке вида **http://172.25.197.93:4444/grid/console**, где вместо моего IP будет IP вашего сервера. Это - консоль Selenium Grid. Здесь можно увидеть, какие узлы подключены к серверу, сколько их, сколько на них можно запустить браузеров и каких. Позже, когда запустите тесты, - можно посмотреть, на каком конкретно узле они запущены. Но сейчас тут пусто и одиноко:



## 8 Шаг 4. А как же тесты?

Если вы запускаетесь с помощью `rakefile`, то в файл `cucumber.yml` вашего проекта нужно добавить строку следующего вида: `remote: REMOTE_HUB=http://ufrvpndevjens1:4445/wd/hub`, где вместо моего хоста будет IP вашего сервера

А если вы запускаетесь с помощью слова `cucumber` или `parallel_cucumber`, то `REMOTE_HUB=http://ufrvpndevjens1:4445/wd/hub` нужно указать в строке запуска.

9 А если у вас не один сервер, а несколько, то скопируйте папку на другие и просто запустите ноды (startNode.bat)