# 概要

对于以下情况,可以构建Selenium Grid环境进行测试。

* 利用多个浏览器进行测试
* 利用多个Selenium Server执行测试,以便分散负荷

接下来，我们使用Docker ( docker-compose )简单地记载构筑Selenium Grid环境的步骤。

# Server配置

|  |
| --- |
| # cat /etc/centos-release  CentOS Linux release 7.5.1804 (Core)  # uname -r  3.10.0-862.14.4.el7.x86\_64  # docker -v  Docker version 18.09.0, build 4d60db4  # docker-compose -v  docker-compose version 1.23.1, build b02f130  # pip -V  pip 18.1 from /usr/lib/python2.7/site-packages/pip (python 2.7) |

# Proxy 配置

|  |  |
| --- | --- |
| yum | 如果使用代理服务器连接网络，需要进行以下设置。  # vi /etc/yum.conf  proxy=http://\*.\*.\*.\*:\* |
| 全体 | # vim /etc/profile  export http\_proxy=http://\*.\*.\*.\*:\*  # source /etc/profile |
| docker | Docker安装成功时  # mkdir /etc/systemd/system/docker.service.d  # touch /etc/systemd/system/docker.service.d/http-proxy.conf  # cat /etc/systemd/system/docker.service.d/http-proxy.conf  [Service]  Environment="HTTP\_PROXY=http://\*.\*.\*.\*:\*"  Environment="NO\_PROXY=localhost,127.0.0.0/8,docker-registry.somecorporation.com"  # systemctl daemon-reload # systemctl restart docker |

# 安装docker

<https://docs.docker.com/install/linux/docker-ce/centos/>

按照上述链接进行docker的安装。

命令一览：

|  |
| --- |
| $ sudo yum remove docker \  docker-client \  docker-client-latest \  docker-common \  docker-latest \  docker-latest-logrotate \  docker-logrotate \  docker-selinux \  docker-engine-selinux \  docker-engine |
| $ sudo yum install -y yum-utils \  device-mapper-persistent-data \  lvm2 |
| $ sudo yum-config-manager \  --add-repo \  https://download.docker.com/linux/centos/docker-ce.repo |
| $ sudo yum-config-manager --enable docker-ce-edge |
| $ sudo yum-config-manager --disable docker-ce-edge |
| $ sudo yum install docker-ce |
| $ sudo systemctl start docker |
| $ docker info |
| $ docker -v |

# 安装docker-compose

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

按照上述链接进行docker的安装。

命令一览：

|  |
| --- |
| $ pip install docker-compose -i http://pypi.douban.com/simple --trusted-host pypi.douban.com |
| $ docker-compose --version |

# docker-compose配置

<https://github.com/SeleniumHQ/docker-selenium/wiki/Getting-Started-with-Docker-Compose>

按照上述链接进行docker的安装。

命令一览：

|  |
| --- |
| # mkdir docker\_compose  # cd docker\_compose/  # touch docker-compose.yml  # cat docker-compose.yml  version: '2'  services:  firefox:  image: selenium/node-firefox:3.14.0-gallium  volumes:  - /dev/shm:/dev/shm  depends\_on:  - hub  environment:  HUB\_HOST: hub  chrome:  image: selenium/node-chrome:3.14.0-gallium  volumes:  - /dev/shm:/dev/shm  depends\_on:  - hub  environment:  HUB\_HOST: hub  hub:  image: selenium/hub:3.14.0-gallium  ports:  - "4444:4444" |

获取docker image

|  |
| --- |
| # docker pull selenium/node-firefox:3.14.0-gallium  # docker pull selenium/node-chrome:3.14.0-gallium  # docker pull selenium/hub:3.14.0-gallium |

启动docker-compose

|  |
| --- |
| # docker-compose up –d |

增加Grid

|  |
| --- |
| $ docker-compose scale chrome=5 |
| $ docker-compose scale firefox=5 |

|  |
| --- |
| # docker image ls  REPOSITORY TAG IMAGE ID CREATED SIZE  selenium/node-firefox 3.14.0-gallium e2942a950b47 5 weeks ago 769MB  selenium/node-chrome 3.14.0-gallium 62ed567565eb 5 weeks ago 907MB  selenium/hub 3.14.0-gallium 9e32a7c55661 5 weeks ago 321MB  # docker container ls  CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  ed388c4be309 selenium/node-firefox:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_firefox\_3\_c706de7f3dd2  3bef899390c1 selenium/node-firefox:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_firefox\_2\_3f86a65e75b0  2e9d1388fdf1 selenium/node-firefox:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_firefox\_5\_43f2cd4cae8a  35a89a2f5fb8 selenium/node-firefox:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_firefox\_4\_231c940df4a9  30c40bedbb1f selenium/node-chrome:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_chrome\_5\_46dfd11cd768  7f8637c70049 selenium/node-chrome:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_chrome\_2\_654efb336dad  d5ce84e10f3f selenium/node-chrome:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_chrome\_4\_45aa84e28401  140cb2f0dd3c selenium/node-chrome:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_chrome\_3\_8f6331a9b91d  66a48c20f466 selenium/node-firefox:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_firefox\_1\_6891cef58b85  e3363384f649 selenium/node-chrome:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours docker\_compose\_chrome\_1\_ae76fcd70395  c346bfe1d63c selenium/hub:3.14.0-gallium "/opt/bin/entry\_poin…" 3 hours ago Up 3 hours 0.0.0.0:4444->4444/tcp docker\_compose\_hub\_1\_dc70abbcc37f |

# Selenium Node on Windows (IE)

1. 首先，下载Selenium Server。

<https://selenium-release.storage.googleapis.com/3.14/selenium-server-standalone-3.14.0.jar>

2. 然后下载32bit 版的Internet Explorer Driver Server。

<http://www.seleniumhq.org/download/>

IE相关配置如下：

* 打开IE的保护模式

首先，打开IE ->设置->Internet选项->安全选项卡，勾上启用保护模式前的对勾即可。

然后，打开IE ->设置->Internet选项->高级选项卡，将扩张保护模式有效的对勾取消即可。

如果是首次在IE浏览器上测试，务必检查以下几个设置：

1. IE选项设置的安全页中，4个区域的启用保护模式的勾选都去掉（或都勾上）

2. IE页面的显示比例要为100%

3. 下载IEDriverServer.exe文件，放到环境变量path路径的目录里或把他所在的目录加到path环境变量中。

4. 如果IE浏览器设置安全性较高，在“Internet Options”中都不要选择“Enable Protected Mode”（保护模式），否则可能遇到错误提示。

* 注册表配置

32bit OS：

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE\_BFCACHE

64 bit OS：

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Internet   
Explorer\Main\FeatureControl\FEATURE\_BFCACHE

iexplorer.exe DWORD32 0

3. Selenium Node 配置

※　创建nodeconfig.json文件

|  |
| --- |
| {  "capabilities":  [  {  "browserName": "internet explorer",  "version": 11,  "platform": "WINDOWS",  "maxInstances": 1,  "seleniumProtocol": "WebDriver"  }  ],  "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",  "maxSession": 2,  "port": -1,  "register": true,  "registerCycle": 5000,  "hub": "http://\*.\*.\*.\*:4444",  "nodeStatusCheckTimeout": 5000,  "nodePolling": 5000,  "role": "node",  "unregisterIfStillDownAfter": 60000,  "downPollingLimit": 2,  "debug": false,  "servlets" : [],  "withoutServlets": [],  "custom": {}  } |

4. 启动Selenium Server。

|  |
| --- |
| > java -Dwebdriver.ie.driver=IEDriverServer.exe -jar selenium-server-standalone-3.14.0.jar -role node -nodeConfig nodeconfig.json |

# Selenium Node on Windows (Microsoft Edge)

1. 首先，下载Selenium Server。

<https://selenium-release.storage.googleapis.com/3.14/selenium-server-standalone-3.14.0.jar>

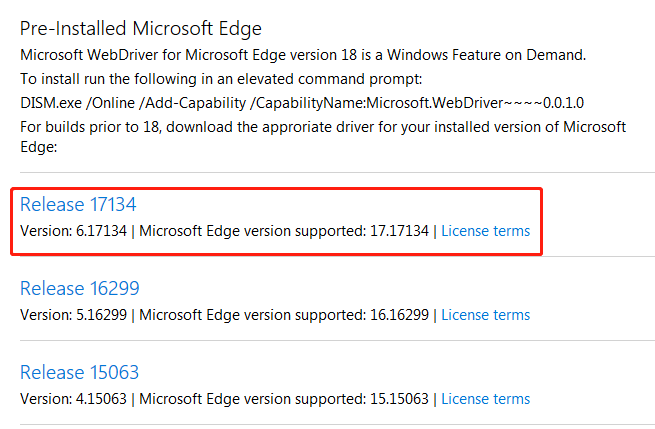
2. 然后下载Microsoft Web Driver。

<https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/>

* 需要根据Edge的版本下载对应的Web Driver.

例如: 42.17134.1.0 Microsoft EdgeHTML 17.17134

该版本对应的Web Driver如下:



注意: 如果Edge版本高于18, 不需要下载Web Driver. 只需要用管理员权限打开控制台,执行以下命令即可.

DISM.exe /Online /Add-Capability /CapabilityName:Microsoft.WebDriver~~~~0.0.1.0

也可以通过图形界面操作, 下载新特性.

Settings - apps - manage optional features - add a feature - Microsoft Webdriver

* 如果测试中Selenium Server的控制台出现以下错误，需要进行Edge的相关配置。

Microsoft Edge can’t be opened using Built-In administrator account.

Edge相关配置如下：

* 启动 gpedit.msc
* 依次打开以下配置项目

计算机配置 -> Windows 设置 -> 安全设置 -> 本地策略 -> 安全选项

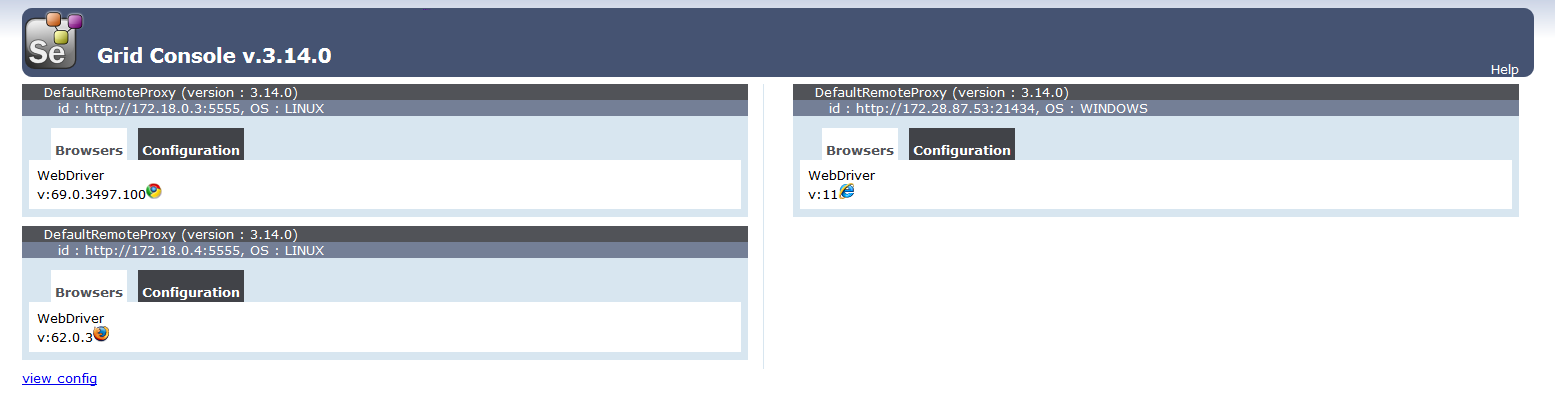
* 启用以下策略

用户账户控制：用于内置管理员账户的管理员批准模式

* 重新计算机

# Grid Console

[http://\*.\*.\*.\*:4444/grid/console](http://172.28.126.49:4444/grid/console)



# TestCase验证

|  |
| --- |
| import java.net.MalformedURLException;  import java.net.URL;  import org.junit.Test;  import org.openqa.selenium.Capabilities;  import org.openqa.selenium.remote.DesiredCapabilities;  import org.openqa.selenium.remote.RemoteWebDriver;  public class MyTest {  Capabilities chromeCapabilities = DesiredCapabilities.chrome();  Capabilities firefoxCapabilities = DesiredCapabilities.firefox();  Capabilities ieCapabilities = DesiredCapabilities.internetExplorer();  @Test  public void test() throws MalformedURLException {  RemoteWebDriver chrome = new RemoteWebDriver(new URL("http://\*.\*.\*.\*:4444/wd/hub"), chromeCapabilities);  RemoteWebDriver firefox = new RemoteWebDriver(new URL("http://\*.\*.\*.\*:4444/wd/hub"), firefoxCapabilities);  RemoteWebDriver ie = new RemoteWebDriver(new URL("http://\*.\*.\*.\*:4444/wd/hub"), ieCapabilities);  // run against chrome  chrome.get("https://www.baidu.com/");  System.out.println(chrome.getTitle());  // run against firefox  firefox.get("https://www.baidu.com/");  System.out.println(firefox.getTitle());    // run against ie  ie.get("https://www.baidu.com/");  System.out.println(ie.getTitle());  chrome.quit();  firefox.quit();  ie.quit();  }  } |