

Выбирается инструмент тестирования, который зависит от технологии, на которой построено веб-приложение.

Область автоматизации – это область тестируемого веб-приложения, которая будет автоматизирована специалистом.

Планирование, проектирование и разработка. На этом этапе создается план и стратегия написания фреймворка для тестирования.

Выполнение может быть выполнено с помощью средства автоматизации напрямую, или через средство управления тестированием, которое вызовет средство автоматизации.

Подход к обслуживанию автоматизации тестирования. На этом этапе тестирования автоматизации, проводится проверка, чтобы удостовериться в работе новых функций, которые были добавлены в веб-приложение.

Для разработки фреймворка автоматизированного тестирования сайта `abiturient.gsu.by` был выбран такой язык как JavaScript и TypeScript. Программная платформа Node.JS и инструмент тестирования Selenium WebDriver.

**А. С. Дробышевский**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **СОЗДАНИЕ И НАСТРОЙКА LINUX АГЕНТА ДЛЯ CI/CD СИСТЕМЫ TEAMCITY**

Агент сборки TeamCity – это часть программного обеспечения, которая прослушивает команды с сервера TeamCity и запускает фактические процессы сборки.

Для дипломного проекта было необходимо автоматизировать сборку и доставку Docker контейнеров. Данную функцию можно было реализовать на существующем сервере TeamCity, установленной на основе операционной системы Windows. Но нюансом послужил тот факт, что проектировался этот сервер точно под нужды сборки и доставки основного приложения, вследствие чего при добавлении дополнительного программного обеспечения ощущалась нехватка оперативной памяти. Была возможность увеличить размер основной виртуальной машины с 8 Гб оперативной память до 16 Гб, но это в свою очередь сильно увеличивало стоимость обслуживания сервера. Последующие расчеты показали, что при использовании отдельного выделенного

сервера на основе Linux появлялась возможность экономии в 8-ми кратном размере.

Настройка созданного сервера под управлением ОС Linux приведена ниже.

Первым шагом будет подготовка операционной системы к обновлениям и установке Java.

```
sudo apt update
```

```
sudo apt install default-jre -y
```

Далее необходимо указать глобальную переменную JAVA\_HOME.  
*JAVA\_HOME="/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java"*

Следующие скрипты предназначены для скачивания изначальных ресурсов и кода агента, с которым и будут работать в последствии.

```
sudo apt install unzip
```

```
mkdir teamcity-agent
```

```
wget https://teamcity.projectdomain.com/update/buildAgentFull.zip --no-check-certificate
```

```
mv buildAgentFull.zip teamcity-agent/
```

```
cd teamcity-agent/
```

```
unzip buildAgentFull.zip
```

```
rm -rf buildAgentFull.zip
```

После того, как были скачаны ресурсы на виртуальную машину, начинается шаг настройки агента. Для этого необходимо перейти в директорию conf и отредактировать параметр serverUrl со значения <http://teamcity.com:8111> на необходимое – домен основного TeamCity сервера, с учетом https соединения.

```
cd conf/
```

```
cp buildAgent.dist.properties buildAgent.properties
```

```
nano buildAgent.properties
```

```
#change serverurl to https://teamcity.projectdomain.com
```

Для обеспечения защищенного соединения необходимо установить сертификаты и связать основной TeamCity сервер и агент по https.

Создать в директории conf директорию ssl, в которую поместить сертификат ser. Помещение сертификата производится с помощью подключения к виртуальной машине через WinSCP и помещением сертификата в директорию ssl.

```
mkdir ssl
```

Далее необходимо добавить данный сертификат в доверенные с помощью утилиты keytool, поставляемой вместе с java. Запускать следует с помощью учетной записи root.

```
sudo su
```

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/keytool -importcert -file  
/home/azureuser/teamcity-agent/conf/ssl/procomplynet.cer -srcstoretype  
pkcs12 -keystore /usr/lib/jvm/java-1.11.0-openjdk-  
amd64/lib/security/cacerts
```

После данной настройки агент необходимо запустить следующей командой.

```
~/teamcity-agent/bin/agent.sh start
```

В последствии агент подключится к основному серверу и останется его авторизовать в веб интерфейсе TeamCity.

Далее надо установить необходимое программное обеспечение для сборки Docker контейнеров.

```
sudo apt-get update  
sudo apt-get install \  
ca-certificates \  
curl \  
gnupg \  
lsb-release
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --  
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

```
echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/key-  
rings/docker-archive-keyring.gpg] https://download.docker.com/linux/ub-  
untu \  
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >  
/dev/null
```

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Протестировать работоспособность Docker можно с помощью следующей команды.

```
sudo docker run hello-world
```

Следующей командой авторизоваться в Docker Registry

```
docker login projectregistry.azurecr.io  
#use credentials from azure registry
```

Последним шагом для реализации сборочной конфигурации является установка Azure cli, с помощью которой контейнеры будут доставляться на вычислительные мощности.

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Остается только залогиниться.

*az login*

Итогом данной работы является отдельно выделенный Linux агент, работающий по безопасному соединению, на который установлено необходимое программное обеспечение для сборки и доставки Docker контейнеров на вычислительные мощности.

**А. А. Зубрицкий**

(Институт физики НАН Беларуси, Минск)

Науч. рук. **М. С. Усачёнок**, канд. физ.-мат. наук

## **ПРИМЕНЕНИЕ СРЕДСТВ ЯЗЫКА PYTHON ДЛЯ ОБРАБОТКИ ДАННЫХ, ПОЛУЧЕННЫХ СВЧ РЕЗОНАТОРНЫМ МЕТОДОМ**

### **Введение**

СВЧ резонатор является одним из основных средств диагностики газоразрядной плазмы, служащий для определения плотности и частоты столкновений электронов. Плотность электронов определяется по смещению резонансной кривой резонатора, а частота столкновений по изменению его добротности. В нашей предыдущей работе [1] мы рассмотрели возможность создания автоматизированной системы регистрации плотности электронов на основе платы National Instruments PCIe-6351. Данная работа посвящена обработке получаемых данных с помощью средств среды программирования Python, как одного из наиболее бурно развивающихся языков программирования.

### **Экспериментальная установка**

Экспериментальная установка представляла собой 10-см СВЧ резонатор, портативный векторный анализатор спектра N9918A (Keysight) и ПК. Цилиндрический резонатор (диаметр – 9 см, высота – 3 см) и без неоднородностей имеет центральную резонансную частоту 2525,4 МГц и добротность 765. Исследуемая газоразрядная плазма создавалась в лампе ГШ-5 (газ неон при пониженном давлении, рисунок 1, а) при пропускании через нее постоянного тока порядка единиц миллиампер. Зарегистрированные резонансные контуры для нескольких значений разрядного тока представлены на рисунке 1, б.