

Настройка проекта VProfile

Требуемое ПО

1. Oracle VM Virtualbox
2. Vagrant
3. Vagrant plugins
 - a. `vagrant plugin install vagrant-hostmanager`
 - b. `vagrant plugin install vagrant-vbguest`
4. Git bash или эквивалент

Настройка виртуальных машин

1. Склонировать исходный код.
2. `cd` в папку репозитория.
3. `cd` в папку `vagrant/Manual_provisioning`.
4. Развернуть виртуальные машины: `vagrant up`

Примечание 1:

Это может занять много времени (скорость работы сети, доступные ресурсы и т.д.). Если настройка прервется, выполните `vagrant up` снова.

Примечание 2:

Имена хостов виртуальных машин и файл `/etc/hosts` обновятся автоматически.

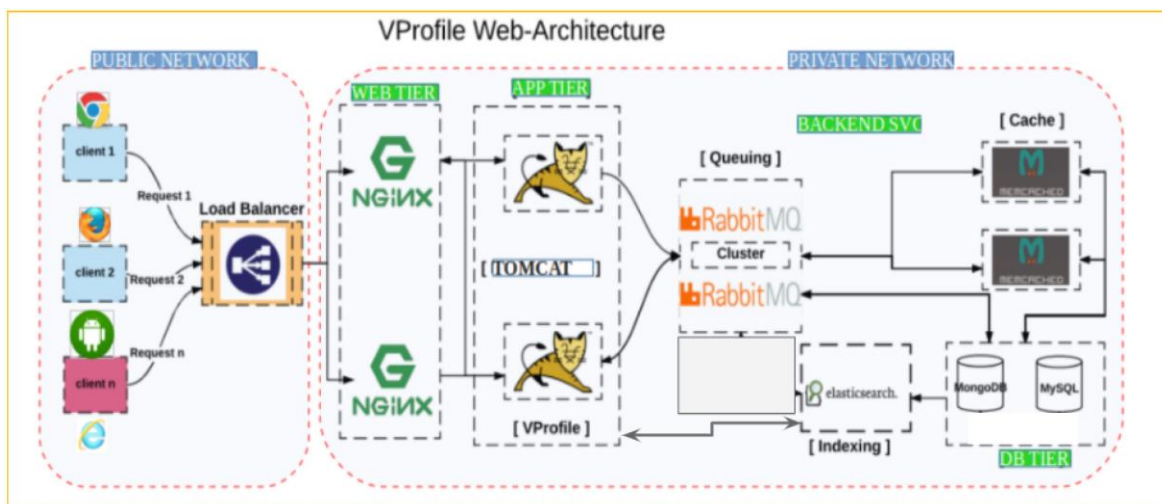
Установка ПО (Provisioning)

Сервисы разворачивать нужно в следующем порядке:

MySQL	SQL Database
Memcached	DB Caching
RabbitMQ	Broker/Queueing Agent
Tomcat	Application Server
Nginx	Web Service

Не разворачиваем в первом упражнении:

ElasticSearch	Indexing/Search service
---------------	-------------------------



Установка MYSQL

Зайти в виртуальную машину **db**:

```
vagrant ssh db01
```

Проверить записи в файле `hosts`, если не хватает – добавьте имена хостов и соответствующие IP-адреса:

```
cat /etc/hosts
```

Обновить ОС

```
yum update -y
```

Установить дополнительные патчи

```
yum install epel-release -y
```

Установить пакет Maria DB

```
yum install git mariadb-server -y
```

Поднять сервис `mariadb-server`

```
systemctl start mariadb  
systemctl enable mariadb
```

Выполнить скрипт настройки безопасности `mysql`

```
mysql_secure_installation
```

```
...  
Remove anonymous users? [Y/n] Y  
... Success!  
...  
Disallow root login remotely? [Y/n] n  
... skipping.  
...  
Remove test database and access to it? [Y/n] Y  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!  
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.  
Reload privilege tables now? [Y/n] Y  
... Success!
```

Примечание:

Установите пароль `root` для `db`, для примера буду использовать пароль `admin123`

Создайте базу данных и пользователей

```
mysql -u root -padmin123  
mysql> create database accounts;  
mysql> grant all privileges on accounts.* TO 'admin'@'%' identified by 'admin123' ;  
mysql> FLUSH PRIVILEGES;  
mysql> exit;
```

Скачайте исходный код и инициализируйте базу данных

```
git clone https://github.com/dogitteamuserzero/devopsclass.git  
cd devopsclass  
mysql -u root -padmin123 accounts < src/main/resources/db_backup.sql  
mysql -u root -padmin123 accounts  
mysql> show tables;
```

Перезагрузите `mariadb-server`

```
systemctl restart mariadb
```

Поднимите брандмауер и дайте `mariadb` доступ к порту 3306

```
systemctl start firewalld  
systemctl enable firewalld  
firewall-cmd --get-active-zones  
firewall-cmd --zone=public --add-port=3306/tcp --permanent  
firewall-cmd --reload  
systemctl restart mariadb
```

Установка MEMCACHED

Установите, запустите memcached и дайте доступ к порту 11211

```
yum install epel-release -y
yum install memcached -y
systemctl start memcached
systemctl enable memcached
systemctl status memcached
memcached -p 11211 -U 11111 -u memcached -d
```

Поднимите брандмауер и дайте Memcached доступ к порту 11211

```
systemctl enable firewalld
systemctl start firewalld
systemctl status firewalld
firewall-cmd --add-port=11211/tcp --permanent
firewall-cmd --reload
memcached -p 11211 -U 11111 -u memcached -d
```

Установка RABBITMQ

Зайти на виртуальную машину RabbitMQ

```
vagrant ssh rmq01
```

Проверить записи в файле hosts, если не хватает – добавьте имена хостов и соответствующие IP-адреса:

```
cat /etc/hosts
```

Обновить ОС

```
yum update -y
```

Установить дополнительные патчи

```
yum install epel-release -y
```

Установить зависимости

```
sudo yum install wget -y
cd /tmp/
wget http://packages.erlang-solutions.com/erlang-solutions-2.0-1.noarch.rpm
sudo rpm -Uvh erlang-solutions-2.0-1.noarch.rpm
sudo yum -y install erlang socat
```

Установить сервер RabbitMQ

```
curl -s https://packagecloud.io/install/repositories/rabbitmq/rabbitmq-
server/script.rpm.sh | sudo bash
sudo yum install rabbitmq-server -y
```

Запустить и разрешить сервис RabbitMQ

```
sudo systemctl start rabbitmq-server
sudo systemctl enable rabbitmq-server
sudo systemctl status rabbitmq-server
```

Изменить конфигурацию

```
sudo sh -c 'echo "[{rabbit, [{loopback_users, []}]}]." >
/etc/rabbitmq/rabbitmq.config'
sudo rabbitmqctl add_user test test
sudo rabbitmqctl set_user_tags test administrator
```

Перезапустить сервис RabbitMQ

```
systemctl restart rabbitmq-server
```

Поднять брандмауер и дать постоянный доступ rabbitmq к порту 25672

```
systemctl start firewalld
systemctl enable firewalld
firewall-cmd --get-active-zones
firewall-cmd --zone=public --add-port=25672/tcp --permanent
firewall-cmd --reload
```

Установка сервера приложений ТОМСАТ

Зайти на виртуальную машину приложения

```
vagrant ssh app01
```

Проверить записи в файле hosts, если не хватает – добавьте имена хостов и соответствующие IP-адреса:

```
cat /etc/hosts
```

Обновить ОС

```
yum update -y
```

Установить дополнительные патчи

```
yum install epel-release -y
```

Установить зависимости

```
yum install java-1.8.0-openjdk -y
```

```
yum install git maven wget -y
```

Перейти в /tmp

```
cd /tmp/
```

Скачать пакет Tomcat

```
wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.37/bin/apache-tomcat-8.5.37.tar.gz  
tar xzvf apache-tomcat-8.5.37.tar.gz
```

Создать пользователя для Tomcat

```
useradd --home-dir /usr/local/tomcat8 --shell /sbin/nologin tomcat
```

Скопировать данные в домашний каталог Tomcat

```
cp -r /tmp/apache-tomcat-8.5.37/* /usr/local/tomcat8/
```

Сделать пользователя владельцем домашнего каталога Tomcat

```
chown -R tomcat.tomcat /usr/local/tomcat8
```

Настроить systemd для Tomcat

Обновить файл следующим образом:

```
nano /etc/systemd/system/tomcat.service
```

```
[Unit]
Description=Tomcat
After=network.target
[Service]
User=tomcat
WorkingDirectory=/usr/local/tomcat8
Environment=JRE_HOME=/usr/lib/jvm/jre
Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment=CATALINA_HOME=/usr/local/tomcat8
Environment=CATALINE_BASE=/usr/local/tomcat8
ExecStart=/usr/local/tomcat8/bin/catalina.sh run
ExecStop=/usr/local/tomcat8/bin/shutdown.sh
SyslogIdentifier=tomcat-%i
[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
```

```
systemctl start tomcat
```

```
systemctl enable tomcat
```

Поднять брандмауер и дать постоянный доступ tomcat к порту 8080

```
systemctl start firewalld
```

```
systemctl enable firewalld
```

```
firewall-cmd --get-active-zones
```

```
firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

```
firewall-cmd --reload
```

Сборка и установка приложения (виртуальная машина *app01*)

Скачать исходный код

```
git clone https://github.com/dogitteamuserzero/devopsclass.git
```

Обновить конфигурацию

```
cd devopsclass
nano src/main/resources/application.properties
```

** Впишите в файл параметры вашего стенда*

Собрать приложение из исходного кода (находясь в папке проекта)

```
mvn install
```

Установить артефакт

```
systemctl stop tomcat
sleep 120
rm -rf /usr/local/tomcat8/webapps/ROOT*
cp target/vprofile-v2.war /usr/local/tomcat8/webapps/ROOT.war
systemctl start tomcat
sleep 300
chown tomcat.tomcat usr/local/tomcat8/webapps -R
systemctl restart tomcat
```

Установка NGINX

Зайти на виртуальную машину *web01*

```
vagrant ssh web01
```

Проверить записи в файле hosts, если не хватает – добавьте имена хостов и соответствующие IP-адреса:

```
cat /etc/hosts
```

Обновить ОС

```
apt update
apt upgrade
```

Установить nginx

```
apt install nginx -y
```

Создать конфиг-файл Nginx и вставить в него следующее:

```
vi /etc/nginx/sites-available/vproapp
```

```
upstream vproapp {
    server app01:8080;
}
server {
    listen 80;
    location / {
        proxy_pass http://vproapp;
    }
}
```

Удалить конфиг по-умолчанию

```
rm -rf /etc/nginx/sites-enabled/default
```

Создать линк для активации веб-сайта

```
ln -s /etc/nginx/sites-available/vproapp /etc/nginx/sites-enabled/vproapp
```

Перезапустить Nginx

```
systemctl restart nginx
```