

Створимо проект Vagrant

Створимо директорию Vagrant_project та перейдемо в її.

Структура проекту буде наступна:

- Основний файл **Vagrantfile** де будуть прописані основні налаштування для віртуальних машин APP_VM та DB_VM
- Файли провідингу **app_provision.sh** та **db_provision.sh** де будуть прописані налаштування до машин APP_VM та DB_VM відповідно

```
dmytro@ubuntu-server:~/dan_it_homeworks/Vagrant_project$ ls -la
total 72
drwxrwxr-x 3 dmytro dmytro 4096 Aug 3 16:33 .
drwxrwxr-x 13 dmytro dmytro 4096 Aug 3 15:05 ..
-rwxrwxr-x 1 dmytro dmytro 2995 Aug 3 16:33 app_provision.sh
-rwxrwxr-x 1 dmytro dmytro 687 Aug 3 15:21 db_provision.sh
-rw-r--r-- 1 dmytro dmytro 46076 Aug 3 15:36 ubuntu-bionic-18.04-cloudimg-console.log
drwxrwxr-x 4 dmytro dmytro 4096 Aug 3 15:10 .vagrant
-rw-rw-r-- 1 dmytro dmytro 4021 Aug 3 15:19 Vagrantfile
dmytro@ubuntu-server:~/dan_it_homeworks/Vagrant_project$
```

Опис файлу Vagrantfile

```
# Вказуємо базовий образ для віртуальної машини
dbvm.vm.box = "ubuntu/bionic64"

# Налаштовуємо приватну мережу з IP адресою для DB_VM
dbvm.vm.network "private_network", ip: "192.168.56.10"

# Встановлюємо hostname для DB_VM
dbvm.vm.hostname = "dbvm"

# Налаштовуємо провізінг для DB_VM за допомогою shell скрипту
# Скрипт db_provision.sh буде виконано для налаштування бази даних
dbvm.vm.provision "shell", path: "db_provision.sh"
end

# Налаштування для віртуальної машини APP_VM
config.vm.define "appvm" do |appvm|
  # Вказуємо базовий образ для віртуальної машини
  appvm.vm.box = "ubuntu/bionic64"

  # Налаштовуємо приватну мережу з IP адресою для APP_VM
  appvm.vm.network "private_network", ip: "192.168.56.11"

  # Встановлюємо hostname для APP_VM
  appvm.vm.hostname = "appvm"

  # Налаштовуємо переадресацію портів для APP_VM
  # Порт 8080 на віртуальній машині буде доступний на порту 8088 на хості
  appvm.vm.network "forwarded_port", guest: 8080, host: 8088

  # Налаштовуємо провізінг для APP_VM за допомогою shell скрипту
  # Скрипт app_provision.sh буде виконано для налаштування додатку
  appvm.vm.provision "shell", path: "app_provision.sh"
end
end
```

Опис файлу db_provision.sh

```
#!/bin/bash

# Оновлення списку пакетів та встановлення MySQL Server
# Це забезпечить, що у вас є остання версія MySQL Server
sudo apt-get update
sudo apt-get install mysql-server -y

# Налаштування MySQL для прийняття підключень з приватної мережі
# Замінюємо значення параметра bind-address на IP адресу приватної мережі
# Це дозволяє з'єднання з базою даних з іншої віртуальної машини
sudo sed -i "s/bind-address.*bind-address = 192.168.56.10/" /etc/mysql/mysql.conf.d/mysqld.cnf

# Перезапускаємо службу MySQL для застосування змін конфігурації
sudo systemctl restart mysql

# Створення нової бази даних з ім'ям 'petclinic'
# Це буде використовуватися додатком для зберігання даних
sudo mysql -u root -e "CREATE DATABASE petclinic;"

# Створення нового користувача 'db_user' з паролем 'db_pass'
# Користувач буде використовуватися для доступу до бази даних 'petclinic'
# '%' дозволяє підключення з будь-якого IP-адреси
sudo mysql -u root -e "CREATE USER 'db_user'@ '%' IDENTIFIED BY 'db_pass';"

# Надання всіх привілеїв користувачу 'db_user' на базу даних 'petclinic'
# Це дозволяє користувачу виконувати всі операції з базою даних
sudo mysql -u root -e "GRANT ALL PRIVILEGES ON petclinic.* TO 'db_user'@ '%' ;"

# Оновлюємо привілеї для застосування змін
sudo mysql -u root -e "FLUSH PRIVILEGES;"
```

Опис файлу `app_provision.sh`

```
#!/bin/bash

# Оновлення списку пакетів та встановлення необхідних пакетів
# Встановлюємо OpenJDK 17 для запуску Java-додатків
# Встановлюємо Git для клонування репозиторіїв
# Встановлюємо Maven для зборки Java-проектів
sudo apt-get update
sudo apt-get install -y openjdk-17-jdk git maven

# Створюємо нового користувача app_user без пароля
# Цей користувач буде використовуватися для виконання додатків
sudo adduser --disabled-password --gecos "" app_user

# Виконуємо команди як користувач app_user
# Клонуємо репозиторій Spring PetClinic в домашню директорію app_user
# Збираємо проект за допомогою Maven
# Переміщуємо зібраний .jar файл до домашньої директорії app_user
sudo -i -u app_user bash << EOF
git clone https://github.com/spring-projects/spring-petclinic.git /home/app_user/project
cd /home/app_user/project
./mvnw package

# Переміщуємо зібраний .jar файл до домашньої директорії app_user
mv target/*.jar /home/app_user/app.jar
EOF

# Налаштовуємо середовищні змінні для підключення до бази даних
# Ці змінні будуть доступні для додатку при його запуску
echo "export DB_HOST=192.168.56.10" >> /home/app_user/.profile
echo "export DB_PORT=3306" >> /home/app_user/.profile
echo "export DB_NAME=petclinic" >> /home/app_user/.profile
echo "export DB_USER=db_user" >> /home/app_user/.profile
echo "export DB_PASS=db_pass" >> /home/app_user/.profile

# Створюємо файл сервісу для запуску додатку при завантаженні системи
# Налаштовуємо сервіс для запуску як app_user, з визначеними середовищними змінними
# Запускаємо додаток за допомогою java -jar
sudo tee /etc/systemd/system/petclinic.service > /dev/null << EOF
[Unit]
Description=PetClinic Spring Boot Application
After=network.target

[Service]
User=app_user
Environment=DB_HOST=192.168.56.10
Environment=DB_PORT=3306
Environment=DB_NAME=petclinic
Environment=DB_USER=db_user
Environment=DB_PASS=db_pass
ExecStart=/usr/bin/java -jar /home/app_user/app.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
EOF

# Оновлюємо конфігурацію systemd для врахування нового сервісу
# Запускаємо сервіс та додаємо його до автозапуску при завантаженні системи
sudo systemctl daemon-reload
sudo systemctl start petclinic
sudo systemctl enable petclinic
```

Результат збірки

```
appvm: Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/3.4.2/plexus-io-3.4.2.jar (79 kB at 497 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/3.4.2/plexus-io-3.4.2.jar (79 kB at 497 kB/s)
appvm: Downloading from central: https://repo.maven.apache.org/maven2/com/github/luben/zstd-jni/1.5.5-11/zstd-jni-1.5.5-11.jar (6.8 MB at 2.6 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/github/luben/zstd-jni/1.5.5-11/zstd-jni-1.5.5-11.jar (6.8 MB at 2.6 MB/s)
appvm: [INFO] Building jar: /home/app_user/project/target/spring-petclinic-3.3.0-SNAPSHOT.jar
appvm: [INFO] --- spring-boot:3.3.0:repackage (repackage) @ spring-petclinic ---
appvm: [INFO] Replacing main artifact /home/app_user/project/target/spring-petclinic-3.3.0-SNAPSHOT.jar with repackaged archive, adding nested dependencies
appvm: [INFO] The original artifact has been renamed to /home/app_user/project/target/spring-petclinic-3.3.0-SNAPSHOT.jar.original
appvm: [INFO] BUILD SUCCESS
appvm: [INFO] Total time: 07:59 min
appvm: [INFO] Finished at: 2024-08-03T15:50:23Z
appvm: [INFO] Created symlink /etc/systemd/system/multi-user.target.wants/petclinic.service -> /etc/systemd/system/petclinic.service.
```

Перевірка статусу сервісу та пінгування із хостової машини

```
vagrant@appvm:~$ sudo systemctl start petclinic
vagrant@appvm:~$ sudo systemctl status petclinic
● petclinic.service - PetClinic Spring Boot Application
   Loaded: loaded (/etc/systemd/system/petclinic.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-08-03 15:50:26 UTC; 7min ago
     Main PID: 8489 (java)
       Tasks: 32 (limit: 1151)
    CGroup: /system.slice/petclinic.service
            └─8489 /usr/bin/java -jar /home/app_user/app.jar

Aug 03 15:50:41 appvm java[8489]: 2024-08-03T15:50:41.138Z INFO 8489 --- [
Aug 03 15:50:41 appvm java[8489]: 2024-08-03T15:50:41.391Z INFO 8489 --- [
Aug 03 15:50:41 appvm java[8489]: 2024-08-03T15:50:41.523Z INFO 8489 --- [
Aug 03 15:50:42 appvm java[8489]: 2024-08-03T15:50:42.283Z INFO 8489 --- [
Aug 03 15:50:45 appvm java[8489]: 2024-08-03T15:50:45.578Z INFO 8489 --- [
Aug 03 15:50:45 appvm java[8489]: 2024-08-03T15:50:45.585Z INFO 8489 --- [
Aug 03 15:50:46 appvm java[8489]: 2024-08-03T15:50:46.512Z INFO 8489 --- [
Aug 03 15:50:50 appvm java[8489]: 2024-08-03T15:50:50.226Z INFO 8489 --- [
Aug 03 15:50:50 appvm java[8489]: 2024-08-03T15:50:50.547Z INFO 8489 --- [
Aug 03 15:50:50 appvm java[8489]: 2024-08-03T15:50:50.608Z INFO 8489 --- [
   main] o.hibernate.jpa.internal.util.LogHelper : HHN000284: Processing Persistence
   main] org.hibernate.Version : HHN000412: Hibernate ORM core vers
   main] o.h.c.internal.RegionFactoryInitiator : HHN000026: Second-level cache dis
   main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; ignoring
   main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHN000489: No JTA platform availab
   main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFacto
   main] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If appl
   main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 14 endpoints beneath base
   main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http
   main] o.s.s.petclinic.PetClinicApplication : Started PetClinicApplication in 22
```

```
vagrant@appvm:~$ sudo systemctl enable petclinic
vagrant@appvm:~$
vagrant@appvm:~$ logout
dmytro@ubuntuserver:~/dan_it_homeworks/Vagrant_project$ ping 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data:
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=0.970 ms
64 bytes from 192.168.56.11: icmp_seq=2 ttl=64 time=0.610 ms
64 bytes from 192.168.56.11: icmp_seq=3 ttl=64 time=0.458 ms
64 bytes from 192.168.56.11: icmp_seq=4 ttl=64 time=0.617 ms
64 bytes from 192.168.56.11: icmp_seq=5 ttl=64 time=0.466 ms
64 bytes from 192.168.56.11: icmp_seq=6 ttl=64 time=0.506 ms
```

Відкриття додатку через по IP від хостової машини

```
dmytro@ubuntuserver:~/dan_it_homeworks/Vagrant_project$ nc -zv 192.168.56.11 8080
Connection to 192.168.56.11 8080 port [tcp/http-alt] succeeded!
dmytro@ubuntuserver:~/dan_it_homeworks/Vagrant_project$ curl -I http://192.168.56.11:8080
HTTP/1.1 200
Content-Type: text/html; charset=UTF-8
Content-Language: en
Transfer-Encoding: chunked
Date: Sat, 03 Aug 2024 16:07:17 GMT

dmytro@ubuntuserver:~/dan_it_homeworks/Vagrant_project$ curl http://192.168.56.11:8080
<!DOCTYPE html>

<html>

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="shortcut icon" type="image/x-icon" href="/resources/images/favicon.png">

  <title>PetClinic :: a Spring Framework demonstration</title>

  <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]-->

  <link href="/webjars/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
  <link rel="stylesheet" href="/resources/css/petclinic.css" />

</head>

<body>

  <nav class="navbar navbar-expand-lg navbar-dark" role="navigation">
    <div class="container-fluid">
```