

# Dennis Tikhomirov. DevOps Spring'19.

## Homework Task #4 (Vagrant).

Available here [https://github.com/dennis00010011b.epam-devops-training/blob/master/Task4/Dennis%20Tikhomirov\\_HW\\_Task4.docx](https://github.com/dennis00010011b.epam-devops-training/blob/master/Task4/Dennis%20Tikhomirov_HW_Task4.docx)

### 1. Vagrant vs VirtualBox

#### 1.1. Install VirtualBox

#### 1.2. Install Vagrant

#### 1.3. Create a Vagrantfile based on the box Windows 10

##### 1.3.1. Install Windows 10 with network settings

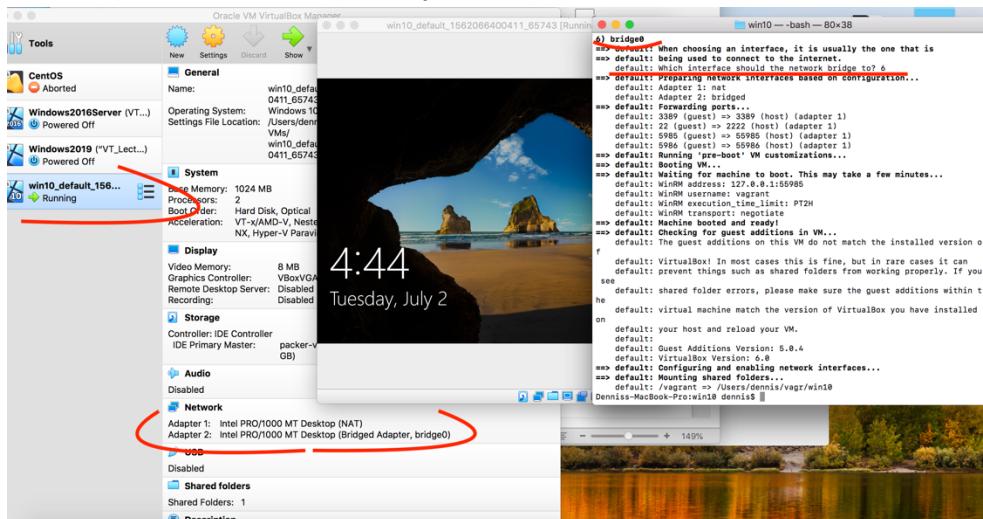
### Vagrantfile

```
Vagrant.configure("2") do |config|  
  config.vm.box = "senglin/win-10-enterprise-vs2015community"  
  config.vm.box_version = "1.0.0"  
  config.vm.network "public_network"  
  config.vm.provider "virtualbox" do |vb|  
    vb.gui = true  
    vb.memory = 1024  
    vb.cpus = 2  
  end  
end
```

### Commands

```
vagrant init senglin/win-10-enterprise-vs2015community \ --box-version 1.0.  
0  
vagrant up
```

### Screenshot #1.1. Installed Win10 on VirtualBox



### Notice:

Downloading too slow !!!! <https://github.com/hashicorp/vagrant/issues/8434>

Faster way to install package it is straight download from source (url is shown in loading process ) and next to install it:

1. In Vagrantfile specify relative path to package.box file:

```
config.vm.box_url = "$relativePath/package.box"
```

2. Run command

```
Vagrant up
```

### 1.3.2. Connect to the box via vagrant ssh

```
vagrant status          // show VMs
vagrant ssh default    // connect to VM [default]
-sh-4.1$ systeminfo    // Windows command
```

Screenshot #1.2. Connect to VirtualBox VM trough Vagrant

```
Denniss-MacBook-Pro:win10 dennis$ vagrant status
Current machine states:
default      running (virtualbox)

The VM is running. To stop this VM, you can run 'vagrant halt' to
shut it down forcefully, or you can run 'vagrant suspend' to simply
suspend the virtual machine. In either case, to restart it again,
simply run 'vagrant up'.
Denniss-MacBook-Pro:win10 dennis$ vagrant ssh default
Last login: Tue Jul 2 05:06:45 2019 from 10.0.2.2
[sh-4.1$ vagrant
[sh: va: command not found
[sh-4.1$ systeminfo

Host Name:          VAGRANT-10
OS Name:           Microsoft Windows 10 Enterprise Evaluation
OS Version:        10.0.10240 N/A Build 10240
OS Manufacturer:  Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type:   Multiprocessor Free
Registered Owner:
Registered Organization: Vagrant Inc.
Product ID:         00329-20000-00001-AA976
Original Install Date: 9/18/2015, 11:44:03 PM
System Boot Time:   7/2/2019, 4:40:12 AM
System Manufacturer: innotek GmbH
System Model:       VirtualBox
System Type:        x64-based PC
Processor(s):      1 Processor(s) Installed.
                    [01]: Intel64 Family 6 Model 58 Stepping 9 GenuineIntel ~2395 Mhz
BIOS Version:       innotek GmbH VirtualBox, 12/1/2006
Windows Directory: C:\Windows
System Directory:  C:\Windows\system32
Boot Device:        \Device\HarddiskVolume1
System Locale:     en-us;English (United States)
Input Locale:      en-us;English (United States)
Time Zone:         (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory: 1,024 MB
Available Physical Memory: 333 MB
Virtual Memory: Max Size: 2,752 MB
Virtual Memory: Available: 1,879 MB
Virtual Memory: In Use: 873 MB
Page File Location(s): C:\pagefile.sys
Domain:            WORKGROUP
Logon Server:      \\VAGRANT-10
Hotfix(s):         7 Hotfix(s) Installed.
                    [01]: KB3074678
                    [02]: KB3074666
                    [03]: KB3081444
                    [04]: KB3081449
                    [05]: KB3081452
                    [06]: KB3081455
                    [07]: KB3087916
Network Card(s):  2 NIC(s) Installed.
                    [01]: Intel(R) PRO/1000 MT Desktop Adapter
```

### 1.3.3. Remove box

```
vagrant halt default          // gracefully stop VM (or --force)
vagrant destroy default       //remove VM.
```

**1.4. Create a multi-box configuration with a **local network**, using for three client VM boxes with Ubuntu 1804 and for a server – VM box with Windows 10**

**1.4.1. Using provisioning install MySQL Server in Windows OS and configure guest connection to MySQL Server using different usernames for client machines**

**1.4.2. Using provisioning install MySQL Client in all Ubuntu OS and set up connections**

**1.4.3. Up all boxes with one command and connect via vagrant ssh to all boxes and check the connection of clients to the MySQL Server**

I used Ubuntu18.04 instead Windows10 as a MySQL server

Vagrant file

```
Vagrant.configure("2") do |config|
  #MYSQL SERVER
    config.vm.define "SServerMySQL" do |server|
      config.vm.provider "virtualbox" do |vb|
        vb.memory = "1024"
      end
      server.vm.box = "geerlingguy/ubuntu1804"
      server.vm.network "private_network", ip: "192.168.50.10"
      server.vm.provision "shell", inline:<<-SHELL
        DIR=/etc/mysql/
        IP=192.168.50
        sudo ufw allow from any to any port 3306 proto tcp
        sudo apt-get update
        sudo apt-get install --reinstall iptables
        sudo apt-get install net-tools
        sudo apt-get install -y mysql-server
        sudo touch $DIR/init.sql
        sudo chmod 777 $DIRinit.sql
        sudo echo "CREATE USER user IDENTIFIED BY
'pass';">>$DIR/init.sql
        sudo echo "CREATE USER user1@$IP.10 IDENTIFIED BY
'pass1';">>$DIR/init.sql

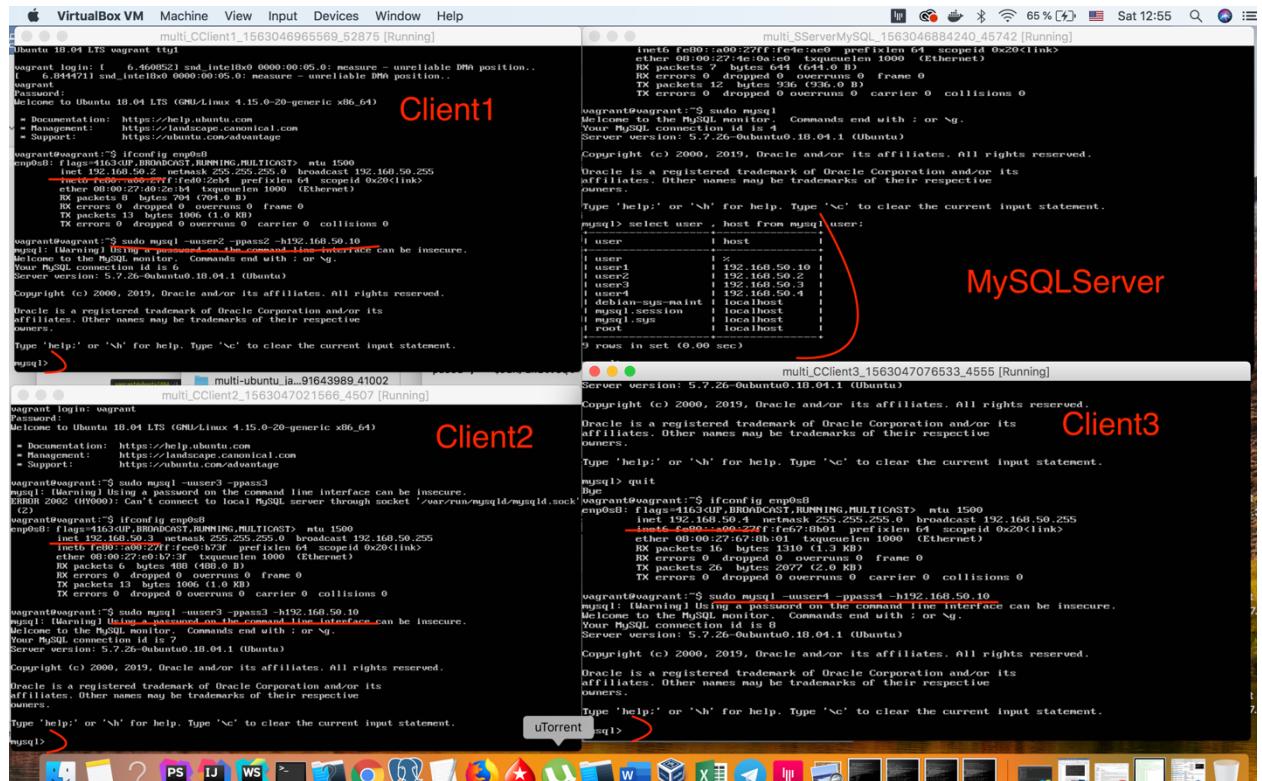
        sudo echo "CREATE USER user2@$IP.2 IDENTIFIED BY
'pass2';">>$DIR/init.sql
        sudo echo "CREATE USER user3@$IP.3 IDENTIFIED BY
'pass3';">>$DIR/init.sql
        sudo echo "CREATE USER user4@$IP.4 IDENTIFIED BY
'pass4';">>$DIR/init.sql
        sudo cat $DIR/init.sql
        sudo echo "[mysqld]" >$DIR/my.cnf
        sudo echo "init-file=/etc/mysql/init.sql" >>$DIR/my.cnf
        sudo echo "bind-address = 0.0.0.0" >>$DIR/my.cnf
        sudo chmod 444 $DIR/my.cnf
        sudo cat $DIR/my.cnf
        sudo /etc/init.d/mysql restart
    SHELL
End
```

```

# MYSQL CLIENTS
(1..3).each do |i|
  config.vm.define "CClient#{i}" do |ubuntu|
    config.vm.provider "virtualbox" do |vb|
      vb.memory = "1024"
    end
    ubuntu.vm.box = "geerlingguy/ubuntu1804"
    ubuntu.vm.network "private_network", ip: "192.168.50.#{i+1}"
    ubuntu.vm.provision "shell", inline:<<-SHELL
      sudo apt-get update
      sudo apt-get install net-tools
      sudo apt-get install -y mysql-client-core-5.7
    SHELL
  end
end
end

```

Screenshot #1.3. MySQL server and three Ubunru VMs are logged in



#### **1.4.4. Using vagrant install the JDK in Windows OS. After installation and configuration, view the java version.**

```
vagrant ssh default  
choco install jdk8  
PATH=$PATH;$JAVA_HOME
```

Screenshot#1.4. Enviroment variables on Windows VM.

The screenshot shows a terminal window titled "1. ssh". The terminal displays the following command sequence:

```
-sh-4.1$ echo $JAVA_HOME  
C:\Program Files\Java\jdk1.8.0_211  
-sh-4.1$ PATH=$PATH;$JAVA_HOME  
-sh: C:\Program: command not found  
-sh-4.1$ echo $PATH  
/bin:/cygdrive/c/Windows/system32:/cygdrive/c/Windows/System32/Wbem:/cygdrive/c/Windows/System32/WindowsPowerShell/v1.0:/usr/bin:/cygdrive/c/ProgramData/chocolatey/bin:/cygdrive/c/Program Files/Git/cmd:/cygdrive/c/Users/vagrant/.dnx/bin:/cygdrive/c/Program Files/Microsoft DNX/Dnvm:/cygdrive/c/Program Files/Java/jdk1.8.0_211/bin  
-sh-4.1$  
4.1$  
-sh-4.1$  
-sh-4.1$  
-sh-4.1$  
-sh-4.1$  
-sh-4.1$  
-sh-4.1$  
-sh-4.1$  
-sh-4.1$  
Error: Could not create the Java Virtual Machine.  
Error: A fatal exception has occurred. Program will exit.  
Unrecognized option: -v  
-sh-4.1$ java -version  
java version "1.8.0_211"  
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)  
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)  
-sh-4.1$
```

Red arrows highlight several parts of the command sequence, specifically the environment variable assignment, the PATH command, the Java command, and the version output.

#### **1.4.5. Destroy via vagrant VM with Ubuntu**

From Vagrantfile directory:

```
vagrant destroy Test1      //remove VM  
vagrant destroy Test2      //remove VM  
vagrant destroy Test3      //remove VM
```

## 1.5. Create your own VagrantBox based on Ubuntu 1804

### 1.5.1. Based on the Ubuntu 1804 image, create a VM in VirtualBox

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "generic/ubuntu1804"
end
```

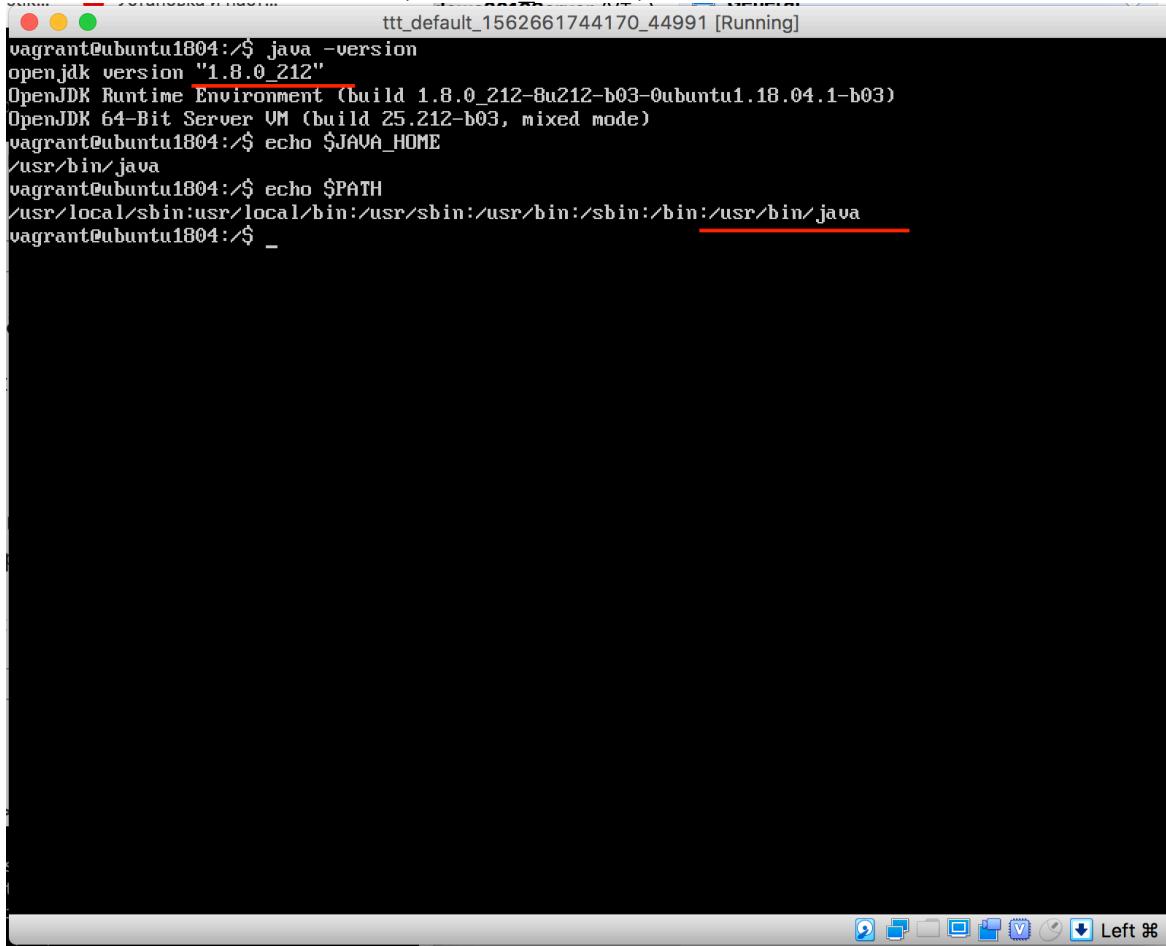
Commands

```
vagrant init generic/ubuntu1804
vagrant up
```

### 1.5.2. Install OpenJDK and configure on this VM

```
sudo apt-get install -y openjdk-8-jre
sudo apt-get install -y openjdk-8-jdk
whereis java
JAVA_HOME=/usr/bin/java
export PATH=$PATH:$JAVA_HOME
```

Screenshot #1.5. Java version, \$JAVA\_HOME , \$PATH values



The screenshot shows a terminal window titled 'ttt\_default\_1562661744170\_44991 [Running]'. The terminal output is as follows:

```
vagrant@ubuntu1804:~$ java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-8u212-b03-0ubuntu1.18.04.1-b03)
OpenJDK 64-Bit Server VM (build 25.212-b03, mixed mode)

vagrant@ubuntu1804:~$ echo $JAVA_HOME
/usr/bin/java

vagrant@ubuntu1804:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/java
```

The '\$JAVA\_HOME' and '\$PATH' lines are highlighted with red boxes.

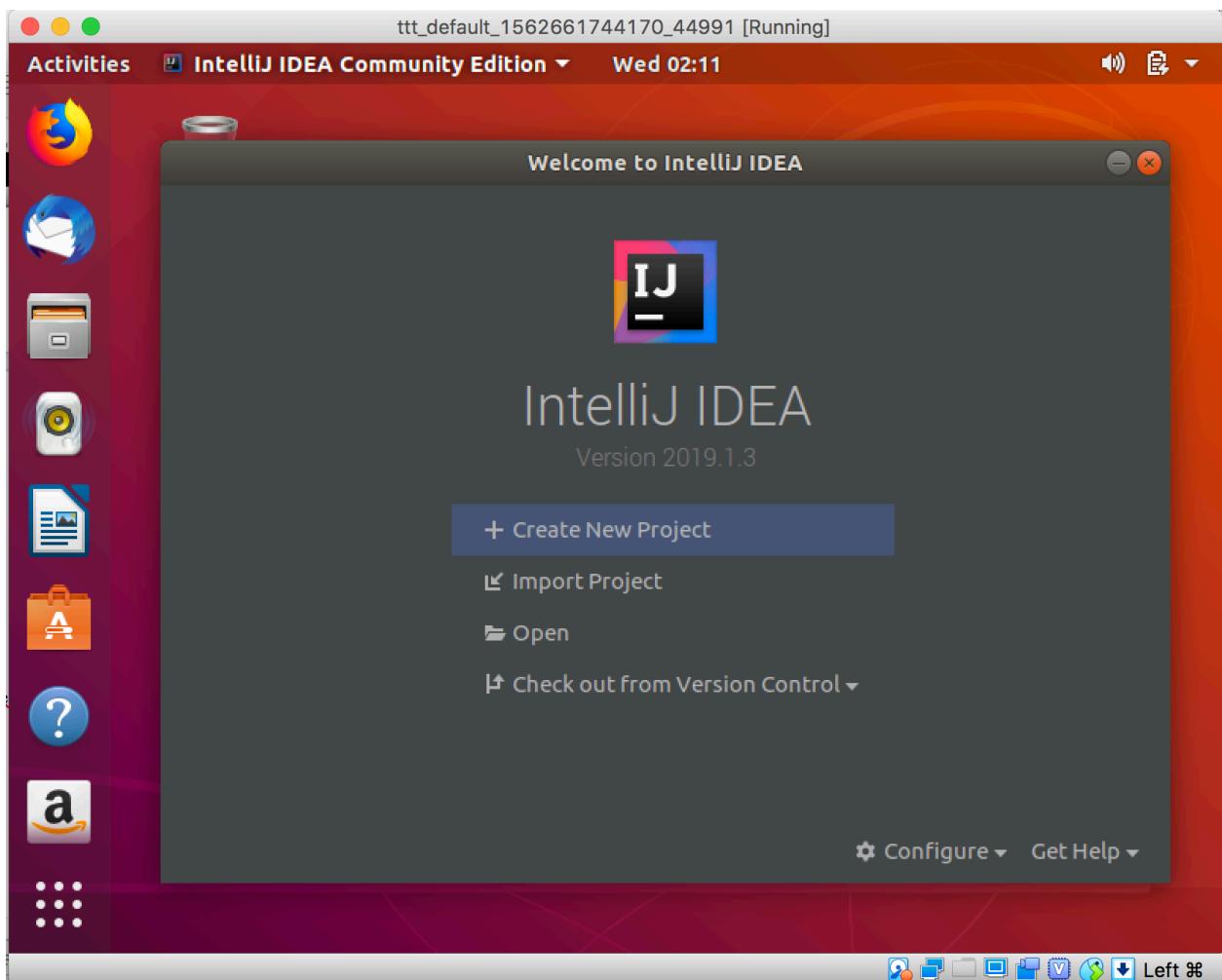
### 1.5.3. Install IntelliJ IDEA on this VM

```
sudo snap install intellij-idea-community --classic
```

Upgrade Ubuntu to Desktop version (add (GUI)

```
sudo apt-get install ubuntu-desktop
```

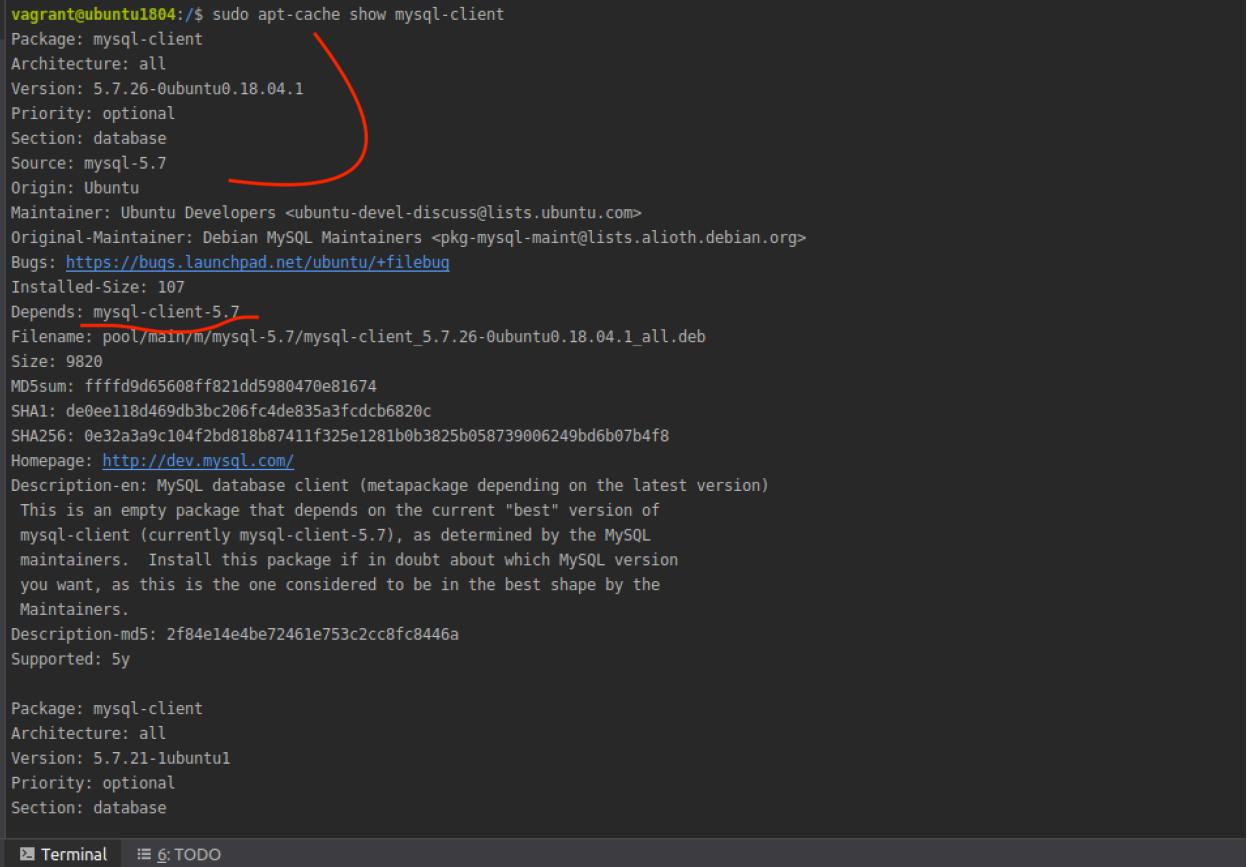
Screenshot #1.6. Installed IntelliJ Idea



#### 1.5.4. Install MySQL Client

```
sudo apt-get install mysql-client
```

Screenshot #1.7. Installed mysql-client



```
vagrant@ubuntu1804:~$ sudo apt-cache show mysql-client
Package: mysql-client
Architecture: all
Version: 5.7.26-0ubuntu0.18.04.1
Priority: optional
Section: database
Source: mysql-5.7
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian MySQL Maintainers <pkg-mysql-maint@lists.alioth.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 107
Depends: mysql-client-5.7
Filename: pool/main/m/mysql-5.7/mysql-client_5.7.26-0ubuntu0.18.04.1_all.deb
Size: 9820
MD5sum: fffffd9d65608ff821dd5980470e81674
SHA1: de0ee18d469db3bc206fc4de835a3fcdb6820c
SHA256: 0e32a3a9c104f2bd818b87411f325e1281b0b3825b058739006249bd6b07b4f8
Homepage: http://dev.mysql.com/
Description-en: MySQL database client (metapackage depending on the latest version)
  This is an empty package that depends on the current "best" version of
  mysql-client (currently mysql-client-5.7), as determined by the MySQL
  maintainers.  Install this package if in doubt about which MySQL version
  you want, as this is the one considered to be in the best shape by the
  Maintainers.
Description-md5: 2f84e14e4be72461e753c2cc8fc8446a
Supported: 5y

Package: mysql-client
Architecture: all
Version: 5.7.21-1ubuntu1
Priority: optional
Section: database
```

#### 1.5.5. Make your own VagrantBox

Follow Oracle documentation for creating base box for Virtulabox provider :

<https://oracle-base.com/articles/vm/create-a-vagrant-base-box-virtualbox>

VM have to be configured (items 1-3).

- 1.Add Guest addition : it has already added in native box generic/ubuntu1804
2. The "root" user password must be set to "vagrant" and there needs to be a user called "vagrant" with a password of "vagrant".
- 3.Add insecure public keys

```
# Add insecure public key.
mkdir /home/vagrant/.ssh
wget -O /home/vagrant/.ssh/authorized_keys
https://raw.githubusercontent.com/hashicorp/vagrant/master/keys/vagrant.pub
chown -R vagrant:vagrant /home/vagrant/.ssh
chmod 0700 /home/vagrant/.ssh
chmod 0600 /home/vagrant/.ssh/authorized_keys
```

4. Add *metadata.json*:

```
{  
  "provider": "virtualbox"  
}
```

5. Run command:

```
vagrant package --base ubuntu1804
```

6. Upload file *package.box* to Vagrant cloud <https://app.vagrantup.com/>

Resulting box:

<https://app.vagrantup.com/dennis00010011b/boxes/ubuntu1804-desktop-java-intellij-mysqlclient/versions/0.3>

7. Add box for local usage:

```
vagrant box add package.box --name CUSTOM
```

### 1.5.6. Destroy VM Ubuntu 1804

From Vagrantfile directory:

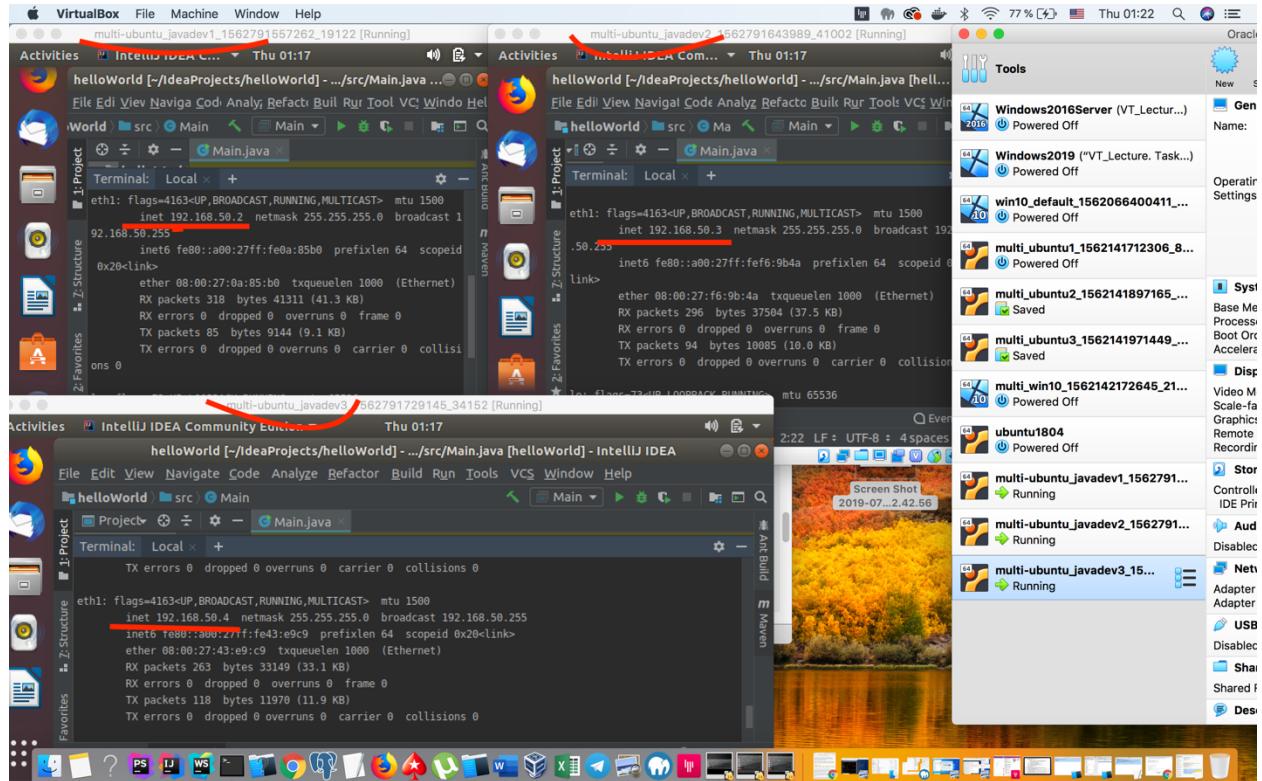
```
vagrant halt default          // gracefully stop VM (or --force)  
vagrant destroy default      //remove VM
```

### 1.5.7. From the box was created in section 1.5.5, create Vagrantfile with three machines and configure the local network

Vagrantfile

```
Vagrant.configure("2") do |config|  
  (1..3).each do |i|  
    config.vm.define "javadev#{i}" do |ubuntu|  
      #it was used for debugging purpose , to avoid the waisting time for  
      #uploading/downloading to Vagrant cloud  
      #ubuntu.vm.box_url = "../ttt/package.box"  
  
      ubuntu.vm.box = "dennis00010011b/ubuntu1804-desktop-java-intellij-mysqlclient"  
      ubuntu.vm.box_version = "0.3"  
      ubuntu.vm.network "private_network", ip: "192.168.50.#{i+1}"  
    end  
  end  
end
```

## Screenshot #1.8. Three VMs from own Vagrant box



**Notice.** New created base box should be tested. It can be done by upload and download to Vagrant cloud but it takes a lot of time. Decision could be to use local file package.box.

- in Vagrantfile specify name box as local path

```
ubuntu.vm.box = "CUSTOM"
```

- Unfortunately versioning will be ignored .In Vagrantfile comment version :

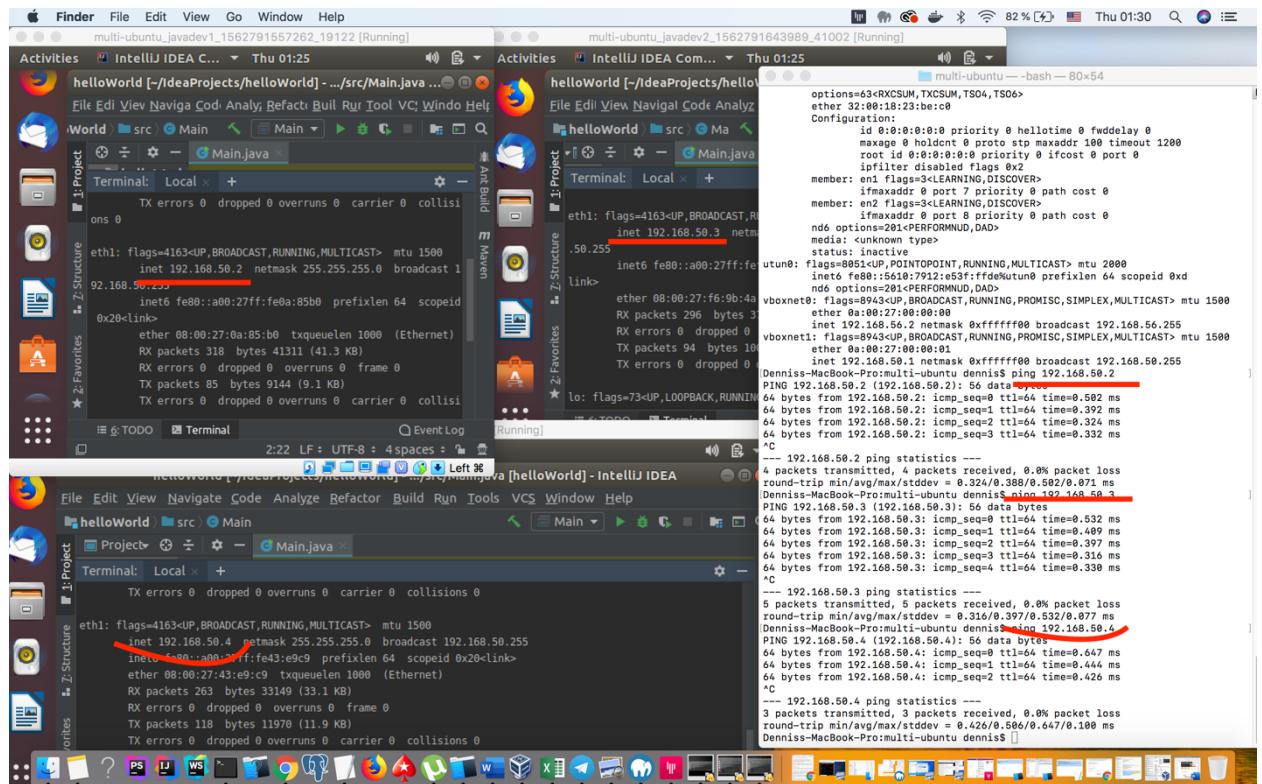
```
#ubuntu.vm.box_version = "0.3"
```

### 1.5.8. Check LAN connections

LAN connection was already configurated in Vagrantfile

```
ubuntu.vm.network "private_network", ip: "192.168.50.#{i+1}"
```

Screenshot #1.9. LAN connection



### 1.5.9. Check connection to MySQL server

All VMs are connected to MySQL server 198.162.50.10:

```
mysql -uuser -ppass -h192.168.50.10
```

Screenshot #1.10. Connections to MySQL Server

The screenshot shows two terminal windows side-by-side. The left window is titled "multi-ubuntu\_javadev2\_1562791729145\_34102 [Running]" and the right window is titled "multi-ubuntu\_mySQL\_156304684240\_45742 [Running]". Both windows are running MySQL command-line interfaces.

**Left Terminal (multi-ubuntu\_javadev2):**

```
vagrant@ubuntu1804:~$ ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.50.2  brd 192.168.50.255  netmask 255.255.255.0
        ether 00:0c:27:ff:fe04:9b  txqueuelen 1000  (Ethernet)
            RX packets 10  bytes 886 (886.0 B)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 78  bytes 936 (936.0 B)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
vagrant@ubuntu1804:~$ sudo mysql -uuser -ppass -h192.168.50.10
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
```

**Right Terminal (multi-ubuntu\_mySQL):**

```
inet6 fe80::a00:27ff:fe04:9b%eth1  brd fe80::ff:27ff:fe04:9b  scopeid 0x20<link>
      ether 00:0c:27:ff:fe04:9b  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 32  bytes 936 (936.0 B)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
root@multi-ubuntu_mySQL:~$ mysql -uuser -ppass -h192.168.50.10
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
```

A red box highlights the word "Server" in the top right of the MySQL prompt in the right terminal window.

## 2. Vagrant vs Hyper-V

### 2.1. Install Hyper-V

### 2.2. Install Vagrant

### 2.3. Create a Vagrantfile based on the box Windows 10

#### 2.3.1. Up three Windows 10 with network settings

**Notice:** Sometime `vagrant up` command doesn't start with Hyperv boxes for some reason. In this case provider should be specified: `vagrant up --provider=hyperv`

#### Vagrantfile

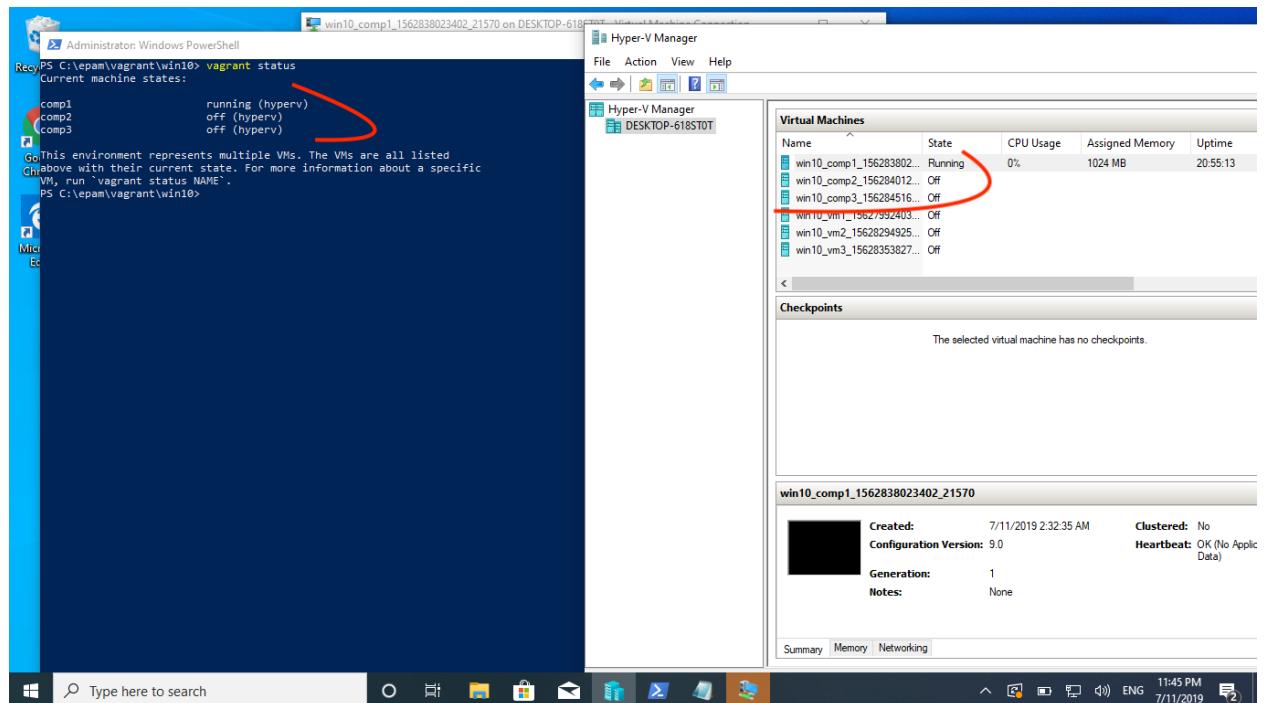
```
Vagrant.configure("2") do |config|
  (1..3).each do |i|
    config.vm.define "comp#{i}" do |win|
      config.win.provider "hyperv" do |h|
        h.vmname = "serverMySQL"
        h.memory = 1024000
        h.cpus = 1
      end

      win.vm.box = "gusztavvargadr/windows-10"
      win.vm.boot_timeout=600
    end
  end
end
```

#Anyway it doesn't work because Vagrant can't create and config new networks  
#for Hyper-V

```
  win.vm.network "private_network", type: "dhcp"
end
end
```

Screenshot #2.1. Installed VMs



### 2.3.2 Connect to the box via vagrant ssh

Screenshot #2.2. SSH connection to Guest

```
PS C:\> Select Administrator: c:\windows\system32\cmd.exe
comp1      running (hyperv)
comp2      off (hyperv)
comp3      off (hyperv)

This environment represents multiple VMs. The VMs are all listed above with their current state. For more information about a specific VM, run `vagrant status NAME`.

PS C:\> cd \vagrant\win10; vagrant ssh comp1
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

vagrant@DESKTOP-H8IB9NG C:\Users\vagrant>systeminfo

Host Name: DESKTOP-H8IB9NG
OS Name: Microsoft Windows 10 Enterprise Evaluation
OS Version: 10.0.18362 N/A Build 18362
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free
Registered Owner:
Registered Organization: gusztavvargadr
Product ID: 00329-20000-00001-AA558
Original Install Date: 7/11/2019, 9:52:00 AM
System Boot Time: 7/12/2019, 2:16:49 AM
System Manufacturer: Microsoft Corporation
System Model: Virtual Machine
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 61 Stepping 4 GenuineIntel ~2195 Mhz
BIOS Version: American Megatrends Inc. 090008 , 12/7/2018
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC) Coordinated Universal Time
Total Physical Memory: 1,034 MB
Available Physical Memory: 136 MB
Virtual Memory: Max Size: 2,752 MB
Virtual Memory: Available: 1,255 MB
Virtual Memory: In Use: 1,497 MB
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: \\WIN-D2080J25E0
Hotfix(s): 5 Hotfix(s) Installed.
[01]: KB4497727
[02]: KB4497727
[03]: KB4497727
[04]: KB4503308

Status: Running |
```

```
win10_comp1_1562838023402_21570 on DESKTOP-618ST0T - Virtual Machine Connection
File Action Media View Help
Administrator: Windows PowerShell https://aka.ms/powershell
PS C:\Users\vagrant> systeminfo

Host Name: DESKTOP-H8IB9NG
OS Name: Microsoft Windows 10 Enterprise Evaluation
OS Version: 10.0.18362 N/A Build 18362
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free
Registered Owner:
Registered Organization: gusztavvargadr
Product ID: 00329-20000-00001-AA558
Original Install Date: 7/11/2019, 9:52:00 AM
System Boot Time: 7/12/2019, 2:16:49 AM
System Manufacturer: Microsoft Corporation
System Model: Virtual Machine
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 61 Stepping 4 GenuineIntel ~2195 Mhz
BIOS Version: American Megatrends Inc. 090008 , 12/7/2018
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC) Coordinated Universal Time
Total Physical Memory: 1,024 MB
Available Physical Memory: 73 MB
Virtual Memory: Max Size: 2,752 MB
Virtual Memory: Available: 1,243 MB
Virtual Memory: In Use: 1,509 MB
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: \\WIN-D2080J25E0
Hotfix(s): 5 Hotfix(s) Installed.
[01]: KB4497727
[02]: KB4497727
[03]: KB4497727
[04]: KB4503308

Status: Running |
```

### 2.3.3. Change the communicator to "winrm"

### 2.3.4. Connect to box via winrm. Check LAN connections.

### 2.3.5. Destroy box

## 2.4 Create a multi-box configuration with a local network, using for three client VM boxes with Windows 10 OS and for a server VM box with Ubuntu 1804 OS

### 2.4.1. Using provisioning in the Ubuntu OS, install MySQL Server and set up a guest connection to MySQL using different usernames for client machines

### 2.4.2. Using Windows provisioning, install the MySQL client and configure a guest connection to the MySQL server using different usernames for client machines.

## Vagrantfile

```
1. Vagrant.configure("2") do |config|
2.   #MYSQL SERVER
3.     config.vm.define "SServerMySQL" do |server|
4.       config.server.provider "hyperv" do |h|
5.         h.vmname = "serverMySQL"
6.         h.memory = 1024000
7.         h.cpus = 1
8.       end
9.       server.vm.box = "geerlingguy/ubuntu1804"
10.      server.vm.network "private_network", ip: "192.168.50.10"
11.      server.vm.provision "shell", inline:<<-SHELL
12.        DIR=/etc/mysql/
13.        IP=192.168.50
14.        sudo ufw allow from any to any port 3306 proto tcp

15.        sudo apt-get update
16.        sudo apt-get install --reinstall iptables
17.        sudo apt-get install net-tools
18.        sudo apt-get install -y mysql-server
19.        sudo touch $DIR/init.sql
20.        sudo chmod 777 $DIRinit.sql
21.        sudo echo "CREATE USER user IDENTIFIED BY
'pass';">>$DIR/init.sql
22.        sudo echo "CREATE USER user1@$IP.10 IDENTIFIED BY
'pass1';">>$DIR/init.sql
23.
24.        sudo echo "CREATE USER user2@$IP.2 IDENTIFIED BY
'pass2';">>$DIR/init.sql
25.        sudo echo "CREATE USER user3@$IP.3 IDENTIFIED BY
'pass3';">>$DIR/init.sql
26.        sudo echo "CREATE USER user4@$IP.4 IDENTIFIED BY
'pass4';">>$DIR/init.sql
27.        sudo cat $DIR/init.sql
28.        sudo echo "[mysqld]" >$DIR/my.cnf
29.        sudo echo "init-file=/etc/mysql/init.sql"
>>$DIR/my.cnf
30.        sudo echo "bind-address = 0.0.0.0" >>$DIR/my.cnf
31.        sudo chmod 444 $DIR/my.cnf
32.        sudo cat $DIR/my.cnf
33.        sudo /etc/init.d/mysql restart
34.        SHELL
35.      End
36.
37.    # MYSQL CLIENTS
38.    (1..3).each do |i|
39.      config.vm.define "Client#{i}" do |win|
40.        config.vm.provider "hyperv" do |h|
41.          h.memory = 1024000
42.          h.cpus = 1
43.        end
44.        win.vm.box = "gusztavvargadr/windows-10"
45.        win.vm.provision "shell", inline:<<-SHELL
46.          choco install mysql-client
47.        SHELL
48.      end
49.    end
50.  end
```

#### 2.4.3. Up all VM with one vagrant command.

I could't do that because my laptop with windows has only 2 Gb RAM

#### 2.4.4. Connect via vagrant ssh to server

#### 2.4.5. Connect via "winrm" to client machines and check the connection of clients to the MySQL server

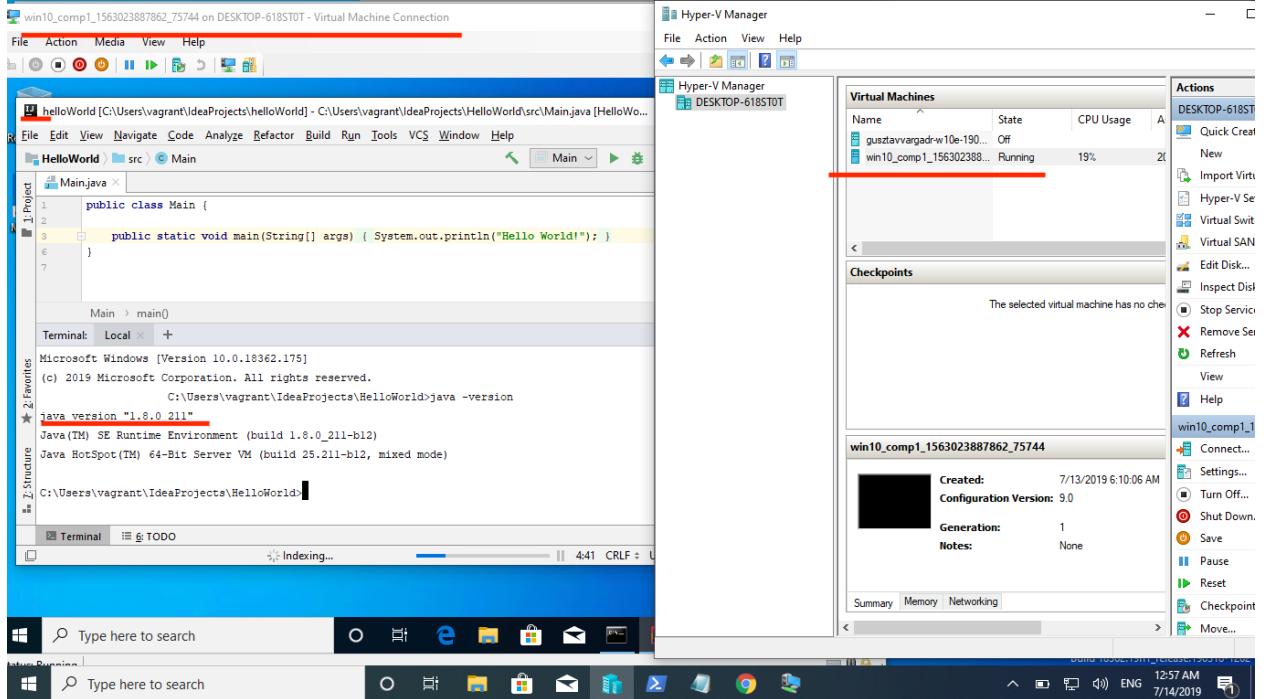
#### 2.4.6. For all client machines using vagrant, install the JDK.

#### 2.4.7. After installation and configuration, view the java version.

### 2.5. Create your own VagrantBox based on Windows 10 OS

#### 2.5.1. Add IntelliJ IDEA to one of the Windows client machines created in section 2.4

Screenshot# 2.2. VM Windows 10 with IntelliJ and Java installed.



#### 2.5.2. Based on this VM, make your own VagrantBox

For creating base box with Hyperv follow the instructions here

<https://www.vagrantup.com/docs/hyperv/boxes.html> :

1. The Hyper-V Virtual Machine Management service must be running  
<https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/best-practices-analyzer/the-hyper-v-virtual-machine-management-service-must-be-running>
2. Export VM from HyperV manager to specified path
3. Add *metadata.json* with at least provider specified:

```
{
  "provider": "hyperv"
}
```

3. Archive folders VirtualHardDisk, Virtual Machines, metadata.json

```
tar cvzf custom.box ./*
```

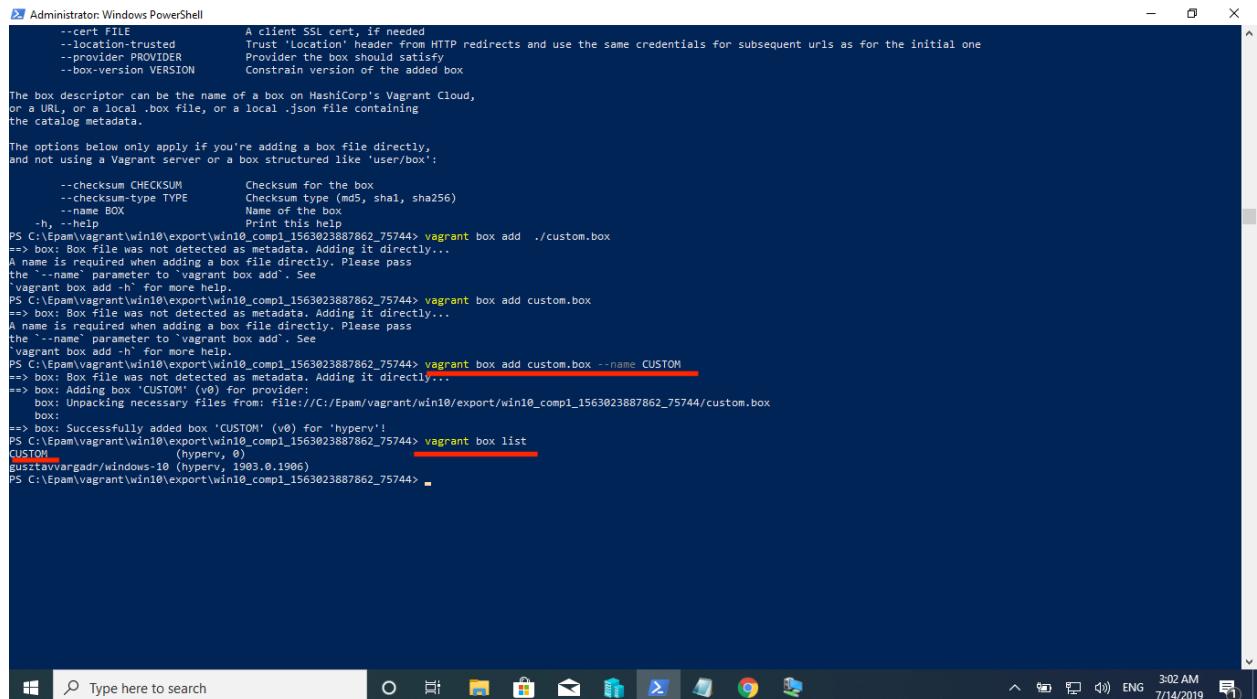
4.Add base box to Vagrant, run from current directory:

```
vagrant box add custom.box --name CUSTOM
```

5. Upload *custom.box* file to Vagrant cloud :

<https://app.vagrantup.com/dennis00010011b/boxes/windows10-enterprise-idk-intellij>

Screenshot#2.3.. Added base box



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered is "vagrant box add custom.box --name CUSTOM". The output shows the process of adding the box, including checksum verification and provider detection. The command "vagrant box list" is also shown at the end, confirming the addition of the "CUSTOM" box.

```
--cert FILE          A client SSL cert, if needed
--location-trusted   Trust 'Location' header from HTTP redirects and use the same credentials for subsequent urls as for the initial one
--provider PROVIDER  Provider the box should satisfy
--box-version VERSION Constrain version of the added box

The box descriptor can be the name of a box on HashiCorp's Vagrant Cloud,
or a URL, or a local .box file, or a local .json file containing
the catalog metadata.

The options below only apply if you're adding a box file directly,
and not using a Vagrant server or a box structured like 'user/box':

--checksum CHECKSUM      Checksum for the box
--checksum-type TYPE     Checksum type (md5, sha1, sha256)
--name BOX                Name of the box
-h, --help                Print this help

PS C:\Epam\vagrant\win10\export\win10_compl_1563023887862_75744> vagrant box add ./custom.box
=> Box 'custom' was not detected as metadata. Adding it directly...
A name is required when adding a box file directly. Please pass
the '--name' parameter to 'vagrant box add'. See
'vegant box add -h' for more help.
PS C:\Epam\vagrant\win10\export\win10_compl_1563023887862_75744> vagrant box add custom.box
=> box: Box file was not detected as metadata. Adding it directly...
A name is required when adding a box file directly. Please pass
the '--name' parameter to 'vagrant box add'. See
'vegant box add -h' for more help.
PS C:\Epam\vagrant\win10\export\win10_compl_1563023887862_75744> vagrant box add custom.box --name CUSTOM
=> box: Box file was not detected as metadata. Adding it directly...
=> box: Adding box 'CUSTOM' (<0>) for provider:
box: Unpacking necessary files from: file:///C:/Epam/vagrant/win10/export/win10_compl_1563023887862_75744/custom.box
=> box: Successfully added box 'CUSTOM' (<0>) for 'hyperv'!
PS C:\Epam\vagrant\win10\export\win10_compl_1563023887862_75744> vagrant box list
CUSTOM
  (hyperv, 0)
gustavavargadr/windows-10 (hyperv, 1903.0.1906)
PS C:\Epam\vagrant\win10\export\win10_compl_1563023887862_75744>
```

### 2.5.3 Destroy all Windows VMs with vagrant

```
vagrant destroy comp1      //remove VM
vagrant destroy comp2      //remove VM
vagrant destroy comp3      //remove VM
```

#### **2.5.4. Create a Vagrantfile to install three VMs with a configured LAN based on the OS created in section 2.5.2 and add a connection to the MySQL Server on the VM created in section 2.4.1**

```
Vagrant.configure("2") do |config|
  (1..3).each do |i|
    config.vm.define "comp#{i}" do |win|
      win.vm.box = "CUSTOM"
      #win.vm.boot_timeout=600

      config.vm.provider "hyperv" do |h|
        h.vmname = "comp1_win10"
        h.memory = 2100000
        h.cpus = 1
      end

      end
    end
  end
```

**Notice.** Looks like Vagrant does not support networking for HyperV

<https://www.vagrantup.com/docs/hyperv/limitations.html>

A result of this is that networking configurations in the Vagrantfile are completely ignored with Hyper-V. Vagrant cannot enforce a static IP or automatically configure a NAT.

#### **2.5.5. Up all VMs and check connection to MySQL Server from VMs with Windows OS**

I couldn't do that because my laptop with windows has only 2 Gb RAM

- 3. Create a report with screenshots and attach script files demonstrating the solution of the tasks**

SEE ABOVE