**Dennis Tikhomirov. DevOps Spring’19.**

**Homework Task #4 (Vagrant).**

**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"

end

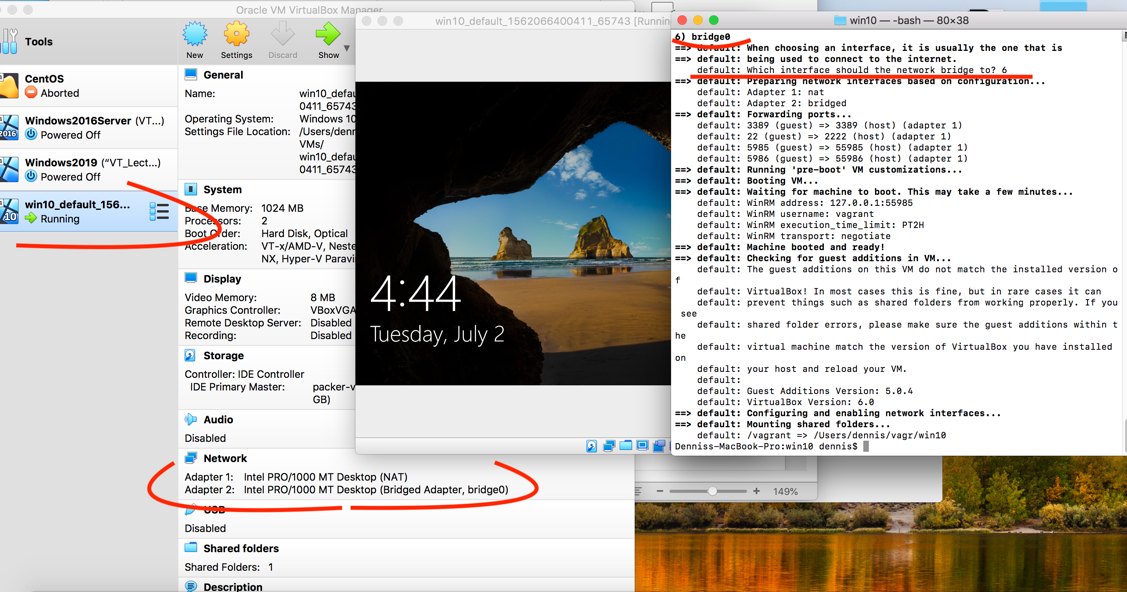
end

Commands

vagrant init senglin/win-10-enterprise-vs2015community \ --box-version 1.0.0

vagrant up

Screenshot #1. Installed Win10 on VirtualBox



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"

1. *Run command*

Vagrant up

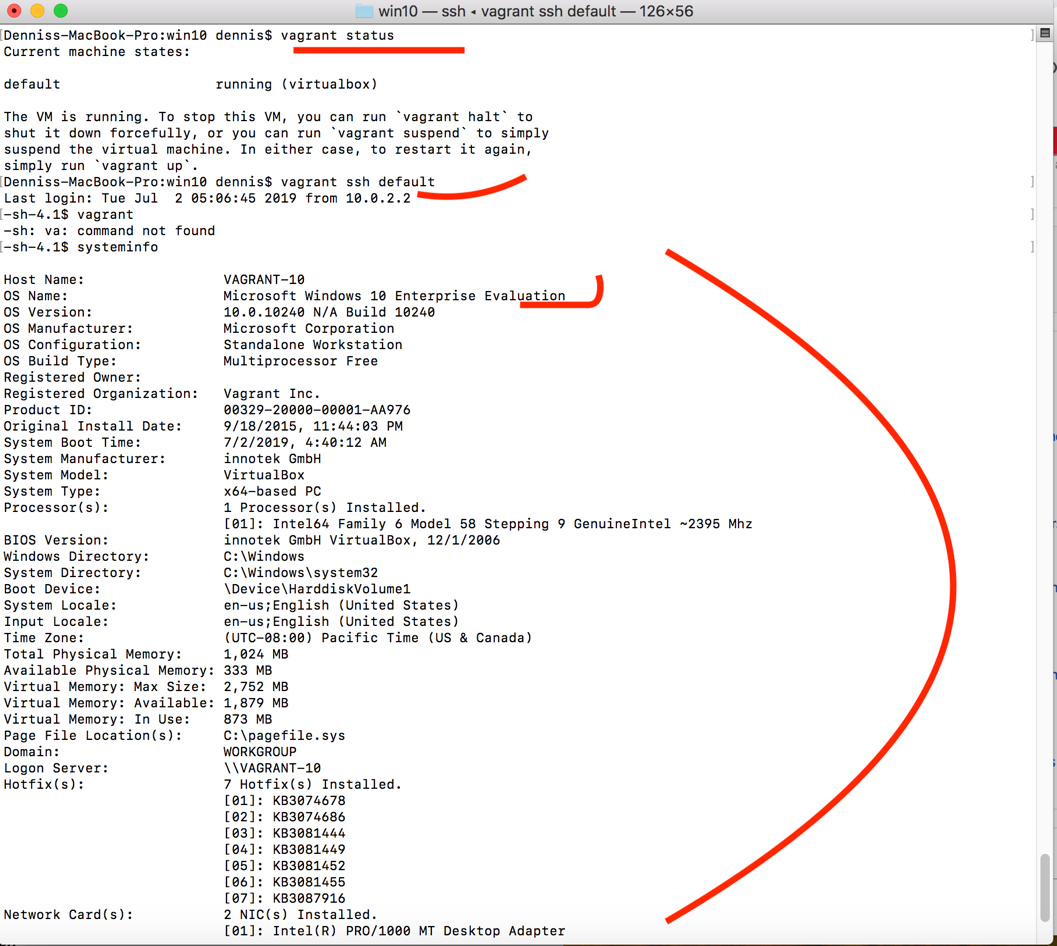
**1.3.2. Connect to the box via vagrant ssh**

vagrant status // show VMs

vagrant ssh default // connect to VM [default]

-sh-4.1$ systeminfo // Windows command

Screenshot #2. Connect to VirtualBox VM trough Vagrant



**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**

**msiexec /i mysql-installer-web-community-8.0.16.0.msi /passive**

**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**

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

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

**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

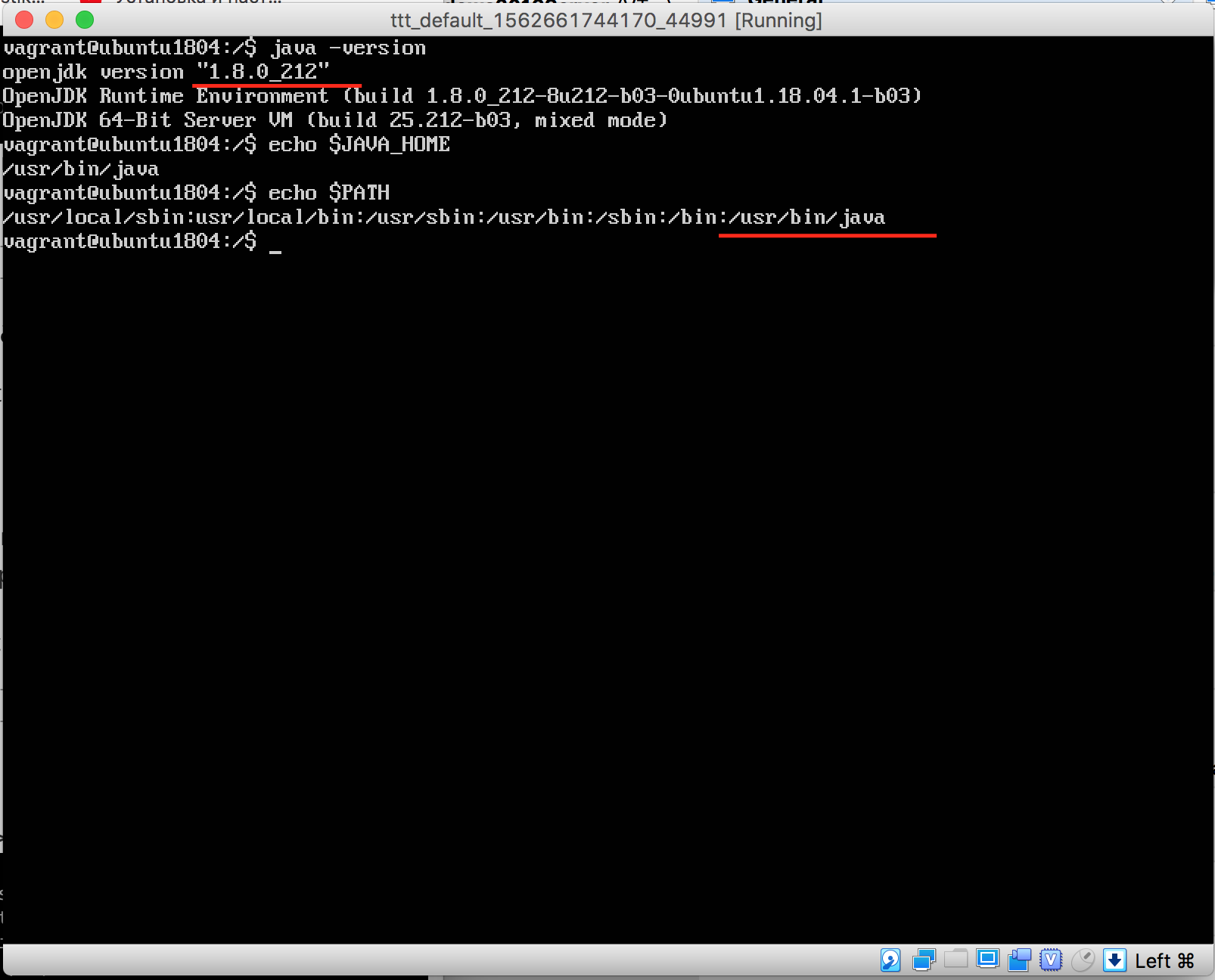
sudo apt-get install -y openjdk-8-jdk

whereis java

JAVA\_HOME=/usr/bin/java

export PATH=$PATH:$JAVA\_HOME

Screenshot #3. Java version, $JAVA\_HOME , $PATH values



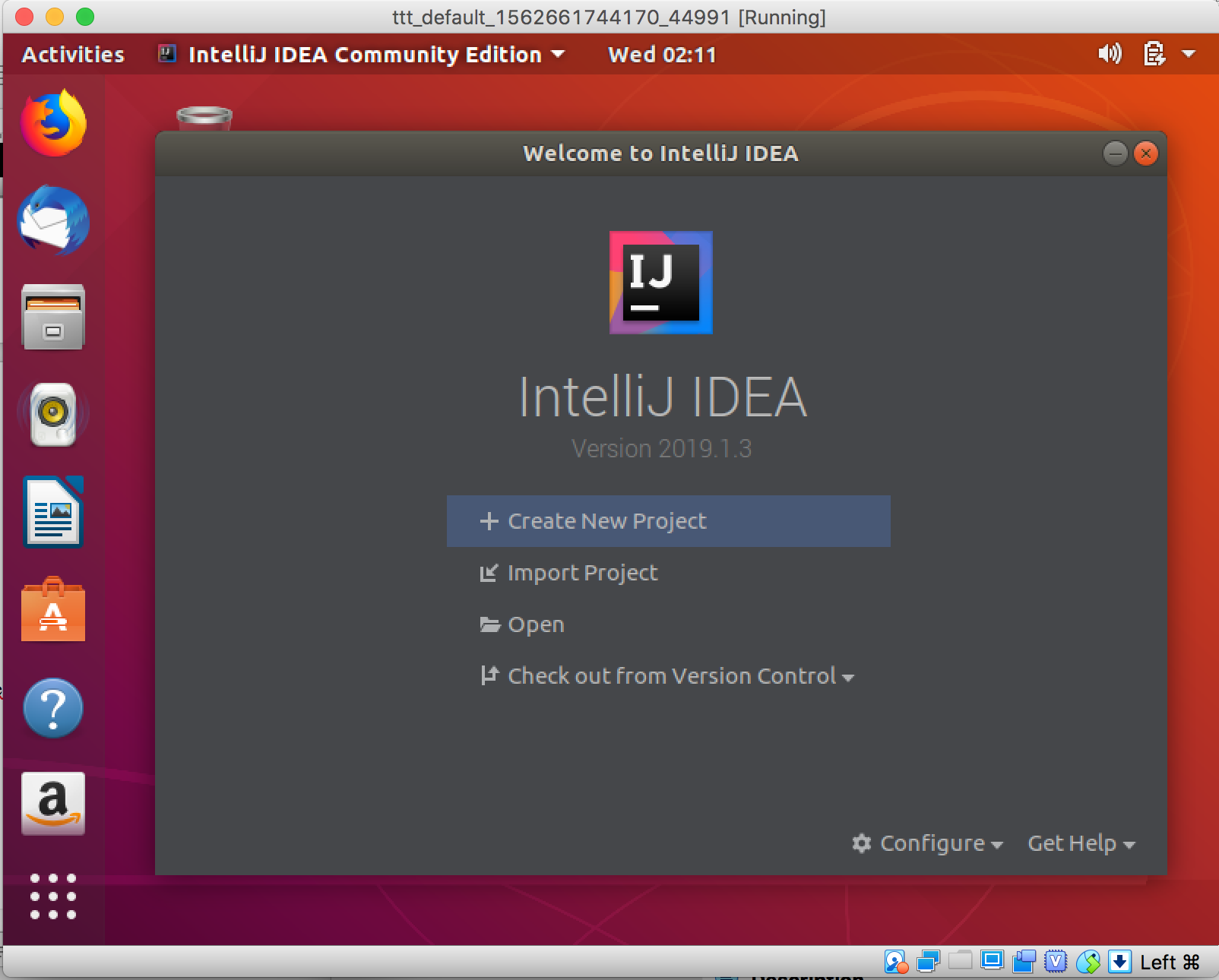
**1.5.3. Install Intellij IDEA on this VM**

sudo snap install intellij-idea-community –-classic

Upgrade Ubuntu to Desktop version (add (GUI)

sudo apt-get install ubuntu-desktop

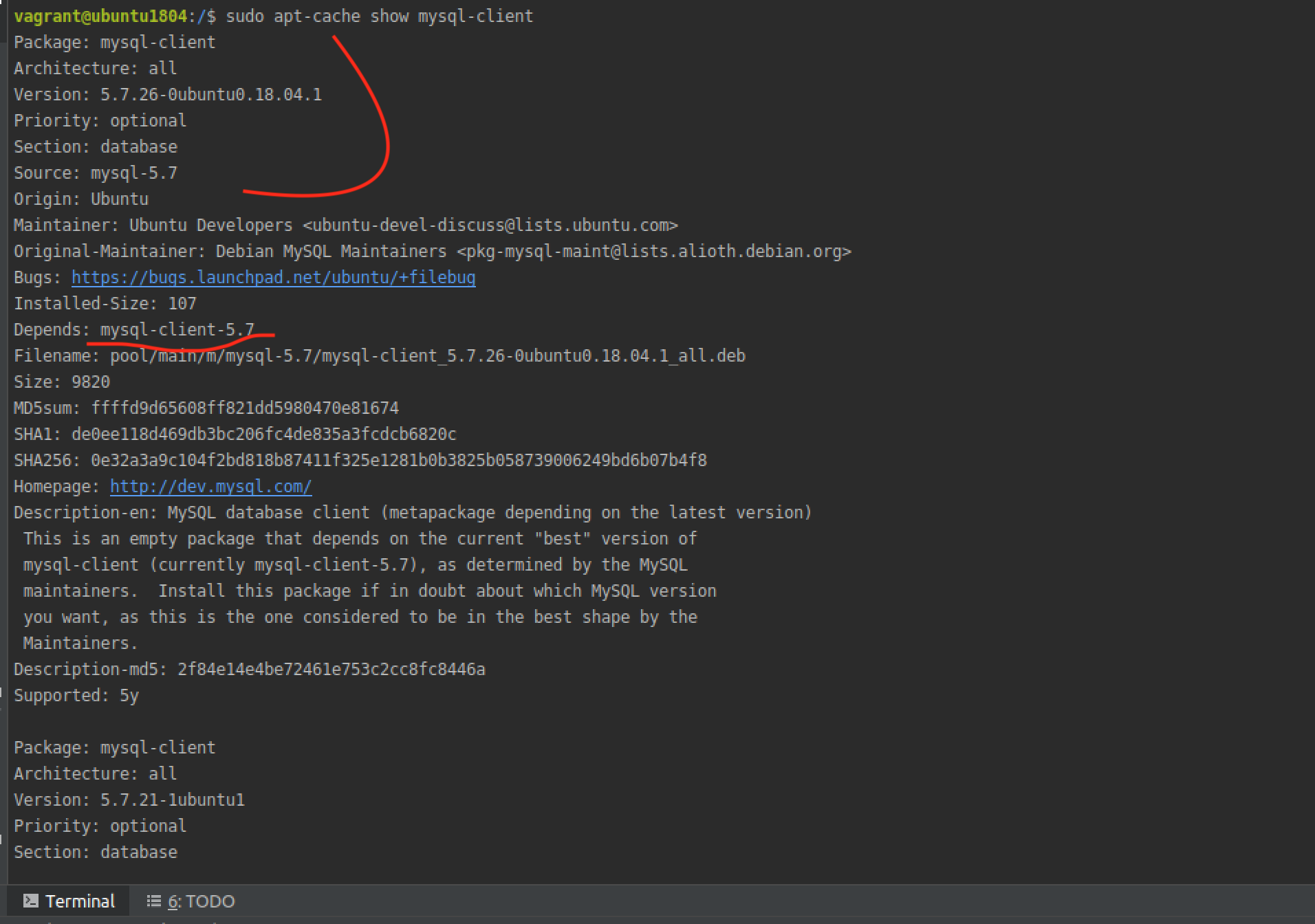
Screenshot #4. Installed Intellij Idea



**1.5.4. Install MySQL Client**

sudo apt-get install mysql-client

Screenshot #5. Installed mysql-client



**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>

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

3. Network adapter#1 must be NAT. Also remember MAC address:

config.vm.base\_mac = "080027AE3BAC"

4.Run command:

Vagrant package –base ubuntu1804

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

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

**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 upload/download

#package.file 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"

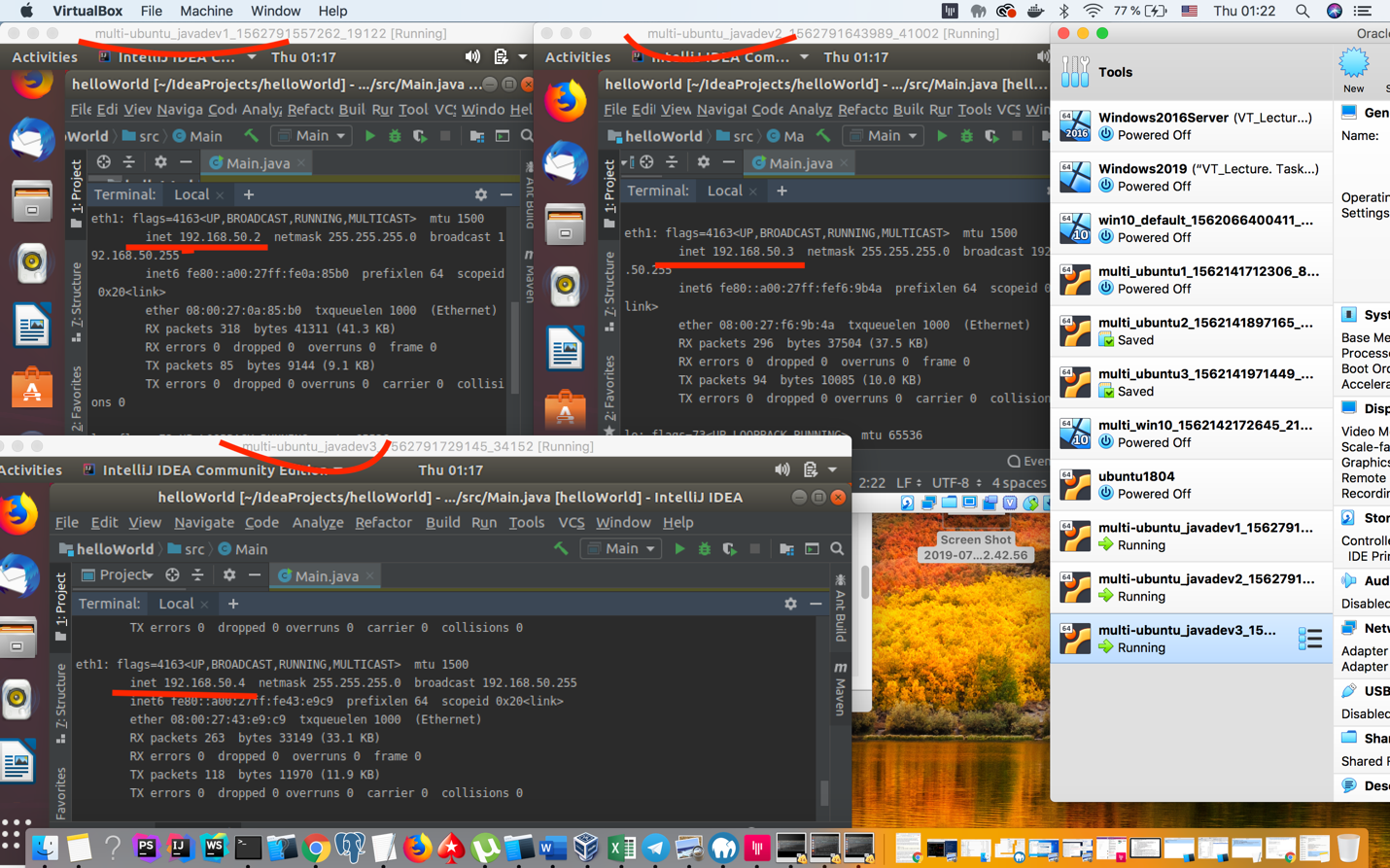
ubuntu.vm.network "private\_network", ip: "192.168.50.#{i+1}"

end

end

end

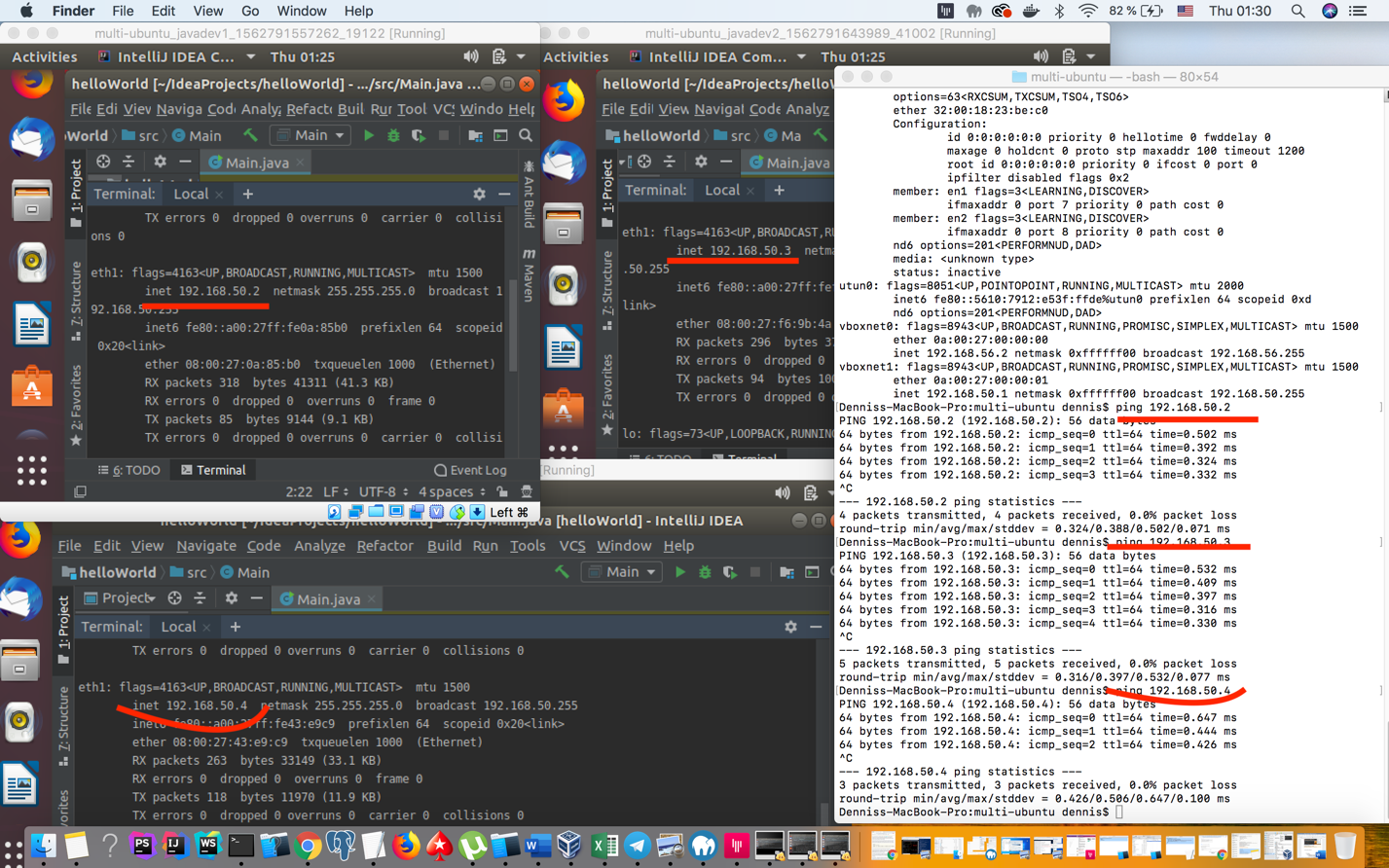
Screenshot #5. Three VMs from own Vagrant box

****

**1.5.8. Check LAN connections**

LAN connection was already configurated in Vagrantfile

ubuntu.vm.network "private\_network", ip: "192.168.50.#{i+1}"

****

**1.5.9. Check connection to MySQL server**

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

2.3.2. Connect to the box via vagrant ssh

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.

2.4.3. Up all VM with one vagrant command.

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

2.5.2. Based on this VM, make your own VagrantBox

2.5.3. Destroy all Windows VMs with vagrant

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

2.5.5. Up all VMs and check connection to MySQL Server from VMs with Windows OS 3. Create a report with screenshots and attach script files demonstrating the solution of the tasks