# Homework M1: DevOps and Containerization

## Virtualization

### Create Vagrant Box with Debian 12

I will use debian-12.11-x64.ova downloaded from <https://zahariev.pro/go/devops-vm-templates>.

The scripts db.sh and web.sh were edited and adjusted for Debian. Check 01-virtualization folder.

#### Import and setup the VM

VBoxManage import debian-12.11-x64.ova --vsys 0 --vmname debian12-vagrant

VBoxManage modifyvm debian12-vagrant --cpus 1

VBoxManage modifyvm debian12-vagrant --memory 2048

VBoxManage modifyvm debian12-vagrant --audio-driver none

VBoxManage modifyvm debian12-vagrant --nic1 nat

VBoxManage modifyvm debian12-vagrant --natpf1 "SSH,tcp,,2222,,22"

#### Start the VM in headless mode and wait for it to boot before ssh.

VBoxManage startvm debian12-vagrant --type=headless

A screenshot of a computer

AI-generated content may be incorrect.

#### Prepare the Debian VM

##### Update and install packages

sudo apt update && sudo apt upgrade

sudo apt-get -qq -y install \

vim \

curl \

ca-certificates \

git \

wget \

openssh-server

##### Setup OpenSSH

sudo sed -i '/.\*PasswordAuthentication.\*/d' /etc/ssh/sshd\_config

echo 'PasswordAuthentication yes' | sudo tee -a /etc/ssh/sshd\_config

sudo sed -i '/.\*UseDNS.\*/d' /etc/ssh/sshd\_config

echo 'UseDNS no' | sudo tee -a /etc/ssh/sshd\_config

sudo service sshd restart

sudo service sshd status

##### Add vagrant user to sudoers

echo "vagrant ALL=(ALL) NOPASSWD:ALL" | sudo tee -a /etc/sudoers.d/vagrant

sudo chmod 0440 /etc/sudoers.d/vagrant

##### Add Vagrant public key

mkdir -p /home/vagrant/.ssh

curl -fsSL https://raw.githubusercontent.com/hashicorp/vagrant/master/keys/vagrant.pub -o /home/vagrant/.ssh/authorized\_keys

chmod 600 /home/vagrant/.ssh/authorized\_keys

chown -R vagrant:vagrant /home/vagrant/.ssh

##### Install VBoxGuestAdditions.

###### Install deps necessary to compile kernel module

sudo apt-get update

sudo apt-get install -y build-essential dkms bzip2 tar linux-headers-"$(uname -r)"

###### Attach the media

VBoxManage storageattach debian12-vagrant --storagectl "IDE" --port 1 --device 0 --type dvddrive --medium additions

###### Verify attachment

VBoxManage showvminfo "debian12-vagrant"

###### Install guest additions

sudo mkdir -p /tmp/vbox;

sudo mount /dev/cdrom /tmp/vbox

sudo bash /tmp/vbox/VBoxLinuxAdditions.run

###### Add the vagrant user to the vboxsf group

usermod -aG vboxsf vagrant

###### Unmount

sudo umount /mnt;

rm -rf /tmp/vbox;

rm -f "$HOME\_DIR"/\*.iso;

###### Remove kernel dev packages and compilers we no longer need

sudo apt-get remove -y build-essential gcc g++ make libc6-dev dkms linux-headers-"$(uname -r)"

###### Reboot

Don’t forget to unmount the VBoxGuestAdditions.iso media.

sudo reboot

###### Verify

lsmod | grep vbox

##### hushlogin

echo -n > /home/vagrant/.hushlogin

##### Configure GRUB

sudo sed -i 's/^GRUB\_TIMEOUT=.\*/GRUB\_TIMEOUT=0/; s/^GRUB\_TIMEOUT\_STYLE=.\*/GRUB\_TIMEOUT\_STYLE=hidden/' /etc/default/grub && sudo update-grub

##### Cleanup

sudo apt-get -y autoremove;

sudo apt-get -y clean;

export HISTSIZE=0

##### Minimize

count=$(df --sync -kP / | tail -n1 | awk -F ' ' '{print $4}')

count=$((count - 1))

sudo dd if=/dev/zero of=/tmp/whitespace bs=1M count=$count || echo "dd exit code $? is suppressed";

sudo rm /tmp/whitespace

#### Switch off the VM

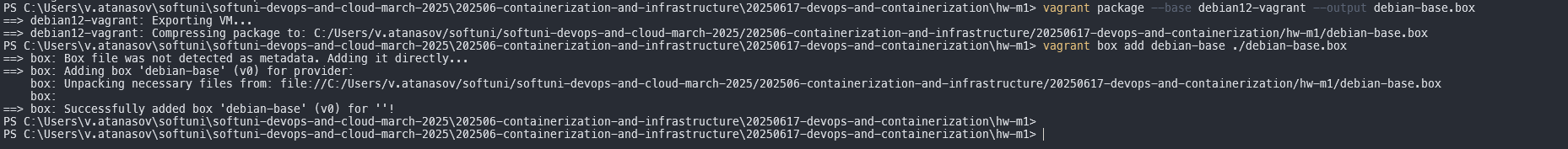
VBoxManage controlvm debian12-vagrant poweroff

#### Create Vagrant box

vagrant package --base debian12-vagrant --output debian-base.box

#### Add the box to the local vagrant repository

vagrant box add debian-base ./debian-base.box



### Bgapp Implementation with Custom Vagrant Box

Brows the 01-virualization folder to explore the Vagrantfile.

PS C:\Users\v.atanasov\softuni\softuni-devops-and-cloud-march-2025\202506-containerization-and-infrastructure\20250617-devops-and-containerization\hw-m1> vagrant up

Bringing machine 'web' up with 'virtualbox' provider...

Bringing machine 'db' up with 'virtualbox' provider...

==> web: Importing base box 'debian-base'...

==> web: Matching MAC address for NAT networking...

==> web: Setting the name of the VM: hw-m1\_web\_1749716763871\_27895

==> web: Clearing any previously set network interfaces...

==> web: Preparing network interfaces based on configuration...

web: Adapter 1: nat

web: Adapter 2: hostonly

==> web: Forwarding ports...

web: 80 (guest) => 8080 (host) (adapter 1)

web: 22 (guest) => 2222 (host) (adapter 1)

==> web: Booting VM...

==> web: Waiting for machine to boot. This may take a few minutes...

web: SSH address: 127.0.0.1:2222

web: SSH username: vagrant

web: SSH auth method: private key

==> web: Machine booted and ready!

==> web: Checking for guest additions in VM...

==> web: Setting hostname...

==> web: Configuring and enabling network interfaces...

==> web: Running provisioner: shell...

web: Running: C:/Users/VDB51~1.ATA/AppData/Local/Temp/vagrant-shell20250612-759520-wo88s0.sh

web: \* Add hosts ...

web: 192.168.89.100 web.do1.lab web

web: 192.168.89.101 db.do1.lab db

web: \* Update and install software ...

...............................

web: \* Enable and start Apache ...

web: Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.

web: Executing: /lib/systemd/systemd-sysv-install enable apache2

web: \* Cloning frontend ...

web: Cloning into 'bgapp'...

web: \* Clean up /var/www/html/ ...

web: removed '/var/www/html/index.html'

web: \* Copy web site files to /var/www/html/ ...

web: total 32K

web: drwxr-xr-x 2 root root 4.0K Jun 12 11:28 .

web: drwxr-xr-x 3 root root 4.0K Jun 12 11:27 ..

web: -rw-r--r-- 1 root root 13K Jun 12 11:28 bulgaria-map.png

web: -rw-r--r-- 1 root root 105 Jun 12 11:28 config.php

web: -rw-r--r-- 1 root root 1.9K Jun 12 11:28 index.php

==> db: Importing base box 'debian-base'...

==> db: Matching MAC address for NAT networking...

==> db: Setting the name of the VM: hw-m1\_db\_1749716902477\_51810

==> db: Fixed port collision for 22 => 2222. Now on port 2200.

==> db: Clearing any previously set network interfaces...

==> db: Preparing network interfaces based on configuration...

db: Adapter 1: nat

db: Adapter 2: hostonly

==> db: Forwarding ports...

db: 22 (guest) => 2200 (host) (adapter 1)

==> db: Booting VM...

==> db: Waiting for machine to boot. This may take a few minutes...

db: SSH address: 127.0.0.1:2200

db: SSH username: vagrant

db: SSH auth method: private key

==> db: Machine booted and ready!

==> db: Checking for guest additions in VM...

==> db: Setting hostname...

==> db: Configuring and enabling network interfaces...

==> db: Running provisioner: shell...

db: Running: C:/Users/VDB51~1.ATA/AppData/Local/Temp/vagrant-shell20250612-759520-c28o56.sh

db: \* Add hosts ...

db: 192.168.89.100 web.do1.lab web

db: 192.168.89.101 db.do1.lab db

db: \* Update and install software ...

...............................

db: Updating mariadb configs in /etc/mysql/mariadb.conf.d/50-server.cnf

db: Updated mariadb bind address in /etc/mysql/mariadb.conf.d/50-server.cnf to 0.0.0.0 to allow external connections.

db: Restarting mariadb ...

db: \* Cloning backend ...

db: Cloning into 'bgapp'...

db: \* Create and load the database ...

A map of the country with a red and green flag

AI-generated content may be incorrect.

## Containerization

Brows 02-containerization folder to explore the Vagrantfile. In the folder are placed bash scripts for installing and configuring docker and docker tools.

### Run a container named **homework** based on the image **shekeriev/animal-stories** interactively

docker run -it --name homework shekeriev/animal-stories

### Find the file **animal-stories.txt** that is inside the container

find -type f -name animal-stories.txt

### Find all the rows about **tigers**

find -type f -name animal-stories.txt -exec grep -i tigers {} \;

### List all the **unique colors** sorted in reverse (descending) order

find -type f -name animal-stories.txt -exec sh -c 'cut -d- -f1 "{}" | sort -r | uniq' \;

A screen shot of a computer

AI-generated content may be incorrect.