Project-2 Automated Setup of Multi Tier App Locally

Monday, June 12, 2023 2:29 PM

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.hostmanager.enabled = true
  config.hostmanager.manage_host = true

### DB vm  ####
  config.vm.define "db01" do |db01|
    db01.vm.box = "geerlingguy/centos7"
    db01.vm.hostname = "db01"
    db01.vm.network "private_network", ip: "192.168.56.15"
    db01.vm.provision "shell", path: "mysql.sh"
  end

### Memcache vm  ####
  config.vm.define "mc01" do |mc01|
    mc01.vm.box = "geerlingguy/centos7"
    mc01.vm.hostname = "mc01"
    mc01.vm.network "private_network", ip: "192.168.56.14"
    mc01.vm.provision "shell", path: "memcache.sh"
  end

### RabbitMQ vm  ####
  config.vm.define "rmq01" do |rmq01|
    rmq01.vm.box = "geerlingguy/centos7"
  rmq01.vm.hostname = "rmq01"
    rmq01.vm.network "private_network", ip: "192.168.56.16"
    rmq01.vm.provision "shell", path: "rabbitmq.sh"
  end

### tomcat vm ###
   config.vm.define "app01" do |app01|
    app01.vm.box = "geerlingguy/centos7"
    app01.vm.hostname = "app01"
    app01.vm.network "private_network", ip: "192.168.56.12"
    app01.vm.provision "shell", path: "tomcat.sh"
    app01.vm.provider "virtualbox" do |vb|
     vb.memory = "1024"
   end
   end


### Nginx VM ###
  config.vm.define "web01" do |web01|
    web01.vm.box = "ubuntu/xenial64"
    web01.vm.hostname = "web01"
  web01.vm.network "private_network", ip: "192.168.56.11"
  web01.vm.provision "shell", path: "nginx.sh"
end

end
```

****

1. The first line Vagrant.configure("2") do |config| initializes the Vagrant configuration and sets the version to "2". This ensures compatibility with the Vagrant 2.x syntax.
2. config.hostmanager.enabled = true enables the Vagrant Hostmanager plugin, which automatically manages the host's /etc/hosts file to resolve hostnames of the virtual machines.
3. config.hostmanager.manage_host = true indicates that Vagrant should manage the host entries for the virtual machines defined in the Vagrantfile.
4. The next section defines a virtual machine called "db01". It uses the "geerlingguy/centos7" box, which is a CentOS 7 image provided by the "geerlingguy" box provider. The virtual machine's hostname is set to "db01".
5. The line db01.vm.network "private_network", ip: "192.168.56.15" creates a private network interface within the virtual machine and assigns the IP address "192.168.56.15" to it.
6. The line db01.vm.provision "shell", path: "mysql.sh" specifies that a shell script called "mysql.sh" should be executed during the provisioning process of the "db01" virtual machine. This script likely contains commands to install and configure MySQL.
7. The same steps (4-6) are repeated for the "mc01" virtual machine, which represents a Memcache server.
8. Similarly, the steps (4-6) are repeated for the "rmq01" virtual machine, representing a RabbitMQ server.
9. The next section defines a virtual machine called "app01". It also uses the "geerlingguy/centos7" box, and the hostname is set to "app01".
10. The line app01.vm.network "private_network", ip: "192.168.56.12" assigns the IP address "192.168.56.12" to the private network interface of the "app01" virtual machine.
11. The line app01.vm.provision "shell", path: "tomcat.sh" specifies that a shell script called "tomcat.sh" should be executed during the provisioning process of the "app01" virtual machine. This script likely contains commands to install and configure Apache

Tomcat.

12. The nested block app01.vm.provider "virtualbox" do |vb| allows for provider-specific configuration. In this case, it configures the VirtualBox provider and sets the memory allocation for the "app01" virtual machine to "1024" MB.
13. The same steps (9-12) are repeated for the "web01" virtual machine, which represents an Nginx server. The box used is "ubuntu/xenial64", an Ubuntu Xenial (16.04) image.
14. Finally, the end statement concludes the Vagrant configuration block.

In summary, this Vagrantfile sets up multiple virtual machines with different purposes using CentOS and Ubuntu images. Each virtual machine is assigned a private IP address, and provisioning shell scripts are executed to install and configure the necessary software on each machine. The Vagrant Hostmanager plugin is enabled to manage the host entries for the virtual machines.

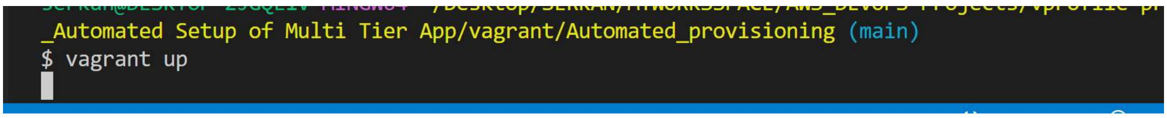
***

README.MD

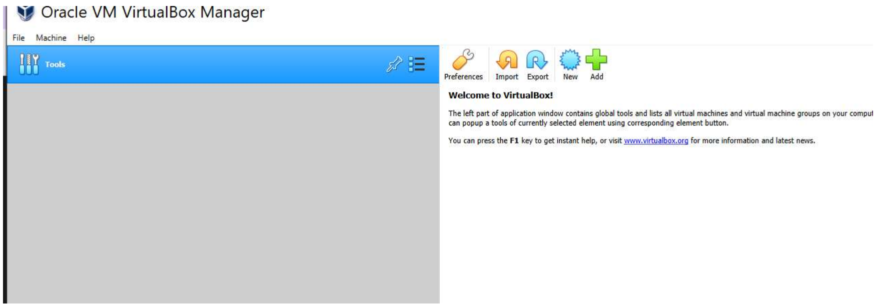
~~~
# Project-2: Vprofile Project: Automated Setup of Multi Tier App, Locally
[*Project Source*](https://www.udemy.com/course/devopsprojects/?src=sac&kw=devops+projects)
![](images/vprofile-project.png)
## Prerequisites
 * Oracle VM VirtualBox Manager
 * Vagrant
 * Vagrant plugins
 * Git
 * IDE (SublimeText, VSCode, etc)
## Step1: Preparing Bash Scripts for VMs
### Bash Script for DB
- In Project-1, we have setup our 3-Tier Application manually. This time we will create bash scripts to automate our VM creation/provisioning through Vagrantfile.
- First we will create `mysql.sh` file for our database.
```sh
#!/bin/bash
DATABASE_PASS='admin123'
sudo yum update -y
sudo yum install epel-release -y
sudo yum install git zip unzip -y
sudo yum install mariadb-server -y

# starting & enabling mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb
cd /tmp/
git clone -b local-setup https://github.com/devopshydclub/vprofile-project.git
#restore the dump file for the application
sudo mysqladmin -u root password "$DATABASE_PASS"
sudo mysql -u root -p"$DATABASE_PASS" -e "UPDATE mysql.user SET Password=PASSWORD('$DATABASE_PASS') WHERE User='root'"
sudo mysql -u root -p"$DATABASE_PASS" -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1', '::1')"
sudo mysql -u root -p"$DATABASE_PASS" -e "DELETE FROM mysql.user WHERE User=''"
sudo mysql -u root -p"$DATABASE_PASS" -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%'"
sudo mysql -u root -p"$DATABASE_PASS" -e "FLUSH PRIVILEGES"
sudo mysql -u root -p"$DATABASE_PASS" -e "create database accounts"
sudo mysql -u root -p"$DATABASE_PASS" -e "grant all privileges on accounts.* TO 'admin'@'localhost' identified by 'admin123'"
sudo mysql -u root -p"$DATABASE_PASS" -e "grant all privileges on accounts.* TO 'admin'@'%' identified by 'admin123'"
sudo mysql -u root -p"$DATABASE_PASS" accounts < /tmp/vprofile-project/src/main/resources/db_backup.sql
sudo mysql -u root -p"$DATABASE_PASS" -e "FLUSH PRIVILEGES"
# Restart mariadb-server
sudo systemctl restart mariadb

#starting the firewall and allowing the mariadb to access from port no. 3306
sudo systemctl start firewalld
sudo systemctl enable firewalld
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
sudo firewall-cmd --reload
sudo systemctl restart mariadb
```
### Bash Script for Memcached
- Next we will create a bash script to provision our memcached server.
```sh
#!/bin/bash
sudo yum install epel-release -y
sudo yum install memcached -y
sudo systemctl start memcached
sudo systemctl enable memcached
sudo systemctl status memcached
sudo memcached -p 11211 -U 11111 -u memcached -d
~~~

~~~
```
### Bash Script for RabbitMQ
- This time we will create a bash script for RabbitMQ.
```sh
#!/bin/bash
sudo yum install epel-release -y
sudo yum update -y
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
curl -s https://packagecloud.io/install/repositories/rabbitmq/rabbitmq-
server/script.rpm.sh | sudo bash
sudo yum install rabbitmq-server -y
sudo systemctl start rabbitmq-server
sudo systemctl enable rabbitmq-server
sudo systemctl status rabbitmq-server
sudo sh -c 'echo "[{rabbit, [{loopback_users, []}]}]." >
/etc/rabbitmq/rabbitmq.config'
sudo rabbitmqctl add_user test test
sudo rabbitmqctl set_user_tags test administrator
sudo systemctl restart rabbitmq-server
```
### Bash Script for Application
- We will create a Bash script to provision Tomcat server for our application.
```sh
TOMURL="https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.37/bin/apache-
tomcat-8.5.37.tar.gz"
yum install java-1.8.0-openjdk -y
yum install git maven wget -y
cd /tmp/
wget $TOMURL -O tomcatbin.tar.gz
EXTOUT=`tar xzvf tomcatbin.tar.gz`
TOMDIR=`echo $EXTOUT | cut -d '/' -f1`
useradd --shell /sbin/nologin tomcat
rsync -avzh /tmp/$TOMDIR/ /usr/local/tomcat8/
chown -R tomcat.tomcat /usr/local/tomcat8
rm -rf /etc/systemd/system/tomcat.service
cat <<EOT>> /etc/systemd/system/tomcat.service
[Unit]
Description=Tomcat
After=network.target
[Service]
User=tomcat
Group=tomcat
WorkingDirectory=/usr/local/tomcat8
#Environment=JRE_HOME=/usr/lib/jvm/jre
Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment=CATALINA_PID=/var/tomcat/%i/run/tomcat.pid
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

RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
EOT
systemctl daemon-reload
systemctl start tomcat
systemctl enable tomcat
git clone -b local-setup https://github.com/devopshydclub/vprofile-project.git
cd vprofile-project
mvn install
systemctl stop tomcat
sleep 60
rm -rf /usr/local/tomcat8/webapps/ROOT*
cp target/vprofile-v2.war /usr/local/tomcat8/webapps/ROOT.war
systemctl start tomcat
sleep 120
cp /vagrant/application.properties /usr/local/tomcat8/webapps/ROOT/WEB-
INF/classes/application.properties
systemctl restart tomcat
```
### Bash Script for Nginx server
- Lastly we will create a bash script to provision Nginx server which will
forward requests to our backend application.
```sh
# adding repository and installing nginx
apt update
apt install nginx -y
cat <<EOT > vproapp
upstream vproapp {
 server app01:8080;
}
server {
  listen 80;
location / {
  proxy_pass http://vproapp;
}
~~~

~~~
}
EOT
mv vproapp /etc/nginx/sites-available/vproapp
rm -rf /etc/nginx/sites-enabled/default
ln -s /etc/nginx/sites-available/vproapp /etc/nginx/sites-enabled/vproapp
#starting nginx service and firewall
systemctl start nginx
systemctl enable nginx
systemctl restart nginx
```
## Step2: Preparing Bash Scripts for VMs
- First clone the repository
```sh
git clone https://github.com/ser-2007/vprofile-project.git
```
- We need to go to directory that our Vagrantfile exists. Before we run our
VBoxes using `vagrant`, we need to install below plugin.
```sh
vagrant plugin install vagrant-hostmanager
```
- After plugin installed, we can run below command to setup our VMs which will
also bootstrap our servers for us.
```sh
vagrant up
```
- Our VMs are ready in VirtualBox.
![](images/vms-are-ready.png)
## Step3: Validate Application from Browser
- We can validate the application using hostname given in Vagrantfile. Go to
browser `http://web01`. Frontend is working successfully.
![](images/nginx-working.png)
- Backend services also up/running.
![](images/app-working.png)
- We can validate RabbitMq service.
![](images/rabbitmq-working.png)
- Next we can check our DB/Memcache services.
![](images/db-working.png)
![](images/cache-working.png)
- If we want to stop our VMs, we can use below command:
```sh
vagrant halt
```
- We can check status of our VMs with below command:
```sh
vagrant status
```
![](images/vagrant-halt-status.png)
- If we want to start again, we can easily run:
```sh
vagrant up
```
- Once we are done, we can destroy our VMs:
```sh
vagrant destroy
```
![](images/vagrant%20destroy.png)
~~~



```
_Automated Setup of Multi Tier App/vagrant/Automated_provisioning (main)
$ vagrant up
```

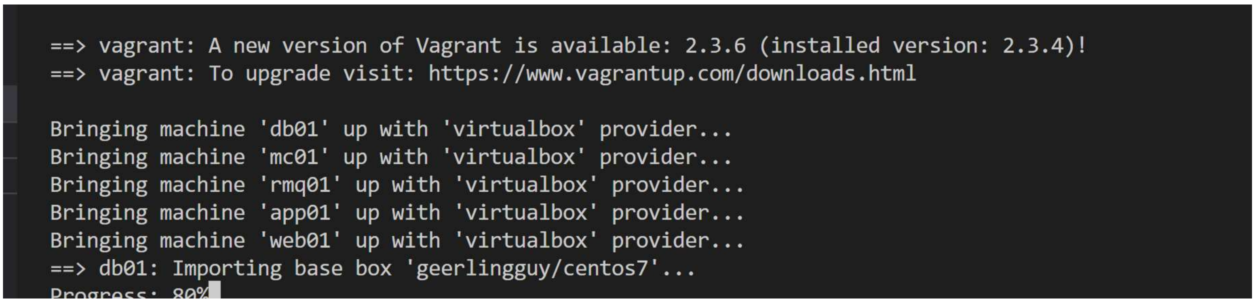


Oracle VM VirtualBox Manager

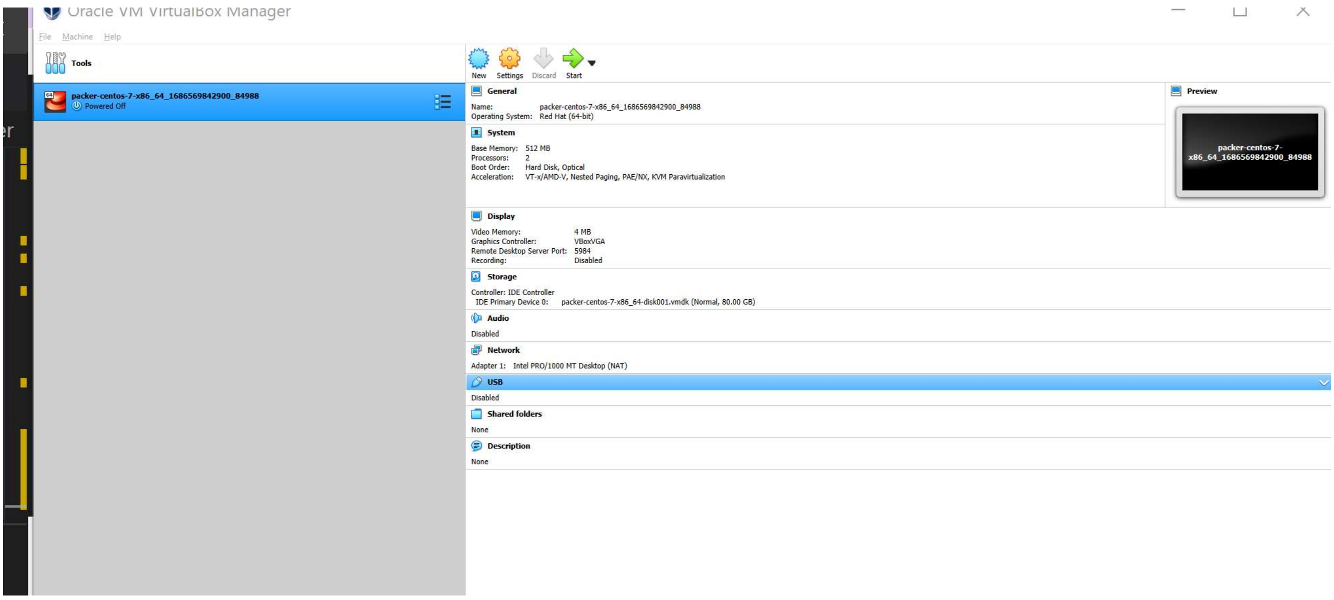
Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your comput
can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.



```
==> vagrant: A new version of Vagrant is available: 2.3.6 (installed version: 2.3.4)!
==> vagrant: To upgrade visit: https://www.vagrantup.com/downloads.html

Bringing machine 'db01' up with 'virtualbox' provider...
Bringing machine 'mc01' up with 'virtualbox' provider...
Bringing machine 'rmq01' up with 'virtualbox' provider...
Bringing machine 'app01' up with 'virtualbox' provider...
Bringing machine 'web01' up with 'virtualbox' provider...
==> db01: Importing base box 'geerlingguy/centos7'...
Progress: 80%
```


Oracle VM VirtualBox Manager
File
Machine
Help
Tools
New
Settings
Discard
Start
Preview
packer-centos-7-x86_64_1686569842900_84988

packer-centos-7-x86_64_1686569842900_84988
Powered Off

General
Name: packer-centos-7-x86_64_1686569842900_84988
Operating System: Red Hat (64-bit)

System
Base Memory: 512 MB
Processors: 2
Boot Order: Hard Disk, Optical
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display
Video Memory: 4 MB
Graphics Controller: VBoxVGA
Remote Desktop Server Port: 5984
Recording: Disabled

Storage
Controller: IDE Controller
IDE Primary Device 0: packer-centos-7-x86_64-disk001.vmdk (Normal, 80.00 GB)

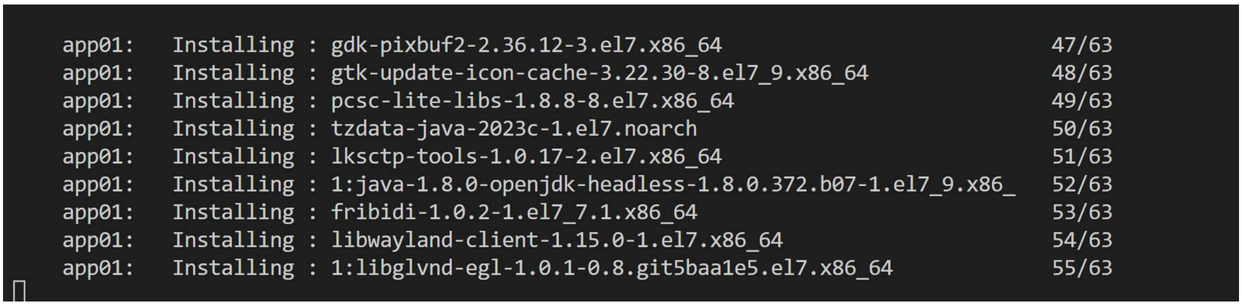
Audio
Disabled

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

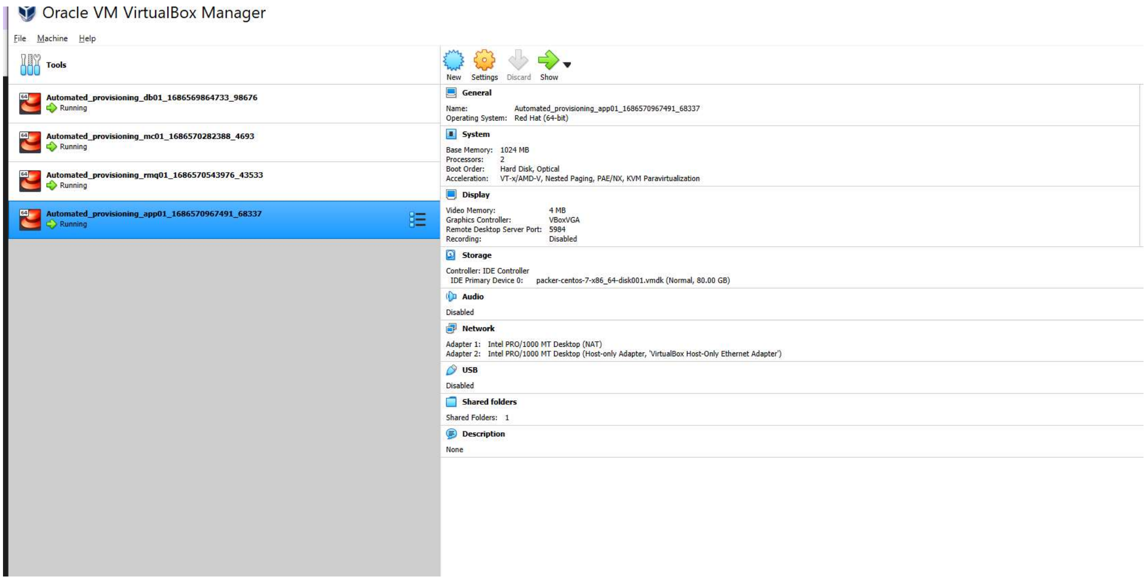
USB
Disabled

Shared folders
None

Description
None



```
app01:    Installing : gdk-pixbuf2-2.36.12-3.el7.x86_64                        47/63
app01:    Installing : gtk-update-icon-cache-3.22.30-8.el7_9.x86_64            48/63
app01:    Installing : pcsc-lite-libs-1.8.8-8.el7.x86_64                       49/63
app01:    Installing : tzdata-java-2023c-1.el7.noarch                          50/63
app01:    Installing : lksctp-tools-1.0.17-2.el7.x86_64                        51/63
app01:    Installing : 1:java-1.8.0-openjdk-headless-1.8.0.372.b07-1.el7_9.x86_ 52/63
app01:    Installing : fribidi-1.0.2-1.el7_7.1.x86_64                          53/63
app01:    Installing : libwayland-client-1.15.0-1.el7.x86_64                   54/63
app01:    Installing : 1:libglvnd-egl-1.0.1-0.8.git5baa1e5.el7.x86_64          55/63
```



Automated_provisioning_db01_1686569864733_98676
Running

Automated_provisioning_mc01_1686570282388_4693
Running

Automated_provisioning_rmq01_1686570543976_43533
Running

Automated_provisioning_app01_1686570967491_68337
Running

General
Name: Automated_provisioning_app01_1686570967491_68337
Operating System: Red Hat (64-bit)

System
Base Memory: 1024 MB
Processors: 2
Boot Order: Hard Disk, Optical
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display
Video Memory: 4 MB
Graphics Controller: VBoxVGA
Remote Desktop Server Port: 5984
Recording: Disabled

Storage
Controller: IDE Controller
IDE Primary Device 0: packer-centos-7-x86_64-disk001.vmdk (Normal, 80.00 GB)

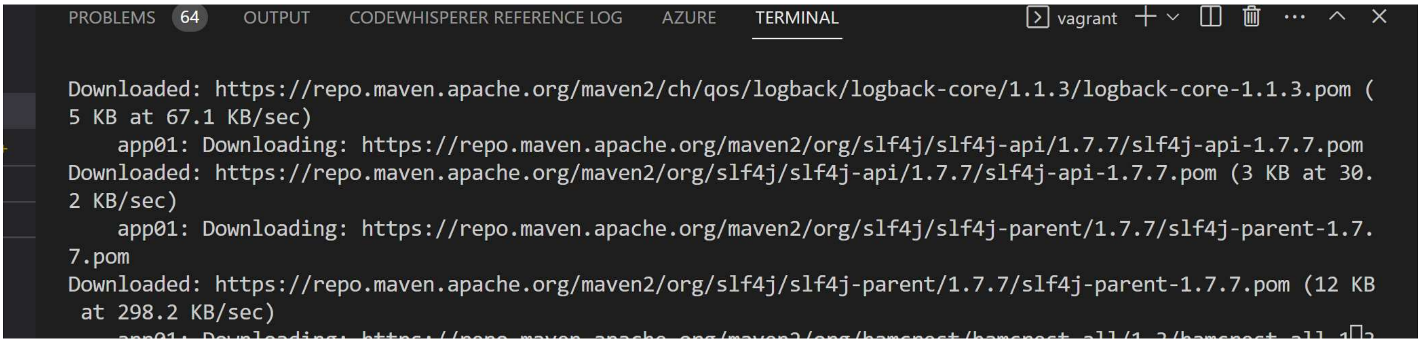
Audio
Disabled

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

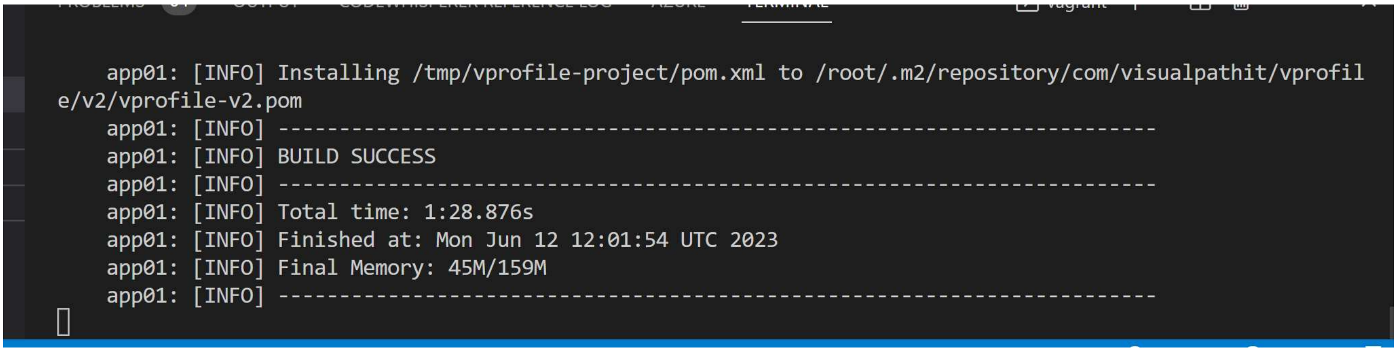
USB
Disabled

Shared folders
Shared Folders: 1

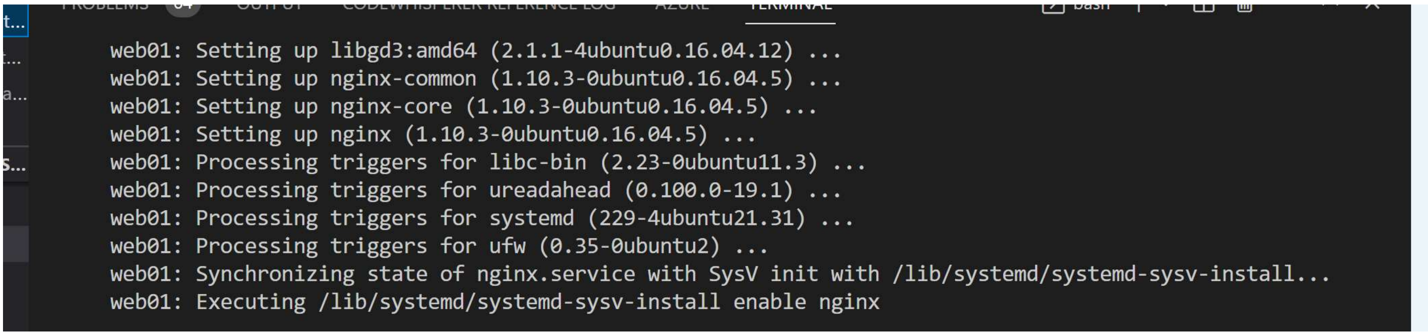
Description
None



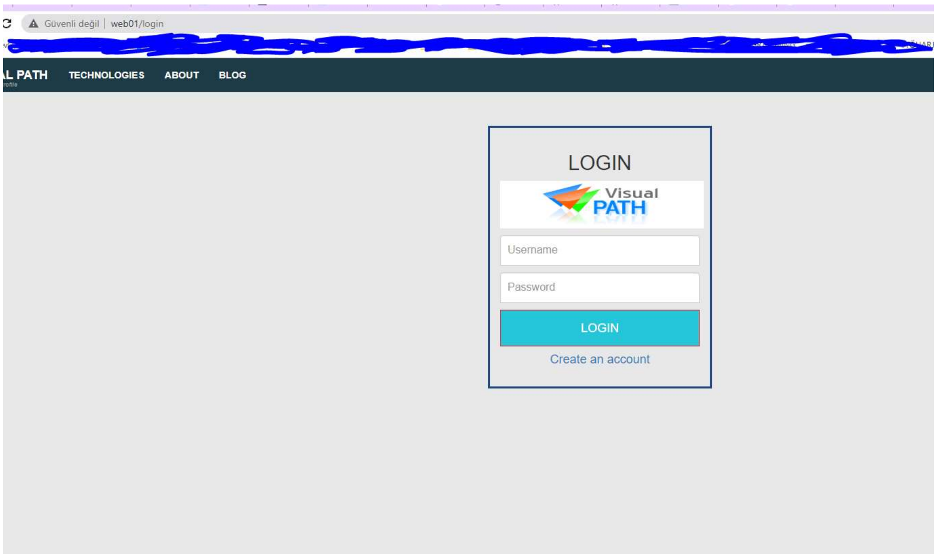
```
Downloaded: https://repo.maven.apache.org/maven2/ch/qos/logback/logback-core/1.1.3/logback-core-1.1.3.pom (
5 KB at 67.1 KB/sec)
    app01: Downloading: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.7/slf4j-api-1.7.7.pom
Downloaded: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.7/slf4j-api-1.7.7.pom (3 KB at 30.
2 KB/sec)
    app01: Downloading: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-parent/1.7.7/slf4j-parent-1.7.
7.pom
Downloaded: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-parent/1.7.7/slf4j-parent-1.7.7.pom (12 KB
 at 298.2 KB/sec)
    app01: Downloading: https://repo.maven.apache.org/maven2/org/hamcrest/hamcrest-all/1.3/hamcrest-all-1.3
```

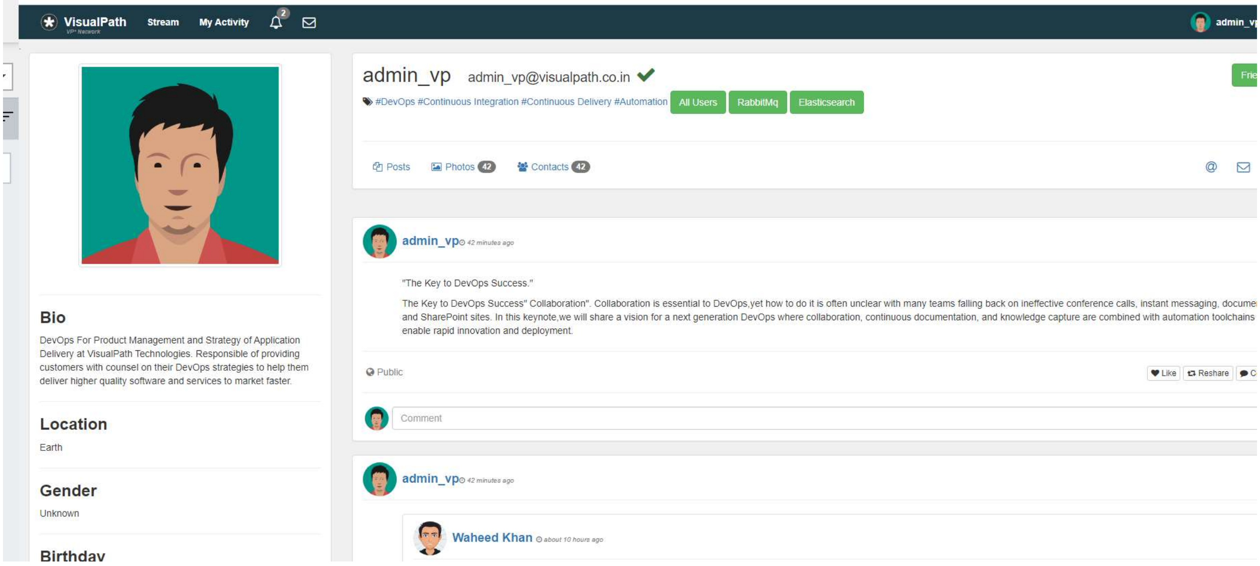


```
    app01: [INFO] Installing /tmp/vprofile-project/pom.xml to /root/.m2/repository/com/visualpathit/vprofil
e/v2/vprofile-v2.pom
    app01: [INFO] ------------------------------------------------------------------------
    app01: [INFO] BUILD SUCCESS
    app01: [INFO] ------------------------------------------------------------------------
    app01: [INFO] Total time: 1:28.876s
    app01: [INFO] Finished at: Mon Jun 12 12:01:54 UTC 2023
    app01: [INFO] Final Memory: 45M/159M
    app01: [INFO] ------------------------------------------------------------------------
```



```
    web01: Setting up libgd3:amd64 (2.1.1-4ubuntu0.16.04.12) ...
    web01: Setting up nginx-common (1.10.3-0ubuntu0.16.04.5) ...
    web01: Setting up nginx-core (1.10.3-0ubuntu0.16.04.5) ...
    web01: Setting up nginx (1.10.3-0ubuntu0.16.04.5) ...
    web01: Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
    web01: Processing triggers for ureadahead (0.100.0-19.1) ...
    web01: Processing triggers for systemd (229-4ubuntu21.31) ...
    web01: Processing triggers for ufw (0.35-0ubuntu2) ...
    web01: Synchronizing state of nginx.service with SysV init with /lib/systemd/systemd-sysv-install...
    web01: Executing /lib/systemd/systemd-sysv-install enable nginx
```





admin_vp admin_vp@visualpath.co.in ✓

#DevOps #Continuous Integration #Continuous Delivery #Automation

admin_vp 42 minutes ago

"The Key to DevOps Success."

The Key to DevOps Success" Collaboration". Collaboration is essential to DevOps,yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, docume and SharePoint sites. In this keynote,we will share a vision for a next generation DevOps where collaboration, continuous documentation, and knowledge capture are combined with automation toolchains enable rapid innovation and deployment.

Public

admin_vp 42 minutes ago

Waheed Khan about 10 hours ago

Bio

DevOps For Product Management and Strategy of Application Delivery at VisualPath Technologies. Responsible of providing customers with counsel on their DevOps strategies to help them deliver higher quality software and services to market faster.

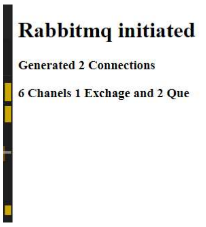
Location

Earth

Gender

Unknown

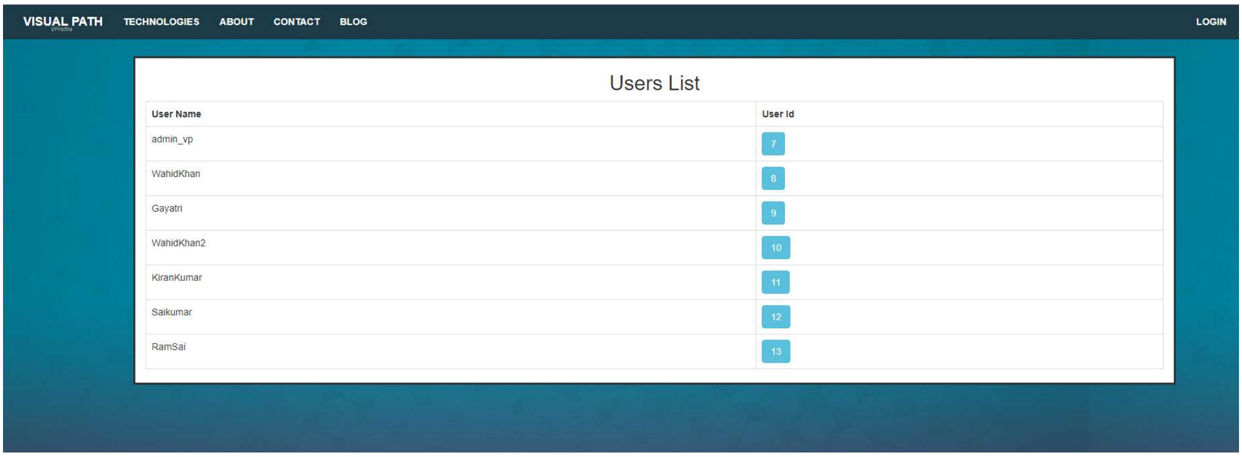
Birthday



Rabbitmq initiated

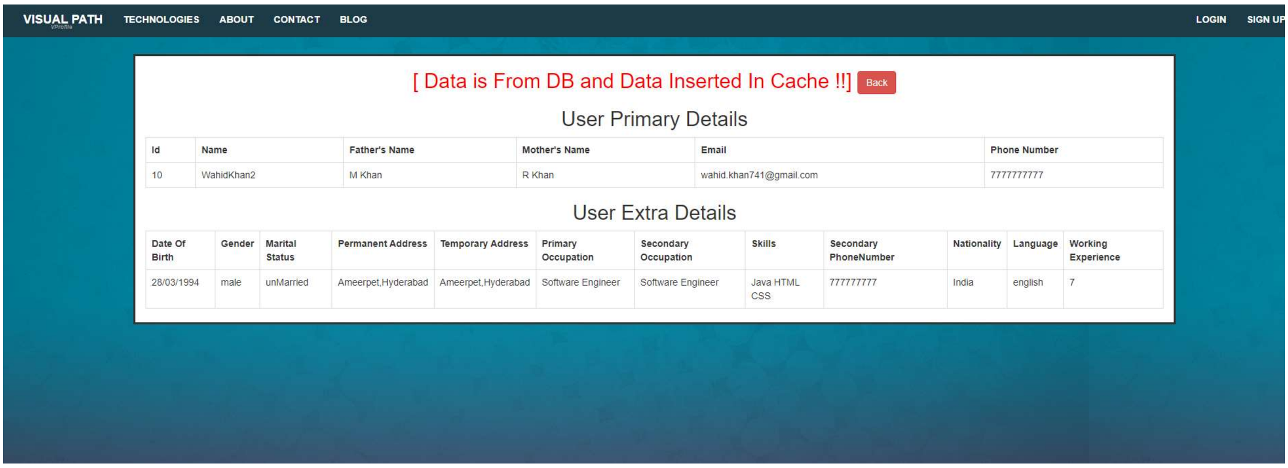
Generated 2 Connections

6 Chanels 1 Exchage and 2 Que



Users List

User Name	User Id
admin_vp	7
WahidKhan	8
Gayatri	9
WahidKhan2	10
KiranKumar	11
Saikumar	12
RamSai	13



[Data is From DB and Data Inserted In Cache !!]

User Primary Details

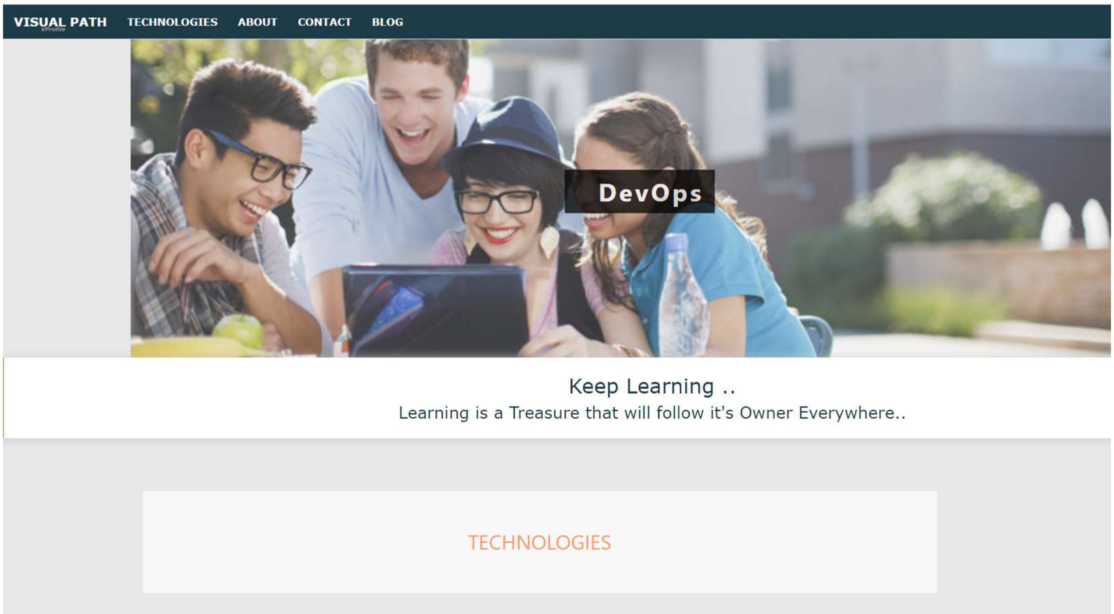
Id	Name	Father's Name	Mother's Name	Email	Phone Number
10	WahidKhan2	M Khan	R Khan	wahid.khan741@gmail.com	7777777777

User Extra Details

Date Of Birth	Gender	Marital Status	Permanent Address	Temporary Address	Primary Occupation	Secondary Occupation	Skills	Secondary PhoneNumber	Nationality	Language	Working Experience
28/03/1994	male	unMarried	Ameerpet,Hyderabad	Ameerpet,Hyderabad	Software Engineer	Software Engineer	Java HTML CSS	777777777	India	english	7

Güvenli değil | web01/index

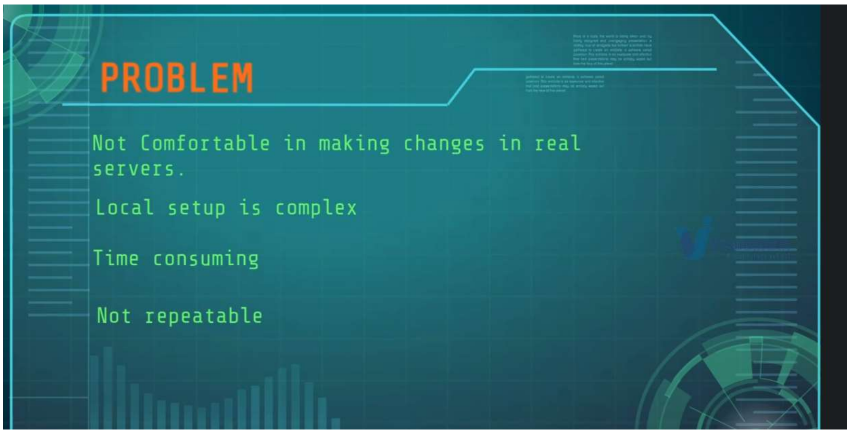
,



Keep Learning ..

Learning is a Treasure that will follow it's Owner Everywhere..

TECHNOLOGIES



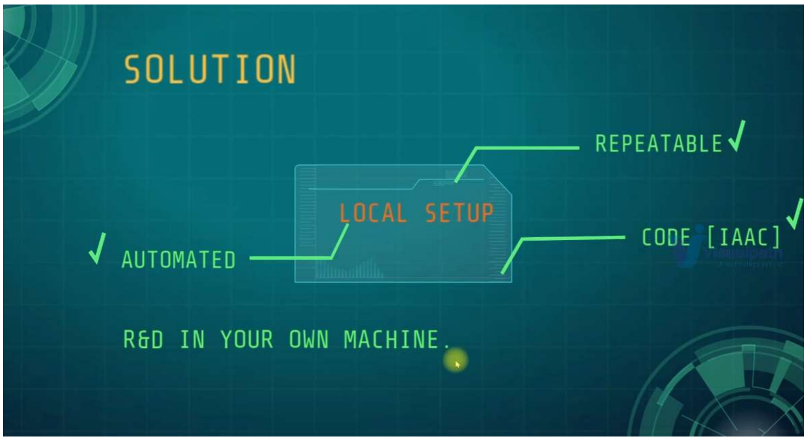
PROBLEM

Not Comfortable in making changes in real servers.

Local setup is complex

Time consuming

Not repeatable



SOLUTION

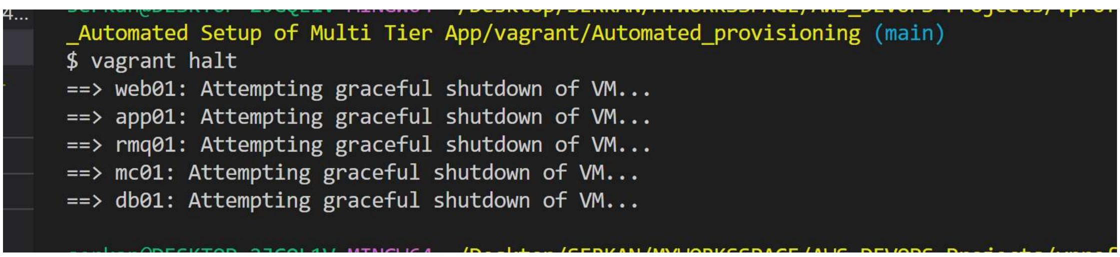
LOCAL SETUP

REPEATABLE ✓

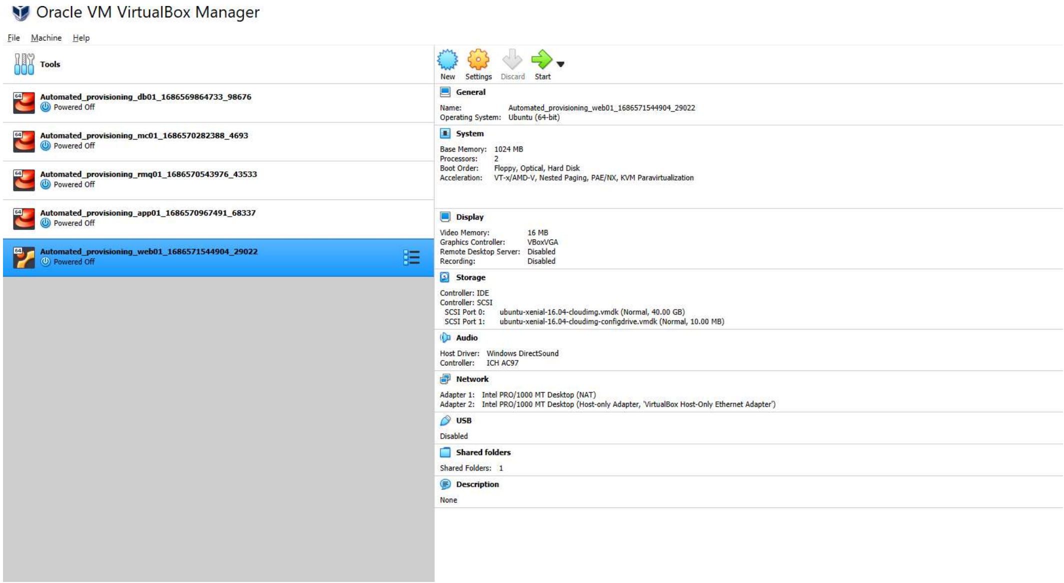
CODE [IAAC] ✓

✓ AUTOMATED

R&D IN YOUR OWN MACHINE.



```
serkan@DESKTOP-23C0L1V MINGW64 ~/Desktop/SERKAN/MYWORKSSPACE/AWS_DEVOPS-Projects/vprof
_Automated Setup of Multi Tier App/vagrant/Automated_provisioning (main)
$ vagrant halt
==> web01: Attempting graceful shutdown of VM...
==> app01: Attempting graceful shutdown of VM...
==> rmq01: Attempting graceful shutdown of VM...
==> mc01: Attempting graceful shutdown of VM...
==> db01: Attempting graceful shutdown of VM...
serkan@DESKTOP-23C0L1V MINGW64 ~/Desktop/SERKAN/MYWORKSSPACE/AWS_DEVOPS-Projects/vprof
```



Automated_provisioning_db01_1686569864733_98676
Powered Off

Automated_provisioning_mc01_1686570282388_4693
Powered Off

Automated_provisioning_rmq01_1686570543976_43533
Powered Off

Automated_provisioning_app01_1686570967491_68337
Powered Off

Automated_provisioning_web01_1686571544904_29022
Powered Off

General

Name: Automated_provisioning_web01_1686571544904_29022
Operating System: Ubuntu (64-bit)

System

Base Memory: 1024 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VBoxVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
Controller: SCSI
SCSI Port 0: ubuntu-xenial-16.04-cloudimg.vmdk (Normal, 40.00 GB)
SCSI Port 1: ubuntu-xenial-16.04-cloudimg-configdrive.vmdk (Normal, 10.00 MB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

USB

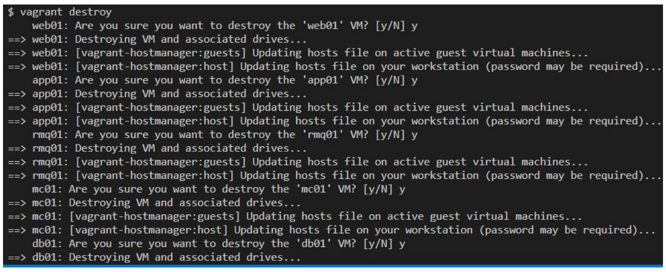
Disabled

Shared folders

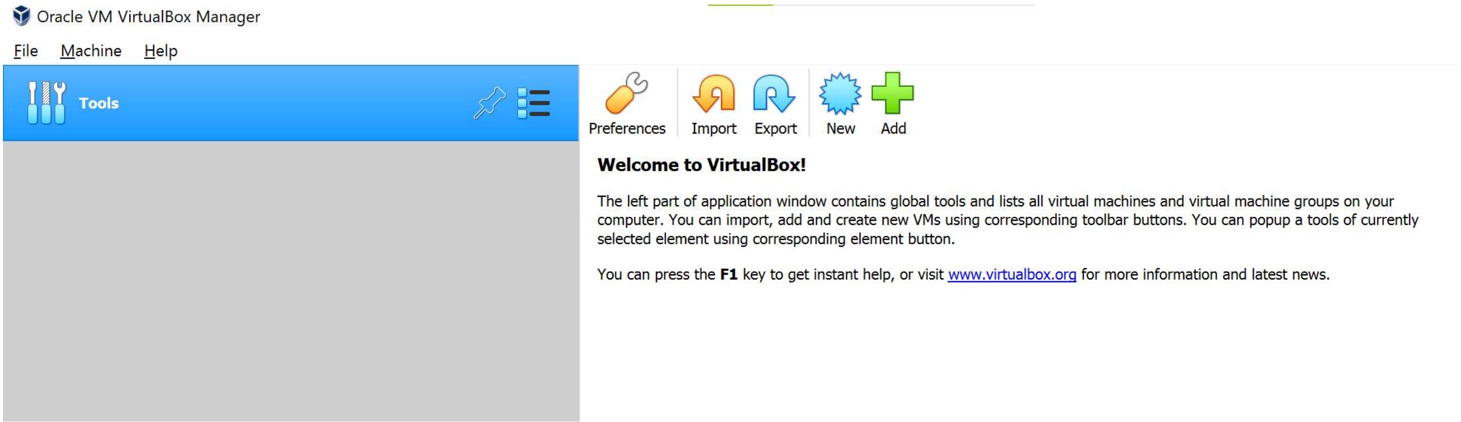
Shared Folders: 1

Description

None



```
$ vagrant destroy
    web01: Are you sure you want to destroy the 'web01' VM? [y/N] y
==> web01: Destroying VM and associated drives...
==> web01: [vagrant-hostmanager:guests] Updating hosts file on active guest virtual machines...
==> web01: [vagrant-hostmanager:host] Updating hosts file on your workstation (password may be required)...
    app01: Are you sure you want to destroy the 'app01' VM? [y/N] y
==> app01: Destroying VM and associated drives...
==> app01: [vagrant-hostmanager:guests] Updating hosts file on active guest virtual machines...
==> app01: [vagrant-hostmanager:host] Updating hosts file on your workstation (password may be required)...
    rmq01: Are you sure you want to destroy the 'rmq01' VM? [y/N] y
==> rmq01: Destroying VM and associated drives...
==> rmq01: [vagrant-hostmanager:guests] Updating hosts file on active guest virtual machines...
==> rmq01: [vagrant-hostmanager:host] Updating hosts file on your workstation (password may be required)...
    mc01: Are you sure you want to destroy the 'mc01' VM? [y/N] y
==> mc01: Destroying VM and associated drives...
==> mc01: [vagrant-hostmanager:guests] Updating hosts file on active guest virtual machines...
==> mc01: [vagrant-hostmanager:host] Updating hosts file on your workstation (password may be required)...
    db01: Are you sure you want to destroy the 'db01' VM? [y/N] y
==> db01: Destroying VM and associated drives...
```



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.