Vagrant Setup for DB2

Contents

[Pre-requesites on Windows 3](#_Toc521927523)

[System proxy configuration 3](#_Toc521927524)

[a) On Windows 3](#_Toc521927525)

[b) On Linux 3](#_Toc521927526)

[Create a new vagrant environment 3](#_Toc521927527)

[a) Setup the vagrant proxy 3](#_Toc521927528)

[b) Boot the vagrant environment 4](#_Toc521927529)

[c) Connect to the Virtual Machine 4](#_Toc521927530)

[a) Vagrant shared folder 5](#_Toc521927531)

[Docker CE installation 5](#_Toc521927532)

[a) Docker repository setup 5](#_Toc521927533)

[b) Docker installation 5](#_Toc521927534)

## Pre-requisites on Windows

* .NET framework 4.5 minimum
* Powershell 3 minimum
* VirtualBox

## System proxy configuration

### On Windows

Open a new terminal:

C:\> set http\_proxy=http://10.152.1.11:8080

C:\> set https\_proxy=%http\_proxy%

C:\> vagrant plugin install vagrant-proxyconf

### On Linux

Open a new terminal:

export http\_proxy="http://user:password@host:port"

export https\_proxy="http://user:password@host:port"

vagrant plugin install vagrant-proxyconf

## Create a new vagrant environment

C:\> cd <local\_directory\_for\_creating\_new\_environment>

C:\> vagrant init hashicorp/precise64

### Setup the vagrant proxy

Add the following lines to the **Vagrantfile** in order to define the proxy:

Vagrant.configure("2") do |config|

if Vagrant.has\_plugin?("vagrant-proxyconf")

config.proxy.http = "http://10.152.1.11:8080/"

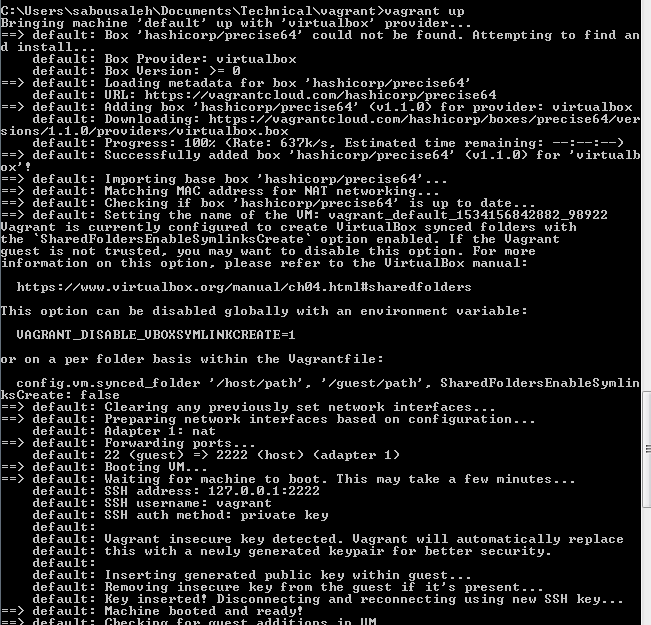
config.proxy.https = "http://10.152.1.11:8080/"

config.proxy.no\_proxy = "localhost,127.0.0.1,.example.com"

end

### Boot the vagrant environment

C:\> vagrant up



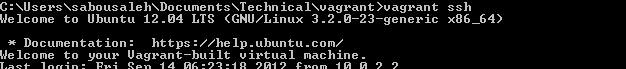
### Connect to the Virtual Machine

Check the vagrant VM status:

C:\> vagrant status

Connect to the new Vagrant environment via ssh:

C:\> vagrant ssh



Set the system date:

vagrant@precise64$> date -s "13 AUG 2018 11:55:00"

Create a new system user:

vagrant@precise64$> sudo useradd -d /home/devuser --shell /bin/bash devuser

vagrant@precise64$> sudo usermod –G sudo devuser

Check the Network connection by performing a system update:

vagrant@precise64$> sudo apt-get update

### Vagrant shared folder

The vagrant Ubuntu VM shares a folder with the host machine. On the host it is the folder containing the **Vagrantfile**, on the vagrant guest it is located under **/vagrant**



### Post install

Run the post install script located at **/home/vagrant** to clean up the installation files:

vagrant@precise64$> sudo ./postinstall.sh

## Docker CE installation

### Docker repository setup

Install packages to allow apt to use a repository over HTTPS:

devuser:~$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

You might need to add username and password to the proxy authentication, before running the command to add Docker’s official GPG key:

devuser:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

devuser:~$ sudo apt-key fingerprint 0EBFCD88

For Ubuntu 13.10 and older only (to get the **add-apt-repository** command ):

devuser:~$ sudo apt-get install -y python-software-properties

Set up the “stable” repository

devuser:~$ sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

Update the apt package repository

devuser:~$ sudo apt-get update

### Docker installation (Ubuntu 14+)

List available Docker CE versions:

devuser:~$ sudo apt-cache madison docker-ce

Install the latest Docker CE version:

devuser:~$ sudo apt-get install docker-ce

Check that Docker was installed successfully:

devuser:~$ docker -v

### Docker installation (Ubuntu 12.04-)

Kernels older than 3.10 lack some of the features required to run Docker containers.

devuser:~$ uname -r

upgrade your kernel version to 3.8+ and install required docker packages

devuser:~$ sudo apt-get install linux-image-generic-lts-raring linux-headers-generic-lts-raring

devuser:~$ sudo reboot

devuser:~$ sudo sh -c "wget -qO- https://get.docker.io/gpg | apt-key add -"

devuser:~$ sudo sh -c "echo deb http://get.docker.io/ubuntu docker main > /etc/apt/sources.list.d/docker.list"

devuser:~$ sudo apt-get update

devuser:~$ sudo apt-get install lxc-docker

### Daemon and service configuration

Create a systemd drop-in directory:

devuser:~$ sudo mkdir /etc/systemd/system/docker.service.d

Create a config file **http-proxy.conf** inside this directory:

devuser:~$ cd /etc/systemd/system/docker.service.d

devuser:~$ sudo touch http-proxy.conf

Edit file **http-proxy.conf** with the following lines:

[Service]

Environment="HTTP\_PROXY=http://10.152.1.11:8080/"

Environment="HTTPS\_PROXY=http://10.152.1.11:8080/"

Environment="NO\_PROXY=localhost,127.0.0.1,localaddress,.localdomain.com"

Configure the Docker daemon using a JSON file, create a file at **/etc/docker/daemon.json**

{

"debug": true

}

Once Docker is installed, you need to start the Docker daemon. Most Linux distributions use **systemctl** to start services. If you do not have **systemctl**, use the **service** command.

Flush changes:

* systemctl

devuser:~$ sudo systemctl daemon-reload

Restart Docker:

devuser:~$ sudo systemctl restart docker

* service

devuser:~$ sudo service docker start

List all services and check the presence of the docker service:

devuser:~$ service --status-all | grep docker

Check that docker is running:

devuser:~$ ps -ef | grep docker

devuser:~$ sudo service docker status

Stop / remove all of Docker containers:

devuser:~$ docker stop $(docker ps -a)

devuser:~$ docker rm $(docker ps -a)