Vagrant: Set IP Address, create user, ssh connection

utrains.org

Before starting this lesson, make sure you have gone through the previous lessons and that you have vagrant install on your computer

Table of content

List of required tools

1. Centos 7 vagrant Box (centos/7)
2. File editor (Vs Code, Notepad, gedit, ...)
3. Vagrantfile : click here to download

1

Introduction

What is Vagrantfile?

Introduction

- When we initialize a vagrant machine, with the command **vagrant init**, we have :
  - A new **Vagrantfile** created in our current directory
  - This file contains information about the location of the **vagrant box** that we will use to run the machine
- We can also modify this file in order to **customize the ip address of our server.**
- To start, we will initialize a **centos 7 box**.

Note: If you already have one, you can just go to the next step

2

Vagrantfile explained

Meaning of some important lines in the Vagrantfile

Vagrantfile explained

- Run the following commands (in your **Visual studio code** terminal) to initialize a centos 7 server using vagrant :
  - **mkdir utrainsCentos7**
  - **cd utrainsCentos7**
  - **vagrant init centos/7**
  - **ls**
- At this level, a **Vagrantfile** is created in the current directory.

```
PS C:\Users\hermann90\utransProject> cd .\utrainsCentos7\
PS C:\Users\hermann90\utransProject\utrainsCentos7> vagrant init centos/7
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
PS C:\Users\hermann90\utransProject\utrainsCentos7> ls


    Directory: C:\Users\hermann90\utransProject\utrainsCentos7


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/19/2022   7:56 AM           3084 Vagrantfile
```

Vagrantfile explained

- After the **vagrant init** command, a **Vagrantfile** is created.
- We can set up some parameters of the virtual machine in there (**network, memory of our server, ...**)
- By default, the file contains the IP address of the server that will be created and many commented lines (the lines that start with the **# character**)
- To activate any line in the file, you just need to remove the **#** in front of it
- Let's modify the content of the Vagrantfile in our Visual studio code editor:
- Run the command: **code Vagrantfile**

Vagrantfile explained

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "centos/7"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
```

This line represent the language in which the file is written: **Ruby** language

This line allows you to specify the **vagrant box** that will be used. We can modify it to any other box (**ubuntu, redhat, ...**) that exist in the **Vagrant cloud**

Vagrantfile explained

```
24  # accessing "localhost:8080" will access port 80 on the guest machine.
25  # NOTE: This will enable public access to the opened port
26  # config.vm.network "forwarded_port", guest: 80, host: 8080
27
28  # Create a forwarded port mapping which allows access to a specific port
29  # within the machine from a port on the host machine and only allow access
30  # via 127.0.0.1 to disable public access
31  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
32
33  # Create a private network, which allows host-only access to the machine
34  # using a specific IP.
35  # config.vm.network "private_network", ip: "192.168.33.10"
36
37  # Create a public network, which generally matched to bridged network.
38  # Bridged networks make the machine appear as another physical device on
39  # your network.
40  # config.vm.network "public_network"
41
42  # Share an additional folder to the guest VM. The first argument is
43  # the path on the host to the actual folder. The second argument is
44  # the path on the guest to mount the folder. And the optional third
45  # argument is a set of non-required options.
46  # config.vm.synced_folder "../data", "/vagrant_data"
47
```

This line specifies the listening port of the server. To make the line actif we can uncomment by removing the # sign.

This line specifies the **IP address of our server.**

Vagrantfile explained

```
47
48    # Provider-specific configuration so you can fine-tune various
49    # backing providers for Vagrant. These expose provider-specific options.
50    # Example for VirtualBox:
51    #
52    # config.vm.provider "virtualbox" do |vb|
53    #   # Display the VirtualBox GUI when booting the machine
54    #   vb.gui = true
55    #
56    #   # Customize the amount of memory on the VM:
57    #   vb.memory = "1024"
58    # end
59    #
60    # View the documentation for the provider you are using for more
61    # information on available options.
62
63    # Enable provisioning with a shell script. Additional provisioners such as
64    # Ansible, Chef, Docker, Puppet and Salt are also available. Please see the
65    # documentation for more information about their specific syntax and use.
66    # config.vm.provision "shell", inline: <<-SHELL
67    #   apt-get update
68    #   apt-get install -y apache2
69    # SHELL
70  end
```

This bloc of lines (line 52 to 58) specifies the **memory and the GUI configuration**

This block (63-69) allows to specify some commands that will be executed in the server when the **vagrant up** command is run

Vagrantfile explained

- The following is an example of vagrantfile with only few lines
- We have reduced this file by deleting most of the commented lines
- We have also configured here the **private network** with the ip address **192.168.50.10**

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  # load de centos7 box from vagrant cloud
  config.vm.box = "centos/7"
  config.vm.network "private_network", ip: "192.168.50.10"

  #change the value of the SSH configuration file, then restart the ssh service
  config.vm.provision "shell", inline: <<-SHELL
    sudo sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
    sudo systemctl restart sshd
  SHELL
end

```

These are some commands that the **system** will execute at startup

3

Vagrantfile modification

Modifying some parameters in the Vagrantfile

Vagrantfile modification

- Let's modify the content of our vagrant file: **code Vagrantfile**
- First, you need to delete the content of your vagrantfile
  - Select all the lines in the next slide (page 16)
  - Copy the lines (this will be our new vagrantfile content)
  - Paste that in your Vagrantfile and save with Ctrl - s

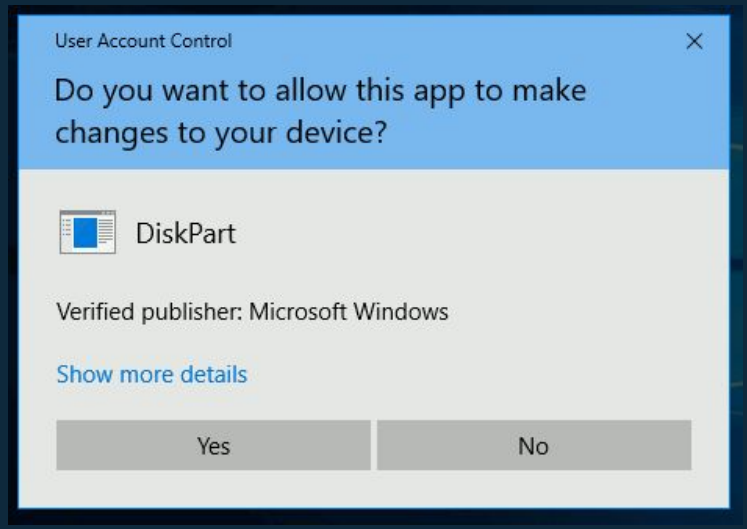
Vagrantfile modification

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
   # load de centos7 box from vagrant cloud
   config.vm.box = "centos/7"
   config.vm.network "private_network", ip: "192.168.50.10"
    #change the value of the SSH configuration file, then restart the ssh service
   config.vm.provision "shell", inline: <<-SHELL
    sudo sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
    sudo systemctl restart sshd
   SHELL
end
```

Vagrantfile modification

- After saving the modifications made in the Vagrantfile, just execute the following command in the terminal to start the server.
  - **vagrant up**
- If the following window appears, just validate on **Yes**

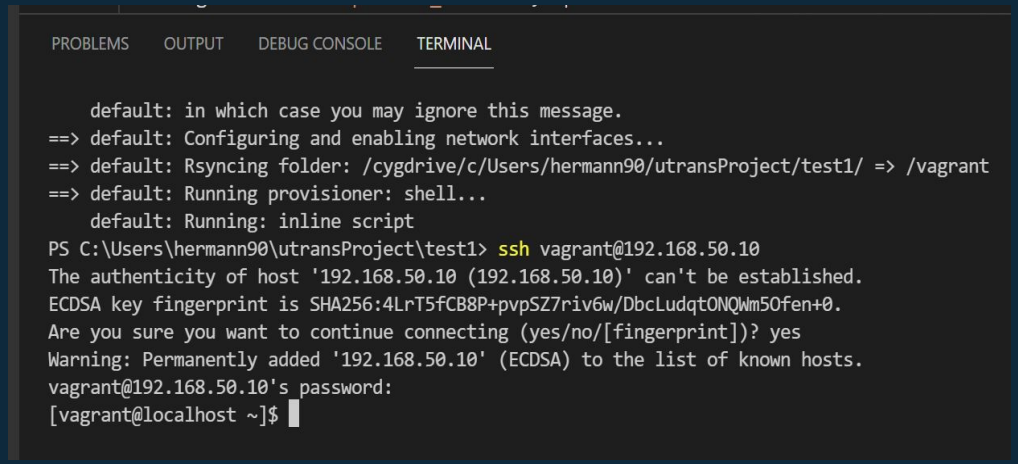


Remote connection to the server

The ssh command

Remote connection

- Now, let's connect remotely to the server using the following command:
  - **ssh vagrant@192.168.50.10**
  - Password: **vagrant**



In the company, we will use the ssh command to connect remotely to Linux servers. The syntax is:

ssh username@IPadress or

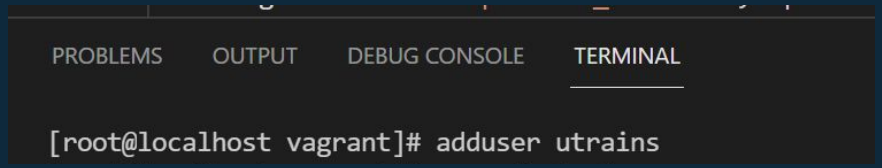
ssh username@serverDomainName

Example: ssh vagrant@192.168.50.10

Username IP address

Remote connection

- Let’s create another user called utrains on the system and connect remotely with the username via ssh.
- To create a user on the system, let’s switch to the root user’s account with the command $ **su** then password: **vagrant**
- Now, use the command: # **adduser utrains**



Remote connection

- Let’s create a password for the user with the command: # **passwd utrains**
- Enter the **Password: utrains**
- Retype the password to validate

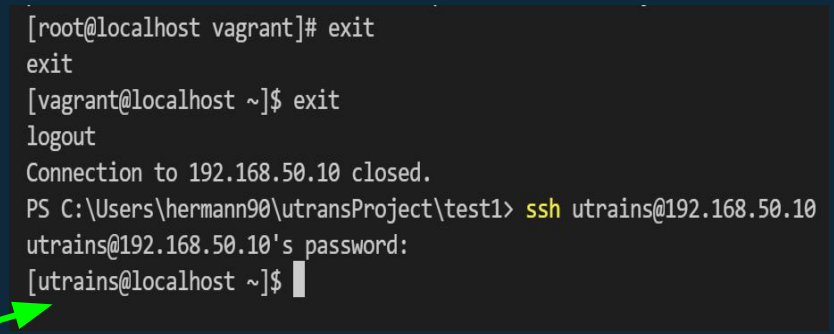
```
[root@localhost vagrant]# passwd utrains
Changing password for user utrains.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost vagrant]#
```

Remote connection

- Let’s logout of the server, then reconnect with the previously created user
  - **# exit**
  - **$ exit**
  - **ssh utrains@192.168.50.10**

    Password: utrains

The user **utrains** is connected



Practise this and make sure you follow the instructions step by step to get to the final result.

See you guys in the next lesson!

Thanks!

Any questions?

You can find us at:

website: http://utrains.org/

Phone: +1 (302) 689 3440

Email: contact@utrains.org

Click on the link below to contact the support team for any issue!

utrains.org/support/

Create a ticket for your problem and we will get back to you soon!