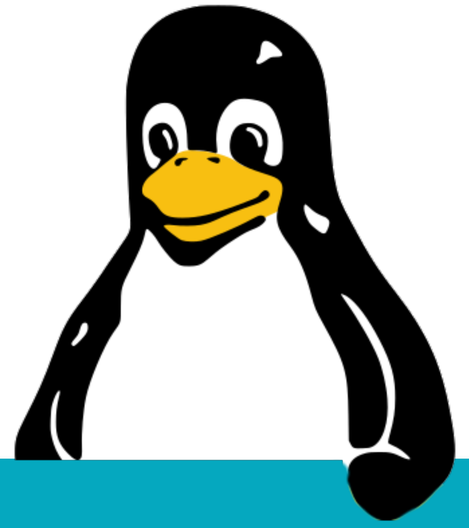
Linux, day 8



Objectives covered

Objective	Summary	Boek
3.2	Container management, container image ops	28
3.4	Virtuatization image files	28
3.4	Continuous integration / Continuous deployment	30
3.5	Bootstrapping	29

LAB: Vagrant

Let's install it!

- We will install Vagrant on our host OS.
  - It will control VirtualBox.

- Go to this site, download for your OS and install.
  - https://www.vagrantup.com/downloads
  - Or: "*[yum|apt|brew|winget] install vagrant*"
  - Windows will require a reboot.

Our first VM

- Open a terminal, or Powershell.
  - Go to your Downloads folder.
  - Make a new directory "*vagrant1*".
  - "*cd*" into the new "*vagrant1*" directory.
- Run: "*vagrant init debian/buster64*"
  - Check the "V*agrantfile*".

Boot your VM

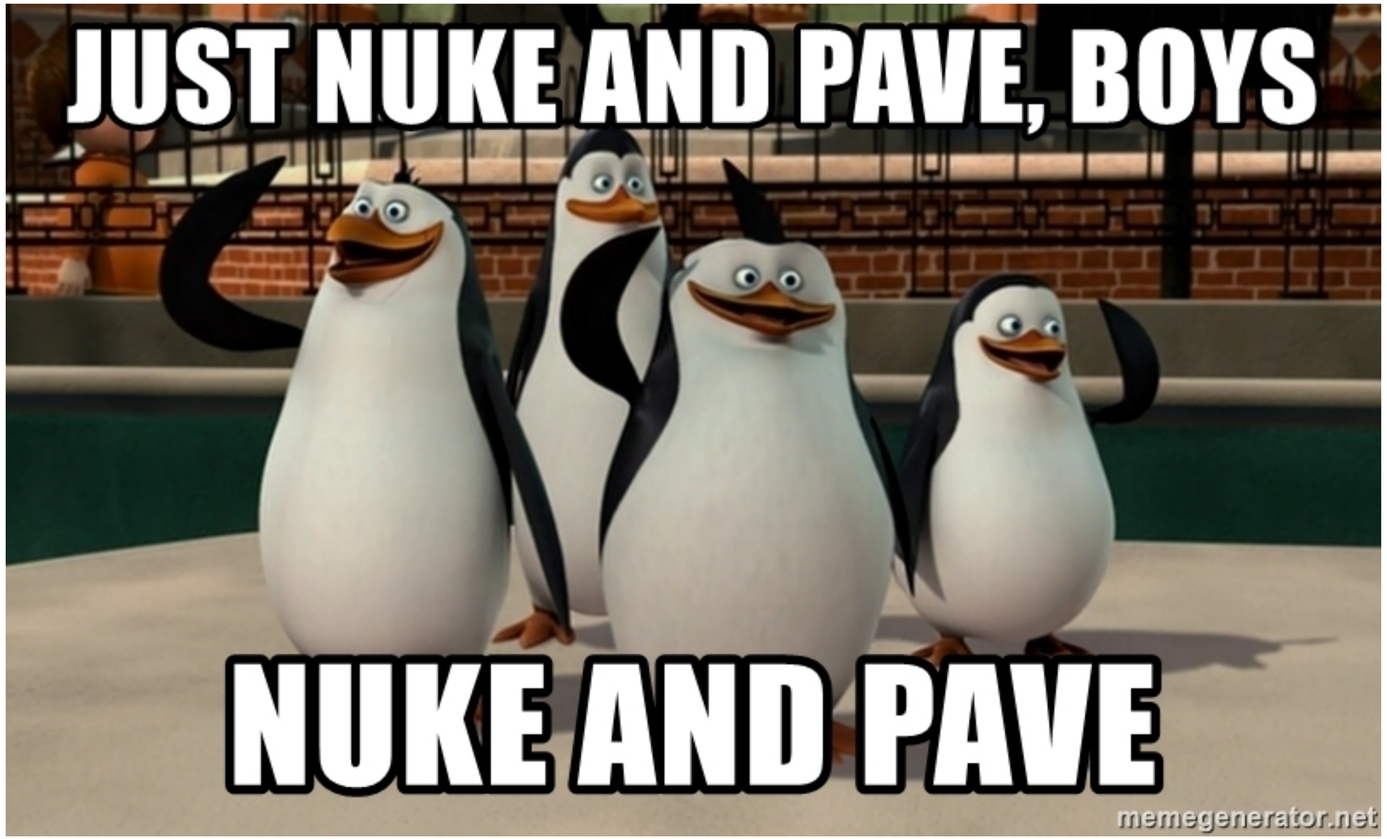
- Run: "*vagrant up*"
- This will:
  - Download the needed VM image.
  - Setup the VM in VirtualBox.
  - Setup the port forward for SSH.
  - And start the VM!

Boot your VM

- Booting will take a while. When it's done:
  - "*vagrant ssh*" logs you into the VM.
  - *"vagrant halt*" stops the VM.
  - "*vagrant destroy*" destroys the VM.
- Go ahead and destroy this VM.
  - If you "*vagrant up*" again it's now faster.

Problem? Broke something?

- Just:
  - *vagrant destroy*
  - *vagrant up*



Let's do something cool

- I have provided you with a sample *Vagrantfile*.
  - *"008 - Vagrantfile"*
- In your Downloads folder, make a dir "*vagrant2*".
- "*cd*" into "*vagrant2*".
- Now copy the *008-Vagrantfile* into "*vagrant2*".
  - Rename to "*Vagrantfile*".
  - Yes, a capital V.

Let's read the Vagrantfile!

- The syntax is more complicated than before!
- It has a number of recognizable blocks.
- Can you figure out what we're doing here?

Note: On Intel i-series and Windows 11, you must change the CPU core count to "2".

Boot the test network

- Run "*vagrant up"* in the *"vagrant2"* directory.
  - This will take longer! Now it's 3 VMs!
  - Afterwards, you can browse to:
    - http://localhost:8081
    - http://localhost:8082
    - http://localhost:8083

Recap: Vagrant

- Vagrant is not a hypervisor, it's IaC.
- Vagrant tells the hypervisor what to build.

Challenge

Challenge!

- Based on my Vagrantfile (with Alpine),
  - Can you make a new Vagrantfile for:
    - One VM, on 192.168.56.33
    - With a port forward of 9080 (host) to 80 (guest).
    - Running *lighttpd*, with the following content?
    - https://github.com/cloudacademy/static-website-example

Made a mistake?

- Mistakes in the post-install script?
  - No need to destroy!
  - Just run "*vagrant provision*".

Step by step

- The Vagrantfile should have:
  - Not three but one host.
  - An adjusted port forward.
  - "*git*" added via the "*apk add*" command
  - A "*git clone*", with the files copied into *htdocs*.
  - Fix the file permissions for files+dirs in *htdocs*.

Spoilers!

- Yes... "*008 - VagrantSpoilers*" is the solution.
  - Try it without spoilers first.

LAB: Docker

Running Docker

- Docker Desktop is great, but...
- But for our labs it might take too much time.
  - It relies on WSL2, a separate install.
  - On MacOS it's easy, but let's all do the same thing.

- So instead, let's use Docker on one of our VMs!

Let's install it on Ubuntu

- Ubuntu is easy.

```
$ sudo apt install -y docker.io

$ sudo systemctl start docker
```

Fedora is harder! 🤕

```
$ sudo yum install -y yum-utils
$ sudo yum-config-manager \
--add-repo \
https://download.docker.com/linux/fedora/
docker-ce.repo

$ sudo yum install docker-ce
$ sudo systemctl start docker
```

A quick test

- Let's see if we can run something!

```
$ sudo docker pull hello-world

$ sudo docker run hello-world
```

Our first container

- In Teams you will find "*008 - Docker.tgz*"
  - Copy this to your VM.
- On your VM, go to your Downloads folder.
  - Extract "*008 - Docker.tgz*".
  - This makes "*~/Downloads/docker-alp/*".

Building the container

- Run:

```
$ sudo docker build -t tess/demo .

$ sudo docker run -ti -p 8080:80 tess/demo
```

Result?

- Use Fedora's browser to visit:
  - http://localhost:8080
- Or on the command line:
  - *curl http://localhost:8080*

Looking at Docker

- More info? Debugging? What's running?
  - *docker image ls*
  - *docker ps*
  - *docker exec -ti ${containerID} /bin/sh*
  - *docker logs ${containerID}*

Recap: Docker

- Docker is not a hypervisor, it's IaC and more.
- Docker tells a container runtime what to do.
- Commands: *pull, build, run*
  ps, kill, image ls
  logs, exec

Challenge

Challenge!

- You have made all kinds of Python scripts, right?
- Can you make a container that runs one?
  - Literally, just run your Python script in a container.

Step by step

- You will need to:
  - Base on a suitable image, like *"python:slim-buster"*.
  - Put your script in the build directory.
  - Set the script as CMD,
  - With Python as ENTRYPOINT.

Spoilers!

- Yes... "*008 - DockerSpoilers*" is the solution.
  - Try it without spoilers first.

Closing

Homework

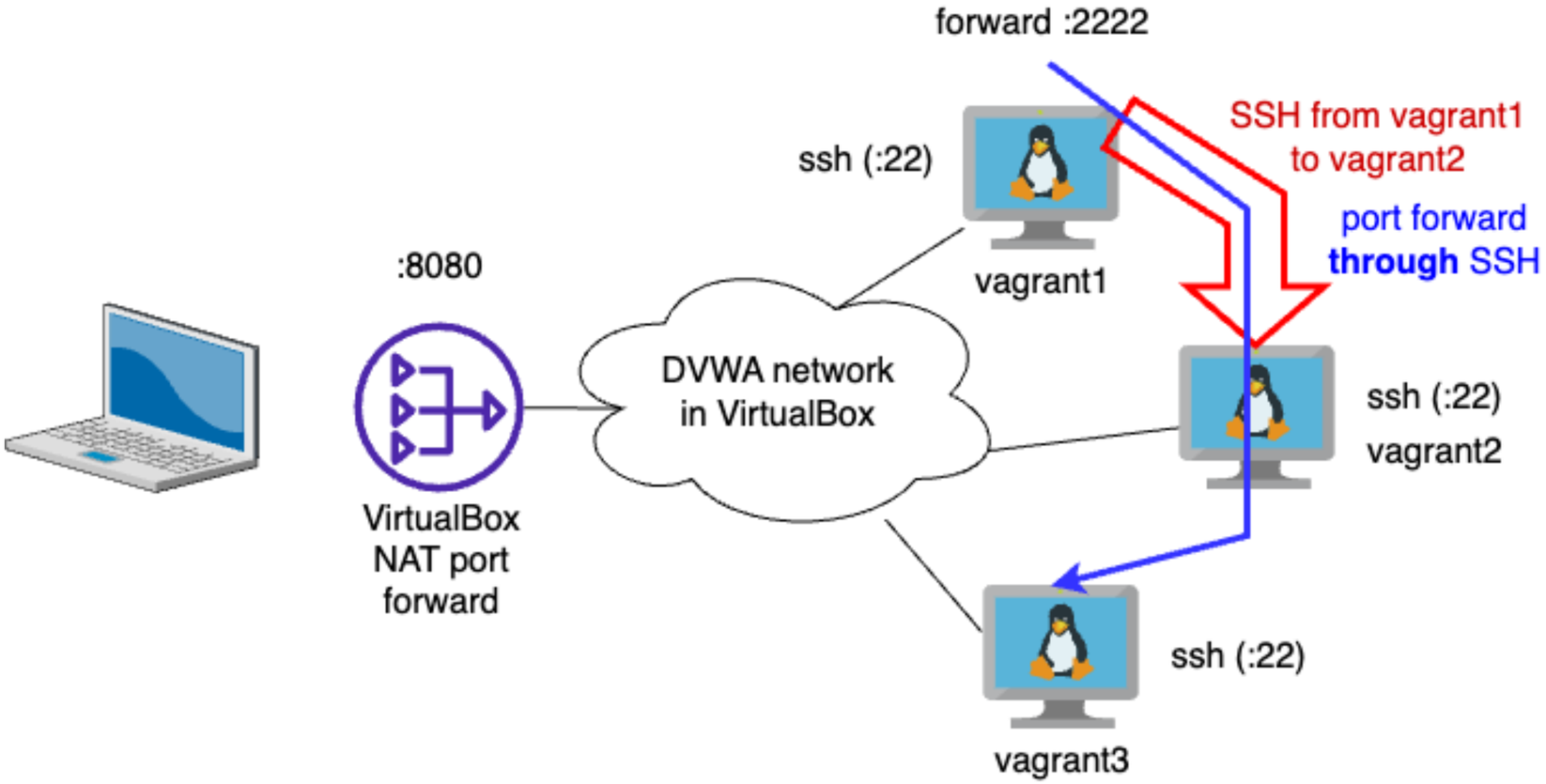
- Reading:
  - Chapter 11, p. 329-348
  - Chapter 16.

Homework - task 1

- Go do:
  - Use the three VMs made by Vagrant (*vagrant1*).
  - SSH from vagrant3 via vagrant2. That's easy!
  - But can you setup a port forward?!
    - Make vagrant3:22 available on vagrant1:2222.
    - Make the traffic flow through vagrant2.

Homework - task 1

- So, like this:



Homework - task 2

- Go do:
  - Use the three VMs made by Vagrant (*vagrant2*).
  - Setup RSync so */var/www/html* is synced,
    - From host 1, to hosts 2 and 3.
    - Make changes to your "*index.html*" and run *rsync*.
  - This does NOT need to go into your *Vagrantfile*.

Reference materials

Resources

- Understanding IaC in 10 minutes
- OVF? OVA? VMDK? File formats explained
- Cloud-Init, the good parts (advanced stuff!)
- Cloud-Init tutorials
- Cloud-init and Vagrant (tutorial)
- Does Docker run on Windows?

Resources

- Vagrant 101 [DevOps Journey]
- Getting started with Vagrant [Digital Ocean]
- CICD in 100 Seconds
- Introduction to DevOps and CICD
- Learn Docker - Full beginner's tutorial
- Ansible 101 with Jeff Geerling

Resources

- Nobody puts Java in the container!