

Labra 3

Palveluiden automatisointi

Ville Pulkkinen

Mikael Romanov

Markus Häkkinen

Harjoitustyö

Maaliskuu 2018

Tieto- ja viestintätekniikan koulutusohjelma

Tekniikan ja liikenteen ala

Sisällysluettelo

1	Johdanto	3
2	Vagrant	3
2.1	Vagrantfile	3
2.2	Vagrant boksit.....	3
2.3	Vagrant provisiointi	4
3	Ympäristön pystyttäminen.....	4
3.1	Koneiden määrittelyt.....	5
3.2	Vagrant komennot.....	6
3.3	Verkot	8
3.4	Docker provisiointi.....	9
4	Ntopng	9
5	Elasticsearch	11
6	Palvelun testaus.....	12
7	Pohdinta	13
8	Lähteet.....	14

Kuvioluettelo

Kuvio 1	VagrantCloud	4
Kuvio 2	Vagrant init	5
Kuvio 3	Vagrant default.....	5
Kuvio 4	Vagrant Config	5
Kuvio 5	Vagrant Provider	6
Kuvio 6	Vagrantfile	6
Kuvio 7	Vagrant up	6
Kuvio 8	Vagrant Provision	7

Kuvio 9 Vagrant Validate	7
Kuvio 10 Vagrant Komennot	7
Kuvio 11 Vagrant Topologia	8
Kuvio 12 Vagrant ntopng verkot	8
Kuvio 13 Vagrant elasticsearch verkot	9
Kuvio 14 Vagrant docker provisionti	9
Kuvio 15 Vagrant docker ntopng.....	10
Kuvio 16 Vagrant up ntopng.....	10
Kuvio 17 Vagrant docker elasticsearch	11
Kuvio 18 Vagrant up elasticsearch	11
Kuvio 19 Vagrant ntopng test.....	12
Kuvio 20 Vagrant ntopng flows	13
Kuvio 21 Vagrant elasticseach tcpdump	13

1 Johdanto

Laboratorion tavoitteena oli luoda usean virtuaalikoneen ympäristö, johon asennettiin aiemmissa labroissa määritellyt palvelut käyttäen vagrantia. Palvelut tuli kontittaa edellisen labran kontteja käyttäen.

2 Vagrant

Vagrant on työkalu, jolla voidaan hallita virtuaalikoneita, ja niiden ympäristöä.

Vagrantilla voidaan automatisoida virtuaalikoneiden luominen ja niiden asetusten muuttaminen.

2.1 Vagrantfile

Vagrantfile on konfiguraatiotiedosto, jota käytetään ympäristön määrittämiseen.

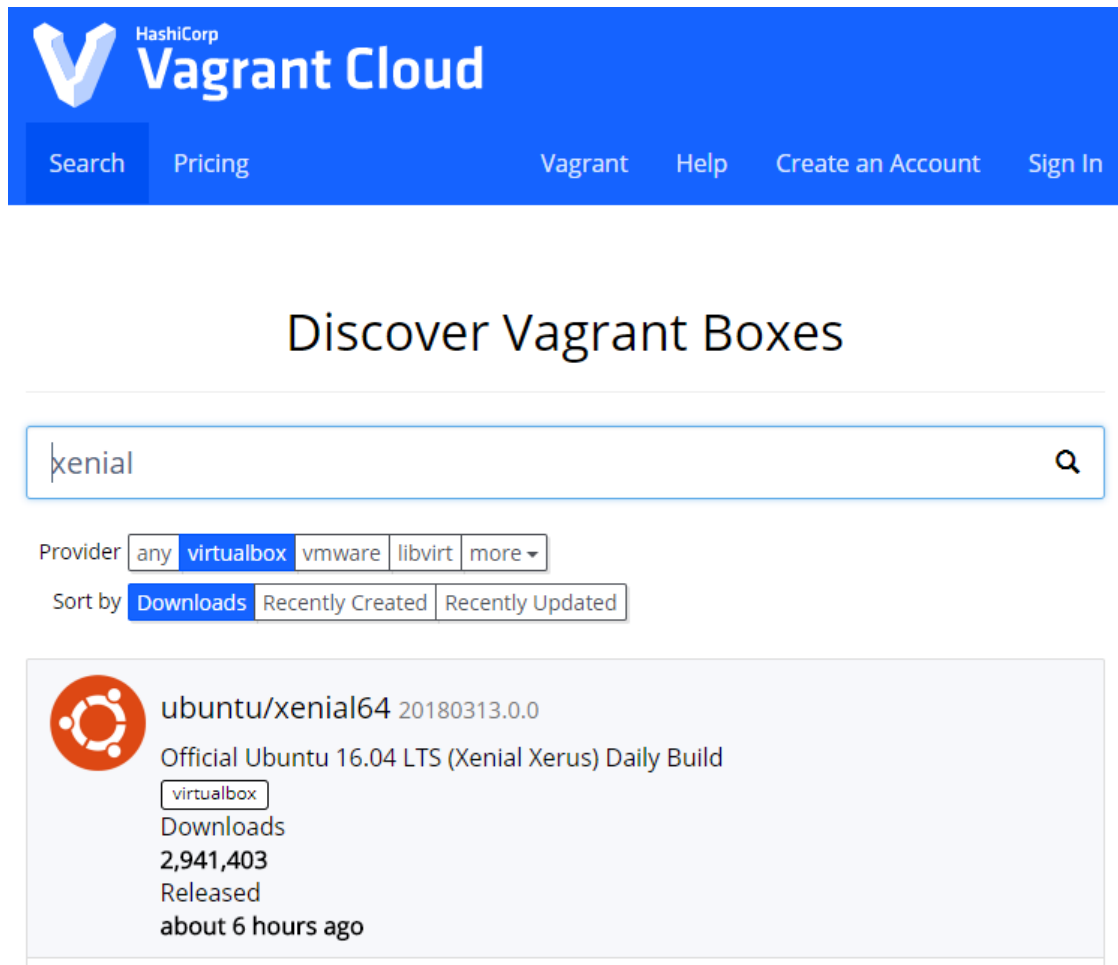
Vagrantfilessä on kuvattu käytettävä virtualisointialusta, virtuaalikoneiden tiedot sekä niiden provisiointikonfiguraatiot. Sama vagrantfile toimii kaikilla vagrantin tukemilla alustoilla.

2.2 Vagrant boksit

Vagrant boksit, ovat valmiita virtuaalikoneen pohjia joita voi käyttää projekteissa.

Vagrant ei sisällä mitään bokseja valmiina vaan voit käyttää julkisia bokseja

VagrantCloudista ladattuna. Vagrant hakee boksit VagrantCloudista ellei paikallista konetta ole määritelty(Kuvio 1 VagrantCloud). Vagrant bokseja voi tehdä myös itse vagrantin sivuilta löytyvän ohjeen mukaan.



The screenshot shows the Vagrant Cloud website interface. At the top is a blue header with the HashiCorp Vagrant Cloud logo and navigation links: Search, Pricing, Vagrant, Help, Create an Account, and Sign In. Below the header is a large heading "Discover Vagrant Boxes". Underneath is a search bar containing the text "xenial" with a magnifying glass icon on the right. Below the search bar are filters for "Provider" (any, virtualbox, vmware, libvirt, more) and "Sort by" (Downloads, Recently Created, Recently Updated). The first search result is for "ubuntu/xenial64 20180313.0.0", described as the "Official Ubuntu 16.04 LTS (Xenial Xerus) Daily Build". It includes a "virtualbox" tag, shows "Downloads 2,941,403", and states it was "Released about 6 hours ago".

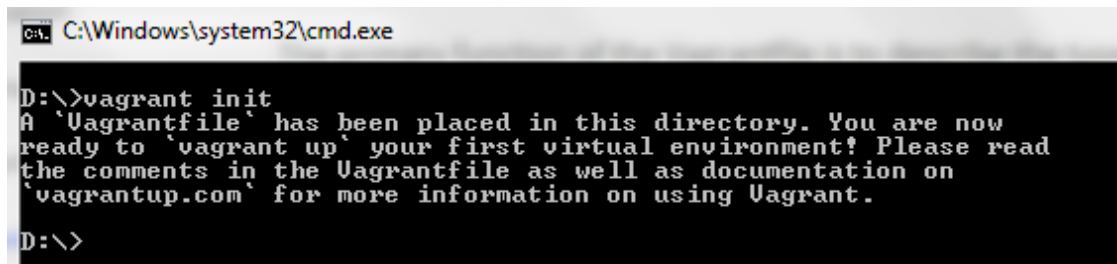
Kuvio 1 VagrantCloud

2.3 Vagrant provisiointi

Vagrant tarjoaa rajapinnan monelle ohjelmistolle provisiontia varten kuten, docker, ansible, puppet, salt ja Chef muutamia mainitakseni. Työssä käytettiin docker, shell ja file provisionteja.

3 Ympäristön pystyttäminen

Vagrant tarvitsee toimiakseen tuetun virtualisointi alustan. Käytimme harjoituksessa Oraclen Virtual boxia. Jotta ympäristö voidaan pystyttää isäntäkoneelle, täytyy ensin asentaa itse Vagrant. Vagranttia voidaan ajaa missä tahansa hakemistossa (Kuvio 2 Vagrant init)

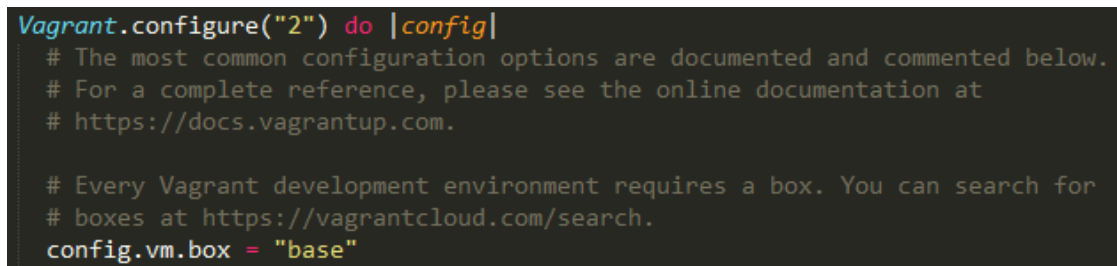


```
C:\Windows\system32\cmd.exe

D:\>vagrant init
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.
D:\>
```

Kuvio 2 Vagrant init

Vagranfilen oletuskonfiguraatio ei tee mitään ellei sille määritä pohjakonetta käsin tai komennolla "vagrant init pohjakoneen_nimi"(Vagrant default).



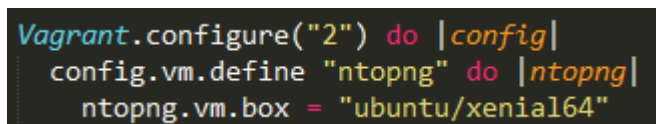
```
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

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

Kuvio 3 Vagrant default

3.1 Koneiden määrittelyt

Vagrantfileissä koneiden määrittäminen on yksinkertaista. Ensin luodaan vagrantin pohjakonfiguraation, jossa määritellään käytettävä virtualisointialusta (Kuvio 4 Vagrant Config). Peruskonfiguraation alle määritellään koneet erikseen komennolla "config.vm.define 'ntopng' do |ntopng| ". Koneelle täytyy vielä antaa pohjaboksi minkä se lataa ja ottaa käyttöön. Tämän jälkeen koneelle voidaan antaa erilaisia asetuksia ja provisionereita.



```
Vagrant.configure("2") do |config|
  config.vm.define "ntopng" do |ntopng|
    ntopng.vm.box = "ubuntu/xenial64"
```

Kuvio 4 Vagrant Config

Konfiguraatiossa määrätyn virtualisointialustan asetuksia voi muuttaa ympäristö, tai konekohtaisesti. Tässä harjoituksessa kaikilla koneilla on samat resurssit(Kuvio 5 Vagrant Provider)

```
config.vm.provider "virtualbox" do |vb|
  vb.gui = true
  vb.memory = 4096
  vb.cpus = 2
```

Kuvio 5 Vagrant Provider

Lopullinen Vagrantfile on hyvin yksinkertainen eikä konekohtaisia eroja ole kuin dockerin asetuksissa.(Kuvio 6 Vagrantfile)

```
1 # -*- mode: ruby -*-
2 # vi: set ft=ruby :
3
4
5 Vagrant.configure("2") do |config|
6   config.vm.define "ntopng" do |ntopng|
7     ntopng.vm.box = "ubuntu/xenial64"
8     ntopng.vm.network "forwarded_port", guest: 3000, host:3000
9     ntopng.vm.network "private_network", ip: "192.168.99.10"
10    ntopng.vm.provision "file", source: "ca.crt", destination: "ca.crt"
11    ntopng.vm.provision "shell", inline: "echo '35.204.211.40 jarmondocker.com' > /etc/hosts"
12    ntopng.vm.provision "shell", inline: "sudo mkdir -p /etc/docker/certs.d/jarmondocker.com:443/ && cp ca.crt /etc/docker/certs.d/jarmondocker.com:443/"
13    ntopng.vm.provision "docker" do |d|
14      d.run "jarmondocker.com:443/ntopng-image",
15        args: "--name 'ntopng' -p '3000:3000'",
16        #cmd: "--export-flows=es;flows;ntopng;http://192.168.99.22:9200/_bulk;"
17        cmd: "-F 'es;flows;ntopng;http://192.168.99.20:9200/_bulk;'"
18    end
19  end
20  config.vm.define "elasticsearch" do |elasticsearch|
21    elasticsearch.vm.box = "ubuntu/xenial64"
22    elasticsearch.vm.network "forwarded_port", guest: 9200, host:9200
23    elasticsearch.vm.network "private_network", ip: "192.168.99.20"
24    elasticsearch.vm.provision "file", source: "ca.crt", destination: "ca.crt"
25    elasticsearch.vm.provision "shell", inline: "echo '35.204.211.40 jarmondocker.com' > /etc/hosts"
26    elasticsearch.vm.provision "shell", inline: "sudo mkdir -p /etc/docker/certs.d/jarmondocker.com:443/ && cp ca.crt /etc/docker/certs.d/jarmondocker.com:443/"
27    elasticsearch.vm.provision "docker" do |d|
28      d.run "jarmondocker.com:443/es-engine-image",
29        args: "--name 'elasticsearch' -p '9200:9200' -p '9300:9300' -e bootstrap.memory_lock=true '-e ES_JAVA_OPTS=-Xms512m -Xmx512m'",
30        cmd: ""
31    end
32  end
33  config.vm.provider "virtualbox" do |vb|
34    vb.gui = true
35    vb.memory = 4096
36    vb.cpus = 2
37  end
38 end
```

Kuvio 6 Vagrantfile

3.2 Vagrant komennot

Vagrantissa on paljon hyödyllisiä komentoja jotka nopeuttavat työskentelyä. Vagrant ympäristö käynnistetään komennolla "vagrant up" jolloin vagrant ajaa kaikki vagrantfilessä määritetyt koneet ylös. Samalla komennolla voidaan ajaa myös vain tietty kone ylös, kirjoittamalla koneen nimi komennon perään(Kuvio 7 Vagrant up). Tämä on hyödyllinen, jos haluaa nopeasti vain tietyn koneen ylös.

```
D:\Vagrant>vagrant up ntopng
Bringing machine 'ntopng' up with 'virtualbox' provider...
==> ntopng: Checking if box 'ubuntu/xenial64' is up to date...
==> ntopng: A newer version of the box 'ubuntu/xenial64' for provider 'virtualbox' is
==> ntopng: available! You currently have version '20180306.0.0'. The latest is version
==> ntopng: '20180313.0.0'. Run 'vagrant box update' to update.
==> ntopng: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
==> ntopng: flag to force provisioning. Provisioners marked to run always will still run.
```

Kuvio 7 Vagrant up

Palvelujen provisionissa olisi todella hankalaa käynnistää koko kone uudestaan muutosten testaamiseksi joten pelkät provisionnit voidaan ajaa uudestaan komennolla "vagrant provision koneen_nimi"(Kuvio 8 Vagrant Provision)

```
D:\Uagrant>vagrant provision
==> ntopng: Running provisioner: file...
==> ntopng: Running provisioner: shell...
ntopng: Running: inline script
==> ntopng: Running provisioner: shell...
ntopng: Running: inline script
ntopng: sudo
ntopng: :
ntopng: unable to resolve host ubuntu-xenial
==> ntopng: Running provisioner: docker...
==> ntopng: Starting Docker containers...
==> ntopng: -- Container: jarmondocker.com:443/ntopng-image
==> ntopng: -- Detected newer image for container 'jarmondocker.com:443/ntopng-image'.
```

Kuvio 8 Vagrant Provision

Vagrantfileen tehtyjen muutosten syntaksivirheet voidaan tarkistaa komennolla "vagrant validate"(Kuvio 9 Vagrant Validate).

```
D:\Uagrant>vagrant validate
Vagrantfile validated successfully.

D:\Uagrant>
```

Kuvio 9 Vagrant Validate

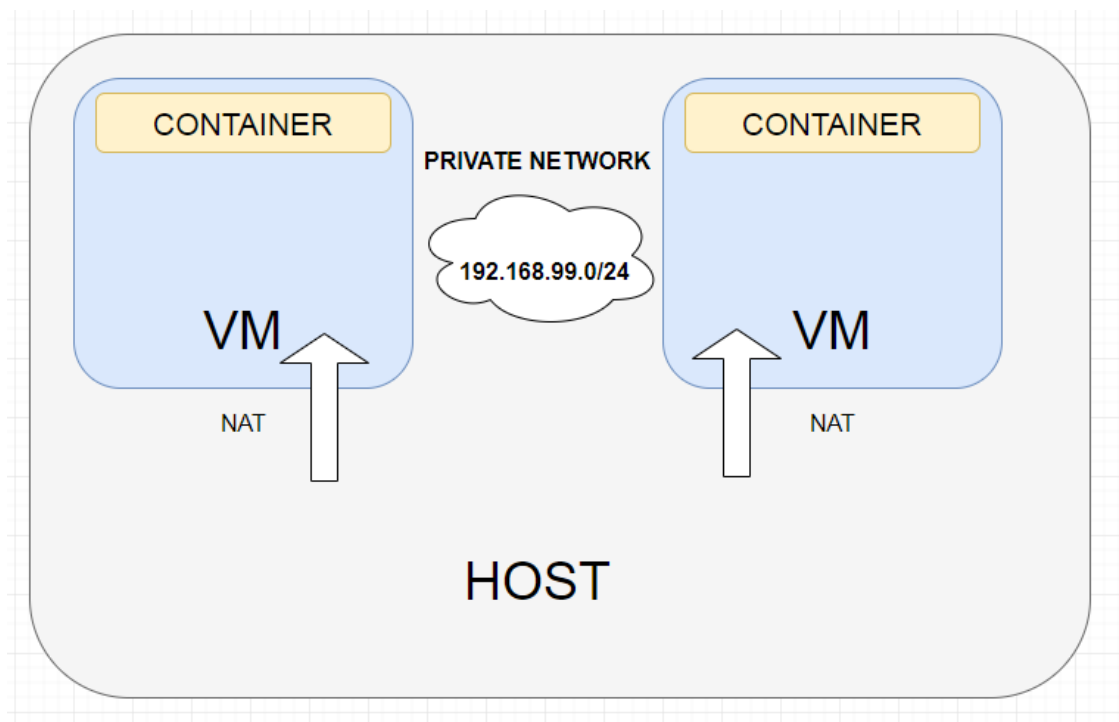
Vagrantissa on myös paljon muita hyödyllisiä työkaluja joita ei käydä läpi tässä harjoitustyössä(Kuvio 10 Vagrant Komennot).

```
Common commands:
  box                manages boxes: installation, removal, etc.
  destroy            stops and deletes all traces of the vagrant machine
  global-status      outputs status Vagrant environments for this user
  halt              stops the vagrant machine
  help              shows the help for a subcommand
  init              initializes a new Vagrant environment by creating a Vagrantfile
  login             log in to HashiCorp's Vagrant Cloud
  package           packages a running vagrant environment into a box
  plugin            manages plugins: install, uninstall, update, etc.
  port              displays information about guest port mappings
  powershell        connects to machine via powershell remoting
  provision          provisions the vagrant machine
  push             deploys code in this environment to a configured destination
  rdp               connects to machine via RDP
  reload            restarts vagrant machine, loads new Vagrantfile configuration
  resume            resume a suspended vagrant machine
  snapshot          manages snapshots: saving, restoring, etc.
  ssh               connects to machine via SSH
  ssh-config        outputs OpenSSH valid configuration to connect to the machine
  status            outputs status of the vagrant machine
  suspend           suspends the machine
  up                starts and provisions the vagrant environment
  validate          validates the Vagrantfile
  version           prints current and latest Vagrant version
```

Kuvio 10 Vagrant Komennot

3.3 Verkot

Konttien ollessa eri virtuaalikoneissa ja erikonteissa, niiden välille on tehtävä verkko, mitä pitkin data liikkuu. Työssä määriteltiin virtuaalikoneille oma host-only virtuaaliverkko 192.168.99.0/24. Vagrant vaati NAT-adapterin käyttöä hallitakseen koneita. Isäntäkoneelta palvelun hallintaa varten vagrantfileissä määritettiin tarvittavat porttiohjaukset(Kuvio 11 Vagrant Topologia)



Kuvio 11 Vagrant Topologia

Ntopng palvelua hallitaan web-käyttöliittymän kautta TCP-portissa 3000, joten ohjataan isäntäkoneen portti 3000, virtuaalikoneen porttiin 3000. Ntopng:lle annettiin kiinteä IP-osoite 192.168.99.10(Kuvio 12 Vagrant ntopng verkot)

```
ntopng.vm.network "forwarded_port", guest: 3000, host:3000
ntopng.vm.network "private_network", ip: "192.168.99.10"
```

Kuvio 12 Vagrant ntopng verkot

Elasticsearch tietokantaan voidaan suorittaa kyselyjä eri ohjelmistoilla, joten ohjaamme isäntäkoneen portin 9200 virtuaalikoneen porttiin 9200. Elasticsearch

koneelle annettiin kiinteä IP-osoite 192.168.99.20(Kuvio 13 Vagrant elasticsearch verkot)

```
elasticsearch.vm.network "forwarded_port", guest: 9200, host:9200
elasticsearch.vm.network "private_network", ip: "192.168.99.20"
```

Kuvio 13 Vagrant elasticsearch verkot

3.4 Docker provisionti

Vagrantfilessä määritetty provisionti dockerille asentaa kohdekoneeseen dockerin automaattisesti. Vagrant filellä voidaan määrittää tavallisimmat dockerissa tehtävät toiminnot, joitakin rajoituksia lukuunottamatta. Vagrantfilessä määritellään dockerin argumentit ja komentoriviparametrit omissa osioissaan(Kuvio 14 Vagrant docker provisionti)

```
ntopng.vm.provision "docker" do |d|
  d.run "jarmondocker.com:443/ntopng-image",
  args: "--name 'ntopng' -p '3000:3000'",
  cmd: "-F 'es;flows;ntopng;http://192.168.99.20:9200/_bulk;'"
end
```

Kuvio 14 Vagrant docker provisionti

4 Ntopng

Ntopng palvelun siirtäminen vagrantfileen on todella suoraaviivainen eikä siihen tarvitse suurempia muokkauksia. Ntopng:n vagrant pohjaboksina toimii ubuntu/xenial64. Edellisessä labrassamme tekemä docker kontti pitää hakea jarmondocker.com registrystä joten tarvitsemme koneeseen saman sertifikaatin jota käytettiin konttien puskemiseen ja hakemiseen. Sertifikaatti siirretään konttiin käyttäen "file" provisiontia. Seuraavana koneelle pitää määrittää missä osoitteessa jarmondocker.com, koska sillä ei ole julkista domain-osoitetta. Osoitteen voi kertoa koneelle käyttäen "shell" provisiontia jolla voidaan antaa kaikki komentorivikomennot mitä koneessa tarvitaan. Seuraavaksi koneelle luodaan kansiorakenne " /etc/docker/certs.d/jarmondocker.com:443/" jonne kopioidaan sertifikaatti "root" hakemistosta. Viimeisenä käytetään vagrantin docker provisiontia

joka tarjoaa yksinkertaisen rajapinnan dockerin käskyttämiseen virtuaalikoneessa(Kuvio 15 Vagrant docker ntopng)

```
ntopng.vm.provision "file", source: "ca.crt", destination: "ca.crt"
ntopng.vm.provision "shell", inline: "echo '35.204.211.40 jarmondocker.com' > /etc/hosts"
ntopng.vm.provision "shell", inline: "sudo mkdir -p /etc/docker/certs.d/jarmondocker.com:443"
ntopng.vm.provision "docker" do |d|
  d.run "jarmondocker.com:443/ntopng-image",
  args: "--name 'ntopng' -p '3000:3000'",
  #cmd: "'--export-flows=es;flows;ntopng;http://192.168.99.22:9200/_bulk;'"
  cmd: "-F 'es;flows;ntopng;http://192.168.99.20:9200/_bulk;'"
end
```

Kuvio 15 Vagrant docker ntopng

Komentoriviltä voidaan huomata missä vaiheessa provisointia vagrant on(Kuvio 16 Vagrant up ntopng).Mahdollisen vian paikallistamiseen voidaan käyttää Vagrantin "verbose" asetusta joka kertoo enemmän tietoa mitä ohjelma suorittaa.

```
D:\Vagrant>vagrant up
Bringing machine 'ntopng' up with 'virtualbox' provider...
Bringing machine 'elasticsearch' up with 'virtualbox' provider...
==> ntopng: Importing base box 'ubuntu/xenial64'...
==> ntopng: Matching MAC address for NAT networking...
==> ntopng: Checking if box 'ubuntu/xenial64' is up to date...
==> ntopng: A newer version of the box 'ubuntu/xenial64' for provider 'virtualbox' is
==> ntopng: available! You currently have version '20180306.0.0'. The latest is version
==> ntopng: '20180313.0.0'. Run 'vagrant box update' to update.
==> ntopng: Setting the name of the VM: Vagrant_ntopng_1520965311621_94019
==> ntopng: Clearing any previously set network interfaces...
==> ntopng: Preparing network interfaces based on configuration...
ntopng: Adapter 1: nat
ntopng: Adapter 2: hostonly
==> ntopng: Forwarding ports...
ntopng: 3000 (guest) => 3000 (host) (adapter 1)
ntopng: 22 (guest) => 2222 (host) (adapter 1)
==> ntopng: Running 'pre-boot' VM customizations...
==> ntopng: Booting VM...
==> ntopng: Waiting for machine to boot. This may take a few minutes...
ntopng: SSH address: 127.0.0.1:2222
ntopng: SSH username: vagrant
ntopng: SSH auth method: private key
ntopng: Warning: Connection aborted. Retrying...
ntopng: Warning: Connection reset. Retrying...
ntopng: Warning: Connection aborted. Retrying...
ntopng:
ntopng: Vagrant insecure key detected. Vagrant will automatically replace
ntopng: this with a newly generated keypair for better security.
ntopng:
ntopng: Inserting generated public key within guest...
ntopng: Removing insecure key from the guest if it's present...
ntopng: Key inserted! Disconnecting and reconnecting using new SSH key...
==> ntopng: Machine booted and ready!
==> ntopng: Checking for guest additions in VM...
ntopng: The guest additions on this VM do not match the installed version of
ntopng: VirtualBox! In most cases this is fine, but in rare cases it can
ntopng: prevent things such as shared folders from working properly. If you see
ntopng: shared folder errors, please make sure the guest additions within the
ntopng: virtual machine match the version of VirtualBox you have installed on
ntopng: your host and reload your VM.
ntopng:
ntopng: Guest Additions Version: 5.0.40
ntopng: VirtualBox Version: 5.2
==> ntopng: Configuring and enabling network interfaces...
==> ntopng: Mounting shared folders...
ntopng: /vagrant => D:\Vagrant
==> ntopng: Running provisioner: file...
==> ntopng: Running provisioner: shell...
ntopng: Running: inline script
==> ntopng: Running provisioner: shell...
ntopng: Running: inline script
ntopng: sudo: unable to resolve host ubuntu-xenial
==> ntopng: Running provisioner: docker...
ntopng: Installing Docker onto machine...
==> ntopng: Starting Docker containers...
==> ntopng: -- Container: jarmondocker.com:443/ntopng-image
```

Kuvio 16 Vagrant up ntopng

5 Elasticsearch

Elasticsearchin komennot ovat melkein samat vagrantin osalta kuin ntopng:n.

Ainoastaan docker provisionnin asetuksia muutetaan (Kuvio 17 Vagrant docker elasticsearch). Jos docker ilmoittaa virheestä voidaan konfiguraatioita korjata ja ajaa provisionti uudestaan komennolla "vagrant provision elasticsearch" (Kuvio 18 Vagrant up elasticsearch).

```

20 config.vm.define "elasticsearch" do |elasticsearch|
21   elasticsearch.vm.box = "ubuntu/xenial64"
22   elasticsearch.vm.network "forwarded_port", guest: 9200, host: 9200
23   elasticsearch.vm.network "private_network", ip: "192.168.99.20"
24   elasticsearch.vm.provision "file", source: "ca.crt", destination: "ca.crt"
25   elasticsearch.vm.provision "shell", inline: ["echo '35.204.211.40 jarmondocker.com' > /etc/hosts"]
26   elasticsearch.vm.provision "shell", inline: ["sudo mkdir -p /etc/docker/certs.d/jarmondocker.com:443/ && cp ca.crt /etc/docker/certs.d/jarmondocker.com:443/"]
27   elasticsearch.vm.provision "docker" do |d|
28     d.run "jarmondocker.com:443/es-engine-image",
29       args: "--name 'elasticsearch' -p '9200:9200' -p '9300:9300' -e bootstrap.memory_lock=true '-e ES_JAVA_OPTS=-Xms512m -Xmx512m'",
30       cmd: ""
31   end
32 end

```

Kuvio 17 Vagrant docker elasticsearch

```

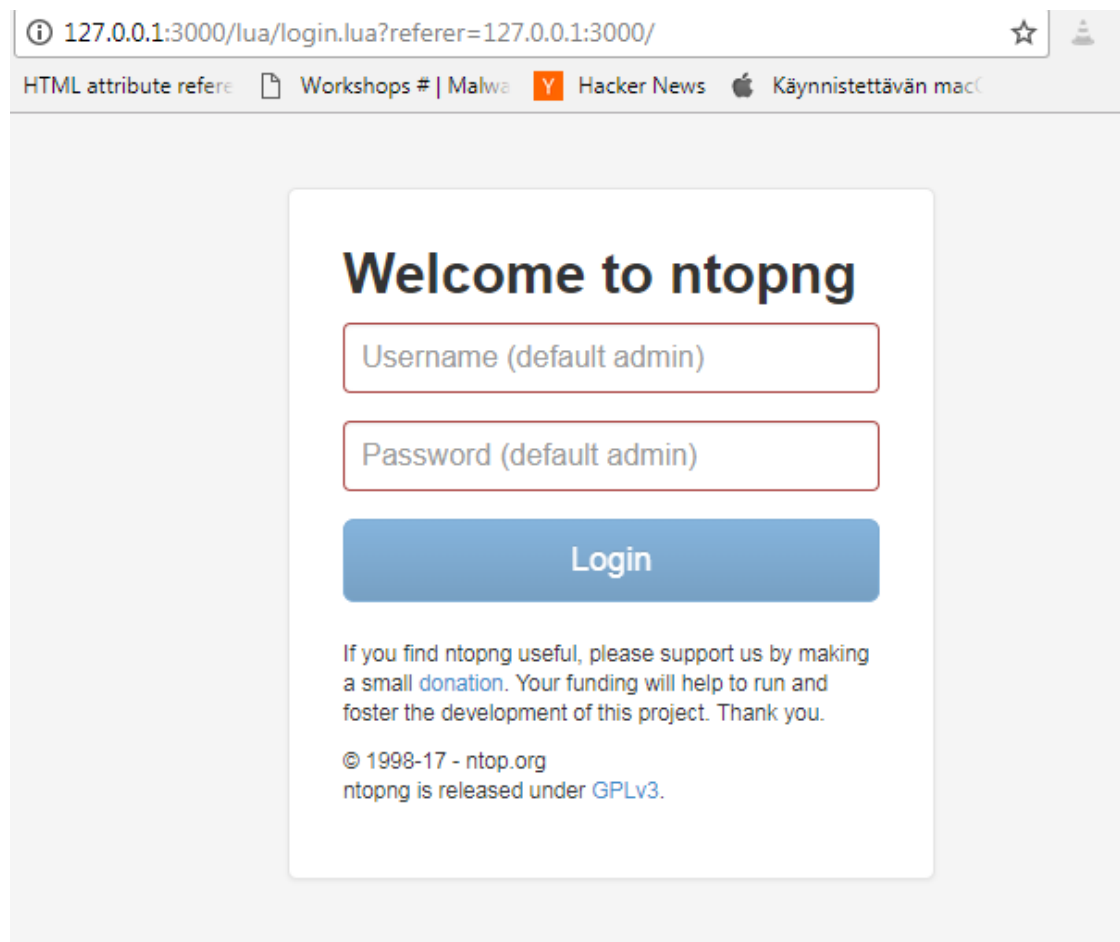
==> elasticsearch: Importing base box 'ubuntu/xenial64'...
==> elasticsearch: Matching MAC address for NAT networking...
==> elasticsearch: Checking if box 'ubuntu/xenial64' is up to date...
==> elasticsearch: A newer version of the box 'ubuntu/xenial64' for provider 'virtualbox' is
==> elasticsearch: available! You currently have version '20180306.0.0'. The latest is version
==> elasticsearch: '20180313.0.0'. Run 'vagrant box update' to update.
==> elasticsearch: Setting the name of the VM: Vagrant_elasticsearch_1520965635192_18109
==> elasticsearch: Fixed port collision for 22 => 2222. Now on port 2200.
==> elasticsearch: Clearing any previously set network interfaces...
==> elasticsearch: Preparing network interfaces based on configuration...
elasticsearch: Adapter 1: nat
elasticsearch: Adapter 2: hostonly
==> elasticsearch: Forwarding ports...
elasticsearch: 9200 (guest) => 9200 (host) (adapter 1)
elasticsearch: 22 (guest) => 2200 (host) (adapter 1)
==> elasticsearch: Running 'pre-boot' VM customizations...
==> elasticsearch: Booting VM...
==> elasticsearch: Waiting for machine to boot. This may take a few minutes...
elasticsearch: SSH address: 127.0.0.1:2200
elasticsearch: SSH username: vagrant
elasticsearch: SSH auth method: private key
elasticsearch: Warning: Connection reset. Retrying...
elasticsearch: Warning: Connection aborted. Retrying...
elasticsearch: Warning: Connection reset. Retrying...
elasticsearch: Warning: Connection aborted. Retrying...
elasticsearch: Vagrant insecure key detected. Vagrant will automatically replace
elasticsearch: this with a newly generated keypair for better security.
elasticsearch: Inserting generated public key within guest...
elasticsearch: Removing insecure key from the guest if it's present...
elasticsearch: Key inserted! Disconnecting and reconnecting using new SSH key...
==> elasticsearch: Machine booted and ready!
==> elasticsearch: Checking for guest additions in VM...
elasticsearch: The guest additions on this VM do not match the installed version of
elasticsearch: VirtualBox! In most cases this is fine, but in rare cases it can
elasticsearch: prevent things such as shared folders from working properly. If you see
elasticsearch: shared folder errors, please make sure the guest additions within the
elasticsearch: virtual machine match the version of VirtualBox you have installed on
elasticsearch: your host and reload your VM.
elasticsearch: Guest Additions Version: 5.0.40
elasticsearch: VirtualBox Version: 5.2
==> elasticsearch: Configuring and enabling network interfaces...
==> elasticsearch: Mounting shared folders...
elasticsearch: /vagrant => D:/Vagrant
==> elasticsearch: Running provisioner: file...
==> elasticsearch: Running provisioner: shell...
elasticsearch: Running: inline script
==> elasticsearch: Running provisioner: shell...
elasticsearch: Running: inline script
elasticsearch: msg: ttyname failed: Inappropriate ioctl for device
elasticsearch: sudo
elasticsearch: :
elasticsearch: unable to resolve host ubuntu-xenial
==> elasticsearch: Running provisioner: docker...
==> elasticsearch: Installing Docker onto machine...
==> elasticsearch: Starting Docker containers...
==> elasticsearch: -- Container: jarmondocker.com:443/es-engine-image

```

Kuvio 18 Vagrant up elasticsearch

6 Palvelun testaus

Kun vagrant on luonut ja provisioinut koneet voimme testata palvelun toimivuutta menemässä isäntäkoneen selaimella osoitteen 127.0.0.1:3000. Jos porttiohjaukset ovat onnistuneet ja kontti on ylhäällä ntopng:n kirjautumisikkuna pitäisi näkyä selaimessa(Kuvi 19 Vagrant ntopng test)



Kuvio 19 Vagrant ntopng test

Kun ntopng:n on todettu toimivaksi, voimme tutkia elasticsearchin toimivuutta. Sen voi todeta menemällä ntopng:n hallintanäkymässä flows välilehdelle, josta voidaan katsoa, minne flow dataa tallennetaan. "Server" kohdassa on elasticsearchin IP-osoite joten voimme todeta että ainakin ntop lähettää dataa palvelimelle(Kuvi 20 Vagrant ntopng flows)

Active Flows

	Application	L4 Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info
Info	HTTP	TCP	f8cffe857873 42798	192.168.99.20:9200	0 sec	Client Serv	0 bps	9.55 KB	/_bulk
Info	HTTP	TCP	f8cffe857873 42796	192.168.99.20:9200	0 sec	Client Serv	0 bps	9.35 KB	/_bulk
Info	HTTP	TCP	10.0.2.2:56460	127.0.0.1:3000	1 sec	Client Server	0 bps	2.65 KB	/lua/network_load.lua

Kuvio 20 Vagrant ntopng flows

Lopullisen varmistuksen saamme tutkimalla verkkoliikennekaappausta elasticsearch koneella. tcpdump liikennekaappauksesta voidaan huomata että dataa tulee ntopng:n osoitteesta 192.168.99.10 dockerkontin porttiin 9200(Kuvio 21 Vagrant elasticsearch tcpdump)

```

19:29:00.115343 IP 172.17.0.2.9200 > 192.168.99.10.50034: Flags [..], ack 140, win 235, options [nop,nop,TS val 68732 ecr 109653], length 0
19:29:00.170832 IP 172.17.0.2.9200 > 192.168.99.10.50034: Flags [P..], seq 1:26, ack 140, win 235, options [nop,nop,TS val 68746 ecr 109653], length 25
19:29:00.171223 IP 192.168.99.10.50034 > 172.17.0.2.9200: Flags [..], ack 26, win 229, options [nop,nop,TS val 109667 ecr 68746], length 0
19:29:00.171453 IP 192.168.99.10.50034 > 172.17.0.2.9200: Flags [P..], seq 140:7313, ack 26, win 229, options [nop,nop,TS val 109667 ecr 68746], length 7173
19:29:00.171508 IP 172.17.0.2.9200 > 192.168.99.10.50034: Flags [..], ack 7313, win 347, options [nop,nop,TS val 68746 ecr 109667], length 0
19:29:00.218068 IP 172.17.0.2.9200 > 192.168.99.10.50034: Flags [P..], seq 26:1614, ack 7313, win 347, options [nop,nop,TS val 68758 ecr 109667], length 1588
19:29:00.218795 IP 192.168.99.10.50034 > 172.17.0.2.9200: Flags [..], ack 1614, win 253, options [nop,nop,TS val 109679 ecr 68758], length 0
19:29:00.218811 IP 192.168.99.10.50034 > 172.17.0.2.9200: Flags [F..], seq 7313, ack 1614, win 253, options [nop,nop,TS val 109679 ecr 68758], length 0
19:29:00.219141 IP 172.17.0.2.9200 > 192.168.99.10.50034: Flags [F..], seq 1614, ack 7314, win 347, options [nop,nop,TS val 68758 ecr 109679], length 0
19:29:00.219359 IP 192.168.99.10.50034 > 172.17.0.2.9200: Flags [..], ack 1615, win 253, options [nop,nop,TS val 109679 ecr 68758], length 0

```

Kuvio 21 Vagrant elasticsearch tcpdump

7 Pohdinta

Harjoituksen tavoitteisiin päästiin ja palvelu saatiin toimimaan moitteetta. Muutamia ongelmia oli docker parametrien kanssa mutta lopulta kaikki saatiin toimimaan. Vagrant on nopea tapa luoda usean koneen devausympäristö helposti. Kunhan vagrantfile on mukana niin saat ympäristön pystyyn missä tahansa resurssit ovat kohdilla. Vagrant ei sovellu koko virtuaalikoneen automatisointiin sillä siinä on himena puutteita jos aikoo tehdä jotain kehittyneempää projektia. Harjoituksessa olisi vielä voinut tehdä ympäristömuuttuja vagrantfileen muutosten helpottamiseksi.

8 Lähteet

<https://www.vagrantup.com/docs>