jamk.fi

Labra 5

Palveluiden automatisointi

Ville Pulkkinen
Mikael Romanov
Markus Häkkinen

Kurssityö
Huhtikuu 2018
Tieto- ja viestintätekniikka
Tekniikan ja liikenteen ala

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

Sisällysluettelo

1 Johdanto 3

2 Ansible 3

2.1 Ansiblen asennus 3

3 Google palvelutunnuksen luominen 4

4 Koneiden luonti pilveen 5

4.1 SSH Avaimet 5

4.2 GCE Virtuaalikoneen luonti 6

4.3 SSH –Avaimen siirtäminen 7

4.4 Koneen luonti 8

5 Roolit 9

5.1 Docker 10

5.2 Ntopng 10

5.3 Elasticsearch 11

6 Testaus 11

7 Kuormantasaus 12

8 Pohdinta 16

9 Lähteet 16

Kuviot

Kuvio 1 Ansiblen asennus 3

Kuvio 2 Ansiblen versio 3

Kuvio 3 Pip asennus 4

Kuvio 4 Service tunnukset luominen 5

Kuvio 5 P12 muutos PEM 5

Kuvio 6 SSH Avain virtuaalikoneeseen 6

Kuvio 7 SSH Avaimen luonti 6
Kuvio 8 GCE Esimerkki 7
Kuvio 9 Ansible SSH avaimen siirto 7
Kuvio 10 SSH Avaimen siirto metadatassa 8
Kuvio 11 Main playbook 8
Kuvio 12 GCE Playbook 9
Kuvio 13 Docker rooli 10
Kuvio 14 Ntopng rooli 11
Kuvio 15 elasticsearch rooli 11
Kuvio 16 Ansiblen ajo 12
Kuvio 17 Toimiva palvelu 12
Kuvio 18 HAproxy rooli 13
Kuvio 19 Haproxy konfiguraatiopohja 14
Kuvio 20 Koneet pilvessä 14
Kuvio 21 HAproxy osoite 15
Kuvio 22 Python koodi 15
Kuvio 23 HAproxy statistiikka 15

1 Johdanto

Laboratorioharjoituksen tavoitteena oli luoda aikaisemmin tekemämme palvelut google pilveen, käyttäen ansiblea, joka on yleiskäyttöinen automatisointiohjelmisto. Palvelut tuli sisältää virtuaalikoneet ja aiemmin luomamme docker-kontit. Lopuksi palvelulle tehtiin kuormantasaus käyttäen HAproxyä.

2 Ansible

Ansible on yleiskäyttöinen automaatio-ohjelmisto, johon on rakennettu yli 1300 lisäosaa eri laite- ja ohjelmistovalmistajien kanssa. Ansiblella voidaan automatisoida provisiointi, konfiguraatioiden hallinta ja sovelluksien kehitys. Ansible tarvitsee toimiakseen käyttöjärjestelmältä vain python ohjelmointikielen ja pip pakettien hallinnan asennettuna.

2.1 Ansiblen asennus

Ansible voidaan asentaa mille tahansa UNIX jakelulle jossa on asennettuna python 2.6 tai uudempi. Ansiblen asennus on yksinkertainen prosessi. Uusimman Ansible version asentaminen vaatii Ansiblen repositoryn asentamisen (Kuvio 1)

```
1  $ sudo apt-get update
2  $ sudo apt-get install software-properties-common
3  $ sudo apt-add-repository ppa:ansible/ansible
4  $ sudo apt-get update
5  $ sudo apt-get install ansible
```

Kuvio 1 Ansiblen asennus

Ansible versiota voidaan tarkastella komennolla ”Ansible –version” (Kuvio 2). Osa käytettävistä moduuleista ovat vanhentuneita mutta Ansible osaa varoittaa niistä.

```
k1521@service-vm-4:~$ ansible --version
ansible 2.4.3.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/k1521/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.12 (default, Dec  4 2017, 14:50:18) [GCC 5.4.0 20160609]
```

Kuvio 2 Ansiblen versio

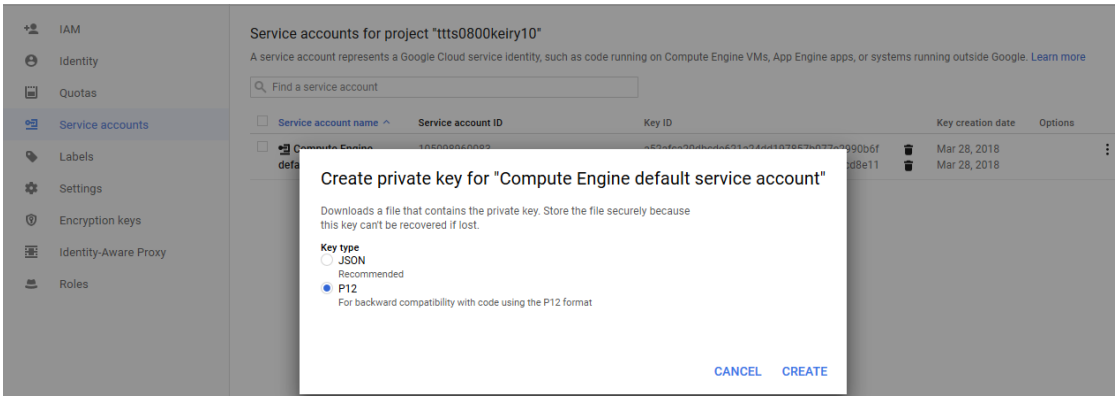
Koska Ansiblen moduulit ovat tehty python ohjelmointikielellä tarvitsemme hallintakoneellemme vielä pythonin pakettienhallintaohjelmisto ”PIP”:in. PIP voidaan asentaa Linuxin pakettien hallinnasta komennolla ”apt-get install python-pip” (Kuvio 3). PIP-pakettien asennus tapahtuu komennolla ”pip install paketin_nimi”.

```
k1521@service-vm-4:/etc/ansible$ sudo apt-get install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  python-pip
0 upgraded, 1 newly installed, 0 to remove and 27 not upgraded.
Need to get 0 B/144 kB of archives.
After this operation, 635 kB of additional disk space will be used.
Selecting previously unselected package python-pip.
(Reading database ... 76560 files and directories currently installed.)
Preparing to unpack .../python-pip_8.1.1-2ubuntu0.4_all.deb ...
Unpacking python-pip (8.1.1-2ubuntu0.4) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up python-pip (8.1.1-2ubuntu0.4) ...
k1521@service-vm-4:/etc/ansible$ pip install apache-libcloud
Collecting apache-libcloud
  Downloading apache_libcloud-2.3.0-py2.py3-none-any.whl (4.9MB)
    100% |[redacted]| 4.9MB 244kB/s
Collecting requests (from apache-libcloud)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% |[redacted]| 92kB 10.5MB/s
Collecting certifi>=2017.4.17 (from requests->apache-libcloud)
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
    100% |[redacted]| 153kB 7.5MB/s
```

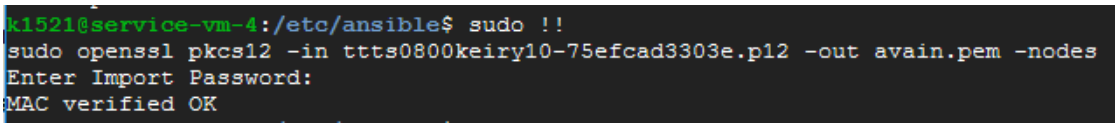
Kuvio 3 Pip asennus

3 Google palvelutunnuksen luominen

Harjoitustyössä virtuaalikoneet tuli luoda googlen pilveen käyttäen ansiblea. Ansiblessa on GCE moduuli, jolla voidaan käskeä googlen GCE rajapintaa luomaan haluamme koneet. Pilvipalveluun pitää luoda ”service account” jolla hallitaan pääsyä GCE:n API(Application Programming Intenrface) rajapintaan(Kuvio 3). Tunnuksen luomisen jälkeen GCE generoi yksityisen avaimen joka lähetetään API:lle tunnistautumisen yhteydessä. Käytimme P12 tiedostoa, jonka muutimme ”pem” tiedostoksi Ansible koneellamme (Kuvio 4)



Kuvio 4 Service tunnukset luominen



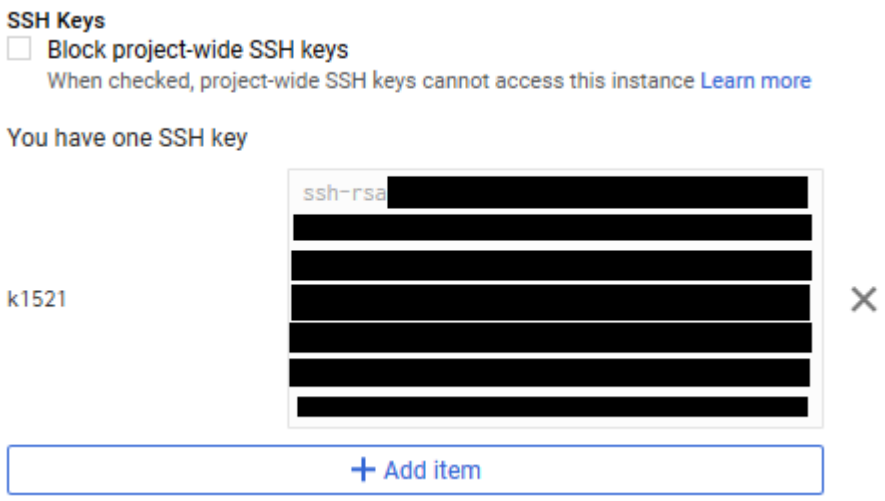
Kuvio 5 P12 muutos PEM

4 Koneiden luonti pilveen

Googlen pilven käskyttäminen vaatii ansibleen ”apache-libcloud” moduulin joka on yli 50 pilvipalvelun käskyttämiseen tehty moduuli. Moduuli voidaan asentaa pipill komennolla ”pip install apache-libcloud”.

4.1 SSH Avaimet

Google käyttää SSH-avaimia yhteyksien autentikoimisessa. Omat privaatit SSH avaimet voidaan asentaa googlen virtuaalikoneisiin suoraan käyttöliittymän kautta.



Kuvio 6 SSH Avain virtuaalikoneeseen

Luomme ansiblea varten oman julkisen avaimen ja yksityisen ssh-rsa avaimen joka siirretään myöhemmin käskytettäviin virtuaalikoneisiin (Kuvio 7)

```
k1521@service-vm-4:/etc/ansible$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/k1521/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/k1521/.ssh/id_rsa.
Your public key has been saved in /home/k1521/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:juzq2nuehtiIu3H4gXAP833J3iFHc0jM8bYU3U4I4mE k1521@service-vm-4
The key's randomart image is:
+---[RSA 2048]----+
|          E oo o |
|         = = .o o|
|          = +  o |
|         . + .  .|
|. +     S + o    |
|.+ = o + o o     |
|+.o+o.+ * o      |
|.+ooo.o+ + .     |
|oooo=*+ . .      |
+----[SHA256]-----+
```

Kuvio 7 SSH Avaimen luonti

4.2 GCE Virtuaalikoneen luonti

Koneenluonti playbookissa on ensin määritelty API rajapinnalle asetukset ja sertifikaatit. Tulevien koneiden asetukset on myös tehty muuttujiin muokattavuuden helpottamiseksi. Esimerkissä oleva playbook luo pilveen yhden koneen jonka nimi on service-vm-5.

```
  GNU nano 2.5.3                File: koneenluonti

- name: Create instance(s)
  hosts: localhost
  connection: local
  gather_facts: no

  vars:
    service_account_email: 105098960083-compute@developer.gserviceaccount.com
    credentials_file: /etc/ansible/serti.crt
    project_id: ttts0800keiry10
    machine_type: f1-micro
    disk: pd-ssd
    disk_size: 20
    image: ubuntu-1604-xenial-v20180306
    timezone: europe-west4-a

  tasks:

   - name: Launch instances
     gce:
         instance_names: service-vm-5
         machine_type: "{{ machine_type }}"
         image: "{{ image }}"
         disk_type: "{{ disk }}"
         disk_gb: "{{ disk_size }}"
         service_account_email: "{{ service_account_email }}"
         credentials_file: "{{ credentials_file }}"
         project_id: "{{ project_id }}"
         zone: "{{ timezone }}"
```

Kuvio 8 GCE Esimerkki

4.3 SSH –Avaimen siirtäminen

Aikaisemmin luotu SSH-avain pitää siirtää virtuaalikoneelle ansiblella. Avaimen voisi siirtää etukäteen mutta automatisoinnista ei olisi tässä kohtaa enään niin suurta hyötyä. SSH avain voidaan siirtää käyttäen ansiblen muuttujia (Kuvio 9)

```
- name: Lue avain
  command: cat ~/.ssh/id_rsa.pub
  register: ssh_key
- debug:
    msg: " {{ ssh_key.stdout }} "
- name: aseta avain
  set_fact: sshkey="{{ssh_key.stdout}}"
```

Kuvio 9 Ansible SSH avaimen siirto

SSH-avain siirretään luodun virtuaalikoneen metadaan jolloin se lisää itsensä automaattisesti tunnettujen avainten listalle (Kuvio 10) Avainten asentamisen jälkeen Ansible voi ottaa SSH-yhteyden kohdekoneisiin.

```
    - name: Launch instance(s)
      gce:
          #disk_auto_delete: true
          #disks:
          #   - name: boot-disk
          #     mode: READ_WRITE
          instance_names: "{{ names }}"
          machine_type: "{{ machine_type }}"
          service_account_email: "{{ service_account_email }}"
          credentials_file: "{{ credentials_file }}"
          project_id: "{{ project_id }}"
          zone: "{{ timezone }}"
          state: "{{ vm_state }}"
          metadata: '{"sshKeys":"root:{{sshkey}}"}'
      register: gce
```

Kuvio 10 SSH Avaimen siirto metadatassa

4.4 Koneen luonti

Loimme ansibleen ”main” playbookin johon on sisällytetty muita playbookeja ja rooleja. Tiedostossa on määritetty kohteiden nimet muuttujiin, joiden perusteella virtuaalikoneet luodaan ja lisätään ryhmiin. Kun koneet on luotu, voidaan niille asentaa rooleja. Molemmille koneille asennetaan rooli docker.

```
ubuntu@ubuntu-vm:~/ansible$ cat pululeiska.yml
---
- include: google/luo_kone_pulu.yml names=elastic1 private=elastic-private
- include: google/luo_kone_pulu.yml names=ntopng1,ntopng2 private=ntopng-private
- include: google/luo_kone_pulu.yml names=haproxy private=haproxypriv
- name: ASENNA NTOPNG
  vars:
    ip: "{{ private['elastic-private'][0] }}"
  hosts: ntopng1
  become: yes
  roles:
     - { role: docker }
     - { role: ntopng }
 #    - docker
 #    - ntopng

- name: ASENNA ELASTIC
  hosts: elastic1
  become: yes
  roles:
     - docker
     - elastic

- name: ASENNA HAPROXY
  hosts: haproxy
  become: yes
  roles:
     - haproxy
```

Kuvio 11 Main playbook

Ansiblen GCE moduulilla voidaan määritellä lähes kaikki luotavien koneiden asetukset. Luotavan koneen nimi on määritelty names muuttujaan, joka asetettiin edellisessä playbookissa. Koneen luonnin jälkeen sen julkinen IP-osoite lisätään saman-nimiseen ryhmään kuin on koneen nimi. Ryhmiä voidaan myöhemmin käyttää roolien asennuksessa. Lopuksi ansible ottaa SSH-yhteyden koneeseen (Kuvio 12)

```
- name: Launch instance(s)
  gce:
      #disk_auto_delete: true
      #disks:
      #  - name: boot-disk
      #    mode: READ_WRITE
      instance_names: "{{ names }}"
      machine_type: "{{ machine_type }}"
      service_account_email: "{{ service_account_email }}"
      credentials_file: "{{ credentials_file }}"
      project_id: "{{ project_id }}"
      image: "{{ image }}"
      zone: "{{ timezone }}"
      state: "{{ vm_state }}"
      metadata: '{"sshKeys":"ubuntu:{{sshkey}}"}'
  register: gce

- name: Save host data
  add_host:
    hostname: "{{ item.public_ip }}"
    groupname: "{{ names }}"
  with_items: "{{ gce.instance_data }}"

- name: Wait for SSH for instances
  wait_for:
    delay: 20
    host: "{{ item.public_ip }}"
    port: 22
    state: started
    timeout: 60
  with_items: "{{ gce.instance_data }}"
```

Kuvio 12 GCE Playbook

5 Roolit

Ansiblessa voidaan tehdä rooleja, joita voidaan asentaa määrätyille kohteille. Palveluumme luotiin 2 virtuaalikonetta joille molemmille asennettuun Docker. Dockerin asennuksesta tehtiin oma rooli.

5.1 Docker

Docker rooli asentaa dockerin ja oman docker rekisterimme sertifikaatin koneelle. Roolin luomisessa on käytetty ansiblen moduuleja APT, SHELL ja GET_URL.

```
- name: Install docker
  become: yes
  apt:
    name: "{{ item }}"
    state: present
    install_recommends: False
  with_items:
    - "apt-transport-https"
    - "ca-certificates"
    - "software-properties-common"
    - "curl"
    - "ca-certificates"
- name: Kurlaa avain
  shell: "sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -"

- apt_repository:
    repo: "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable"
    update-cache: yes
- name: Update cache
  apt:
    #name: docker-ce
    update_cache: yes
- name: Install docker-ce
  apt:
    name: docker-ce
- name: Asenna pip
  apt:
    name: python-pip
- name:
  shell: pip install docker-py

- name: Create certs folder
  shell: mkdir -p /etc/docker/certs.d/pulunpurkki.com:443/
  register: mkdir
- debug: msg="{{ mkdir.stdout }}"

- name: ADD REG IP
  shell: echo "35.204.126.93 pulunpurkki.com" >> /etc/hosts

- name: Hae sertti
  get_url:
    url: http://student.labranet.jamk.fi/~K1521/ca.crt
    dest: /etc/docker/certs.d/pulunpurkki.com:443/

- name: RESTART DOCKER
```

Kuvio 13 Docker rooli

5.2 Ntopng

Ntopng rooli käyttää ansiblen docker moduulia. Docker rooli asensi koneille jo rekisterimme sertifikaatit, joten voimme hakea kontit suoraan sieltä. Ntopng tarvitsee yhden muuttujan, joka on elasticsearchin private-ip.

```
ubuntu@ubuntu-vm:~/ansible$ cat roles/ntopng/tasks/main.yml
- name: Kontti UP
  docker_container:
    name: ntopng
    image: pulunpurkki.com:443/ntopng-image
    ports:
     - "3000:3000"
    command: "-F 'es;flows;ntopng-%Y.%m.%d;http://{{ groups[\'elastic-private\'][0] }}:9200/bulk;'"

ubuntu@ubuntu-vm:~/ansible$
```

Kuvio 14 Ntopng rooli

5.3 Elasticsearch

Elasticsearch kontin ei tarvitse kuin kuunnella portteja 9200 ja 9300.

```
  GNU nano 2.5.3          File: roles/elastic/tasks/main.yml

- name: Kontti UP
  docker_container:
    name: ntopng
    image: pulunpurkki.com:443/es-engine-image
    ports:
     - 9200:9200
     - 9300:9300
```

Kuvio 15 elasticsearch rooli

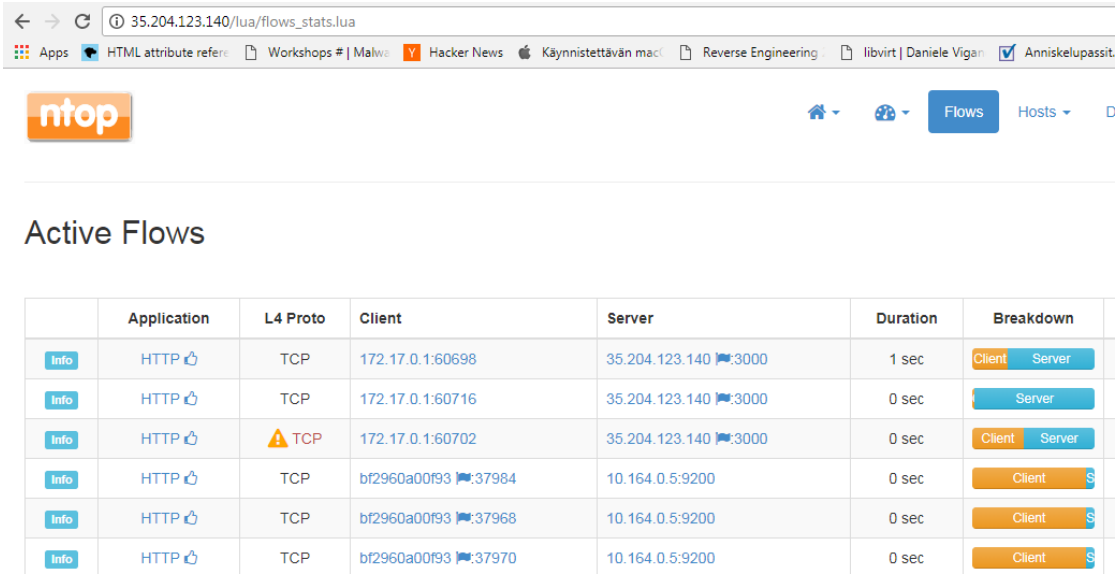
6 Testaus

Ajamalla oman ”main” playbookimme ansible luo halutut virtuaalikoneet google pilveen, ja asentaa koneille tarvittavat roolit. Onnistunut asennus todettiin menemällä ntopng koneen IP-osoitteeseen selaimella joka avaa ntopng kirjautumis-ikkunan.

```
TASK [docker : Hae sertti] ***********************************************
*****************************************************************************
********
ok: [35.204.199.24]

TASK [docker : RESTART DOCKER] *******************************************
*****************************************************************************
********
changed: [35.204.199.24]

TASK [elastic : Kontti UP] ***********************************************
*****************************************************************************
********
changed: [35.204.199.24]

PLAY RECAP ***************************************************************
*****************************************************************************
********
35.204.123.140             : ok=14   changed=7    unreachable=0    failed=0
35.204.199.24              : ok=14   changed=7    unreachable=0    failed=0
localhost                  : ok=14   changed=6    unreachable=0    failed=0

ubuntu@ubuntu-vm:~/ansible$
```

Kuvio 16 Ansiblen ajo

Elasticsearch kontin toiminta voidaan todeta ntopng:n flows näkymästä josta nähdään että ntop lähettää dataa elastic koneelle onnistuneesti.



Kuvio 17 Toimiva palvelu

7 Kuormantasaus

Teimme palvelullemme vielä kuormantasauksen käyttäen HAproxy välityspalvelinta. HAproxylle tehtiin oma rooli, joka asentaa palvelun ja generoi konfiguraation käyttäen jinja2 template pohjaa (Kuvio 18). HAproxyn roolissa on määritelty

muuttujat jotka ansible syöttää templatelle luoden konfiguraatiotiedoston HAproxylle. Ansible lataa tiedoston kohdekoneelle. Tieto koneista joiden välillä kuormaa tasataan, on kerrottu ansiblen ”host” muuttujien avulla. Kaikki ntopng-private ryhmään kuuluvat koneet lisätään HAproxyn konfiguraation (Kuvio 19)

```
ubuntu@ubuntu-vm:~/ansible$ cat roles/haproxy/tasks/main.yml

- apt_repository:
    repo: "ppa:vbernat/haproxy-1.5"
    update-cache: yes

- name: APT UPDATE
  apt:
    update_cache: yes
    state: present
- name: DIST
  apt:
    upgrade: dist
- name: ASENNA HAPROXY
  apt:
    name: haproxy

- name: Update HAproxy config
  vars:
    haproxy_app_name: myapp
    haproxy_mode: http
    haproxy_enable_stats: enable
    haproxy_algorithm: roundrobin
    haproxy_backend_servers:
      - {name: ntopng1, ip: "{{ groups['ntopng-private'][0] }}", port: 80, paramstring: cookie A check}
      - {name: ntopng2, ip: "{{ groups['ntopng-private'][1] }}", port: 80, paramstring: cookie A check}
    haproxy_stats_users:
      - {username: rommi, password: rommi}

  template: src=templates/haproxy.cfg
        dest=/etc/haproxy/haproxy.cfg
        backup=yes
```

Kuvio 18 HAproxy rooli

Template käy läpi jinjan for loopilla läpi halutut hostit jolle kuormantasaus tehdään.

```
ubuntu@ubuntu-vm:~/ansible$ cat roles/haproxy/tasks/templates/haproxy.cfg
global
  log 127.0.0.1 local0 notice
  maxconn 2000
  user haproxy
  group haproxy

defaults
  log     global
  mode    http
  option  httplog
  option  dontlognull
  retries 3
  option redispatch
  timeout connect  5000
  timeout client  10000
  timeout server  10000

listen {{haproxy_app_name}}
  bind 0.0.0.0:80
  mode {{haproxy_mode}}
  stats {{haproxy_enable_stats}}
  {% if haproxy_enable_stats == 'enable' %}
  stats uri /haproxy?stats
  stats realm Strictly\ Private
  {% for user in haproxy_stats_users %}
  stats auth {{user.username}}:{{user.password}}
  {% endfor %}
  {% endif %}
  balance {{haproxy_algorithm}}
  option httpclose
  option forwardfor
  {% for server in haproxy_backend_servers %}
  server {{server.name}} {{server.ip}}:{{server.port}} {{server.paramstring}}
  {% endfor %}
ubuntu@ubuntu-vm:~/ansible$
```

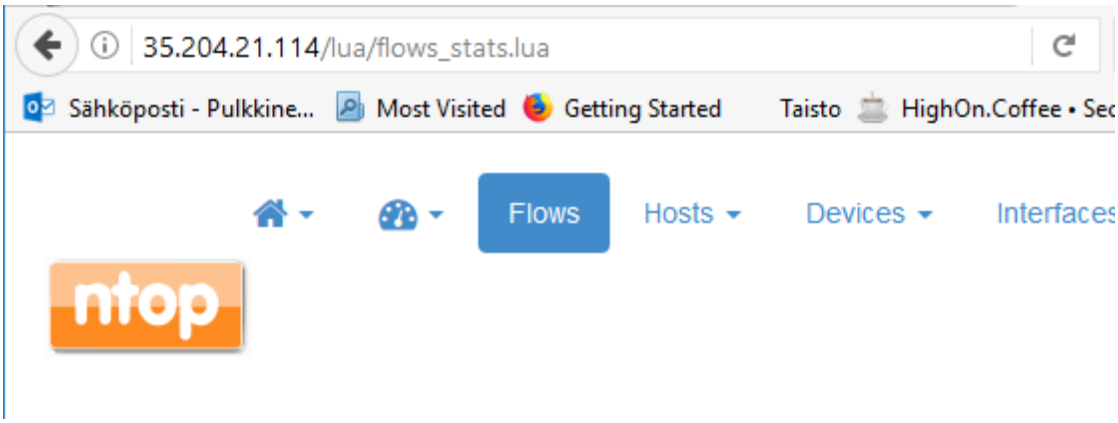
Kuvio 19 Haproxy konfiguraatiopohja

Kun kaikki halutut koneet ovat luotu pilveen voimme testata kuormantasausta yhdistämällä HAproxy koneen julkiseen IP-osoitteeseen selaimella (Kuvio 20)

Name	Zone	Recommendation	Internal IP	External IP	Connect
elastic	europe-west4-a		10.164.0.2	35.204.181.172	SSH
elastic1	europe-west4-a	Save $25 / mo	10.164.0.5	35.204.199.24	SSH
haproxy	europe-west4-a		10.164.0.7	35.204.21.114	SSH
ntopng	europe-west4-a	Increase perf.	10.164.0.3	35.204.239.67	SSH
ntopng1	europe-west4-a	Increase perf.	10.164.0.6	35.204.123.140	SSH
registery	europe-west4-a	Increase perf.	10.164.0.4	35.204.126.93	SSH

Kuvio 20 Koneet pilvessä

Jomman kumman ntopng koneen näkymä pitäisi ilmestyä ruudulle(Kuvio 21) Tarkempaa statistiikkaa voidaan katsoa HAproxy stats välilehdeltä osoitteesta ”35.204.21.114/haproxy?stats”(Kuvio 23)

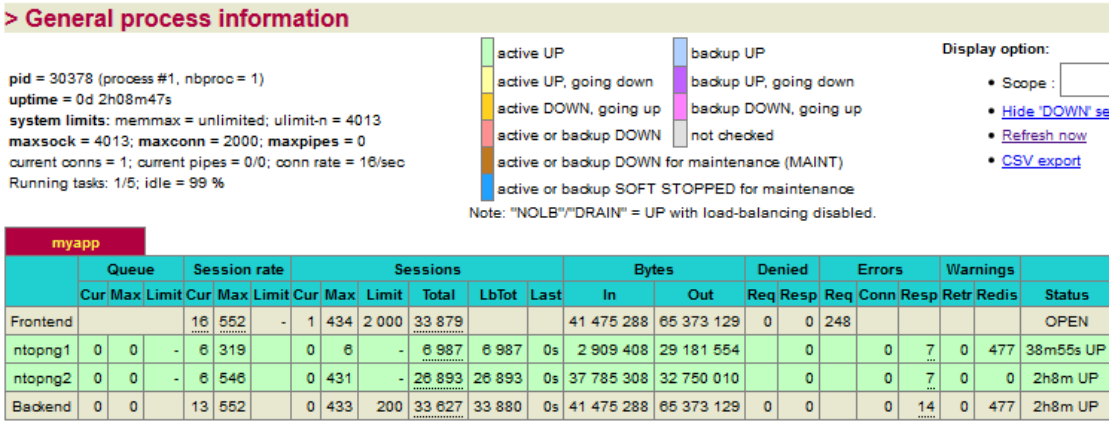


Kuvio 21 HAproxy osoite

Testasimme kuormantasausta tekemällä HTTP pyyntöjä pythonilla (Kuvio 22), samaan aikaa HAproxyn statistiikkaa seuraamalla todettiin, että kuorma jaettiin molemmille palvelimille tasaisesti.

```
import urllib2
urli = "http://35.204.21.114"
import time
for x in range(0,1000):
        urllib2.urlopen(urli)
```

Kuvio 22 Python koodi



> General process information

pid = 30378 (process #1, nbproc = 1)
uptime = 0d 2h08m47s
system limits: memmax = unlimited; ulimit-n = 4013
maxsock = 4013; maxconn = 2000; maxpipes = 0
current conns = 1; current pipes = 0/0; conn rate = 16/sec
Running tasks: 1/5; idle = 99 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
• Scope :
• Hide 'DOWN' se
• Refresh now
• CSV export

myapp

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status
Frontend				16	552	-	1	434	2 000	33 879			41 475 288	65 373 129	0	0	248					OPEN
ntopng1	0	0	-	6	319		0	6	-	6 987	6 987	0s	2 909 408	29 181 554		0		0	7	0	477	38m55s UP
ntopng2	0	0	-	6	546		0	431	-	26 893	26 893	0s	37 785 308	32 750 010		0		0	7	0	0	2h8m UP
Backend	0	0		13	552		0	433	200	33 627	33 880	0s	41 475 288	65 373 129	0	0		0	14	0	477	2h8m UP

Kuvio 23 HAproxy statistiikka

8 Pohdinta

Ansible vaikuttaa todella tehokkaalta automaatiotyökalulta ja se varmasti helpottaa asioiden tekemistä suuremmissa projekteissa. Googlen pilvi ei vakuuttanut ollenkaan, se on vielä keskeneräinen ja todella kankea verrattavissa kilpailijoihin. Saimme luotua kaikki koneet pilveen ja asennettua niille rooleja, mutta ansiblesta jäi vielä paljon asioita mitkä olisi voinut tehdä paremmin. Opimme HAproxyn perusteet mikä tuli myös uutena asiana.

9 Lähteet

Hochstein, L. 2012. Ansible UP & Running. Automating configuration management and deployment the easy way. O'Reilly. Viitattu 25.4.2018