

## Pilvipalvelut

### OpenStack harjoitustyö

#### Ryhmä 3

Alexander Andreev

Rami Ojala

Ilari Rajala

Asko Ropponen

Harjoitustyö

TTTW0430 - Pilvipalvelut, Jarmo Viinikanoja

9.11.2019

Tieto- ja viestintätekniikan tutkinto-ohjelma

# Sisältö

<b>1</b>	<b>Johdanto .....</b>	<b>1</b>
<b>2</b>	<b>Tehtävänanto .....</b>	<b>1</b>
<b>3</b>	<b>Teoreettiset lähtökohdat .....</b>	<b>1</b>
3.1	OpenStackin komponentteja.....	2
3.1.1	Compute .....	2
3.1.2	Dashboard .....	2
3.1.3	Image Service .....	2
3.1.4	Object Storage.....	3
3.1.5	Identity Service.....	3
3.1.6	Networking.....	3
3.1.7	Block Storage.....	4
3.1.8	Telemetry .....	4
3.1.9	Orchestration .....	5
<b>4</b>	<b>Stack.sh Skripti .....</b>	<b>5</b>
<b>5</b>	<b>Työn kulku.....</b>	<b>6</b>
5.1	Käyttäjän luominen .....	6
5.2	DevStacking asennus .....	7
5.3	DevStack:in ajaminen ja hallinnointi. ....	9
5.4	Virtuaalikoneen lisääminen OpenStackissa.....	10
5.4.1	Levykuvan lisäys .....	10
5.4.2	Instanssin luominen .....	11
<b>6</b>	<b>Pohdinta .....</b>	<b>16</b>
	<b>Lähteet .....</b>	<b>17</b>

# 1 Johdanto

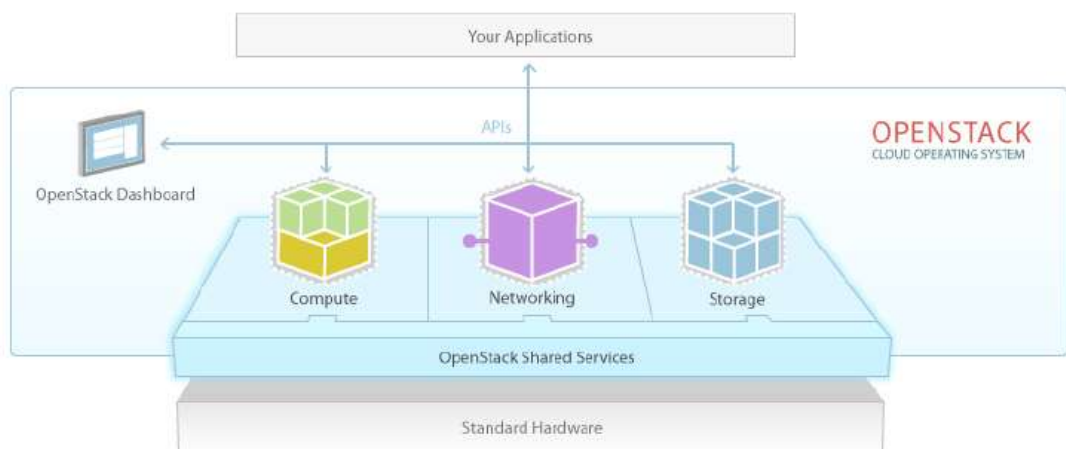
Pilvipalvelut harjoitustyönä tehtiin ennalta määritettyjen ryhmien kesken DevStackin pystytys virtuaalikoneelle ja perehdyttiin OpenStackin erilaisiin komponentteihin. Työssä käytettiin harjoitustyön ohjetta, opettaja Viinikanojan asiantuntemusta ja OpenStackista löytyvää dokumentaatiota.

## 2 Tehtävänanto

Harjoitustyössä tutustutaan OpenStack-alustaan ja sen komponentteihin. Työssä myös asennetaan DevStack ja otetaan virtuaalikone käyttöön DevStackin päälle.

## 3 Teoreettiset lähtökohdat

Vuonna 2010 Rackspace Hosting ja NASA tekivät yhdessä aloitteen avoimenlähdekoodin pilvipalvelusta, joka tunnetaan OpenStackinä. Tarkoituksena oli luoda julkisen ja yksityisen pilvien tarpeita vastaava alusta koosta riippumatta, joka on helposti implementoitava ja massiivisesti skaalautuva. OpenStackiä käytetään pohjana monessa itse tehdyssä pilvessä. Sitä käytetään sekä julkisissa että yksityisissä pilvissä.



Kuvio 1. OpenStack komponentteja.

## 3.1 OpenStackin komponentteja

### 3.1.1 Compute

Nova on OpenStack-projekti, joka hallitsee tietokone resursseja ja laskee instasseja. Nova tukee virtuaalikoneiden, baremetal-palvelimien luomista (ironic), ja se tukee järjestelmäkонтteja.

Nova pyörii daemoneina olemassa olevien Linux-palvelimien päällä palvelua tarjotakseen. Novan arkkitehtuuri tarjoaa joustavuuden pilven suunnittelussa ilman omistusoikeuden alaisia ohjelmistoja tai laitteistovaatimuksia ja tarjoaa myös kyvyn integroida vanhoja järjestelmiä ja kolmansien osapuolien tuotteita.

Nova voidaan ottaa käyttöön hypervisor-tekniikoilla, kuten KVM, VMware, LXC, XenServer, jne. (OpenStack: Core Components, OpenStack Services).

### 3.1.2 Dashboard

Horizon-projekti on OpenStack'in hallintapaneelin toteutus, joka on verkkopohjainen graafinen käyttöliittymä OpenStack-palveluille, jolla voi automatisoida pilvipohjaisia resursseja. Palveluntarjoajille ja muille kaupallisille myyjille se tukee kolmansien osapuolien palveluita kuten seuranta, laskutusta ja muita hallintatyökaluja.

Kehittäjät voivat automatisoida työkaluja hallitsemaan OpenStack-resursseja EC2-yhteensopivuusrajapinnan tai alkuperäisen OpenStack-sovellusliittymän avulla (OpenStack: Core Components, OpenStack Services).

### 3.1.3 Image Service

Glance-projekti tarjoaa palvelun, joka mahdollistaa virtuaalikoneiden kuvien löytämisen, rekisteröinnin ja palauttamisen, jotka on tarkoitettu käytettäväksi muiden palveluiden kanssa. Glance:ssa on asiakaspalvelin-arkkitehtuuri ja se toimittaa käyttäjän REST API:n, joka mahdollistaa virtuaalikoneen kuvan metatietojen kyselyn ja myös todellisen kuvan haun.

Glance tukee Raw, VirtualBox (VDI), VMWare (VMDK, OVF), Hyper-V (VHD) ja Qemu / KVM (qcow2) virtuaalikonekuvia (OpenStack: Core Components, OpenStack Services).

### 3.1.4 Object Storage

Swift-projekti, tarjoaa pilvitallennusohjelmistoja, jolla voidaan tallentaa ja hakea paljon tietoja yksinkertaisella API:lla. Sillä on hajautettu arkkitehtuuri, joka on mitoitettu ja optimoitu kestävyiden, saatavuuden ja samanaikaisuuden suhteen ilman keskitettyä hallintapistettä.

Swiftiä käytetään skaalautuvasti varastoimaan petatavuja käytettävää tietoa, ja se on ihanteellinen sellaisen jäsentelemättömän tiedon tallentamiseen, joka voi kasvaa ilman rajoja. Tallennettuja tietoja voidaan käyttää, hakea ja päivittää (OpenStack: Core Components, OpenStack Services).

### 3.1.5 Identity Service

Keystone on identiteettipalvelu, jota OpenStack käyttää todennukseen (authN) ja korkean tason valtuutukseen (authZ). Se integroituu olemassa oleviin taustapalveluihin, kuten LDAP, samalla kun se toimii yhteisenä todennusjärjestelmänä pilvipalvelujärjestelmän kautta.

Keystone tarjoaa myös keskitetyn luettelon käyttäjistä ja OpenStack-palveluista, joihin he pääsevät. Lisäksi se tarjoaa endpoint-rekisterin, joka on hakuja tukeva lista, josta löytyy tiedot OpenStack-pilvessä käytettävistä palveluista. Keystone tukee tällä hetkellä token-pohjaista authN ja user-service valtuutusta (OpenStack: Core Components, OpenStack Services).

### 3.1.6 Networking

Neutron-projekti tarjoaa verkko-ominaisuuksia, kuten verkkojen ja IP-osoitteiden hallintaa OpenStackille. Neutronin tehtävä on varmistaa, ettei verkko ole rajoittava te-

kijä pilvien käyttöönotossa, ja tarjoaa käyttäjille itsepalvelukyvyyn verkon kokoonpanoissa. Se tarjoaa myös NaaS (Network as a Service) palvelua käyttöliittymälaitteiden välillä, joita hallitsevat muut OpenStack-palvelut. OpenStack-verkkotoiminnon avulla käyttäjät voivat luoda omia verkkojaan ja yhdistää laitteita ja palvelimia yhteen tai useampaan verkkoon.

Neutron tarjoaa myös laajennus frameworkin, joka tukee muiden verkkopalveluiden, kuten virtuaalisten yksityisten verkkojen (VPN), palomuurien, kuorman tasausta ja tunkeutumisen havaitsemisjärjestelmän (IDS), käyttöönottoa ja hallintaa (OpenStack: Core Components, OpenStack Services).

### 3.1.7 Block Storage

Cinder-projekti on suunniteltu esittämään tallennusresursseja loppukäyttäjille, joita Nova-projekti (Compute) voi käyttää. Tämä tapahtuu käyttämällä joko vertailutoteutusta (LVM) tai laajennusohjaimia muihin varastoihin. Cinder virtualisoi Block Storage laitteiden hallinnan ja tarjoaa loppukäyttäjille itsepalvelu API:a pyytämään ja kuluttamaan näitä resursseja ilman, että vaaditaan mitään tietoa siitä, missä niiden varastointi oikeasti on käytössä tai minkä tyyppisellä laitteella.

Cinder voi käyttää tallennusaloja, kuten Linux-palvelin, EMC (ScaleIO, VMAX ja VNX), Ceph, Coraid, CloudByte, IBM, Hita-chi-tietojärjestelmät, SAN-äänenvoimakkuuden säädin jne. (OpenStack: Core Components, OpenStack Services).

### 3.1.8 Telemetry

Ceilometer-projekti on tiedonkeruupalvelu, joka tarjoaa yhden yhteispisteen laskutusjärjestelmille, jotka saavat kaikki mittaukset asiakkaan laskutuksen sallimiseksi kaikissa OpenStack-ydinkomponenteissa. Seuraamalla olemassa olevien palveluiden ilmoituksia, kehittäjät voivat kerätä tietoja ja määrittää tietotyyppin vastaamaan käyttövaatimuksiaan. Ceilometer luo myös mahdollisuuden normalisoida ja muuntaa dataa kaikissa nykyisissä OpenStack-ydinkomponenteissa (OpenStack: Core Components, OpenStack Services).

### 3.1.9 Orchestration

Heat on OpenStack Orchestration-ohjelman pääprojekti. Se laittaa orkestrointimootorin käynnistämään useita composite cloud sovelluksia, jotka perustuvat templateihin tekstitiedostojen muodossa, joita voidaan käsitellä kuten koodia.

Natiivi Heat-template on tällä hetkellä kehittymässä, mutta Heat pyrkii myös tarjoamaan yhteensopivuuden AWS CloudFormation template kanssa, jotta monia olemassa olevia CloudFormation templateja voidaan käynnistää OpenStackissa. Heat tarjoaa sekä OpenStack-natiivin ReST API:n, että myös CloudFormation -yhteensopivan Query API:n (OpenStack: Core Components, OpenStack Services).

## 4 Stack.sh Skripti

DevStackin asennuksessa käytettiin OpenStack sivustolta löytyvää ohjetta. Tässä kappaleessa kuvaillaan lyhyesti ohjeen repositoriossa olevaa stack.sh skriptiä. Skripti on noin 1500 riviä pitkä, joten tässä raapaistaan vain pintaa kevyesti.

Skriptin ajo virtuaalikoneella, jossa on 4096MB muistia ja kaksi ydintä kesti noin 30 minuuttia. Asennuksessa käytettiin repositorion oletus local.conf tiedostoa, joka löytyy devstack/samples kansioista.

Skripti asentaa OpenStack komponentit: Cinder, Glance, Horizon, Keystone, Nova, Neutron ja Swift. Nämä komponentit ovat kuvattu kappaleessa 3.1.

Skripti käyttää sisäisesti sudoa, joten sitä ei tarvitse ajaa sudo etuliitteellä, mutta on otettava huomioon, että stack käyttäjä on lisätty sudoers ryhmään. Skripti myös varoittaa käyttäjää, mikäli käyttäjä jona skripti ajetaan ei ole sudo ryhmässä. Skripti myös vaihtaa linuxin oletus unicode merkistö koodauksen, sekä kielen UTF-8 merkistökoodaukselle ja kieleksi en\_US (englanti USA).

Stack.sh skripti käyttää hyväkseen useita eri konfiguraatio tiedostoja, näitä ovat esim: stackrc, localrc ja local.conf tiedostot. Näistä skripti hakee esimerkiksi, LOGFILE muuttujan, joka kertoo minne skripti kirjoittaa logitiedostoa. Skripti kysyy sellaiset oleelliset asiat käyttäjältä, joita se tarvitsee toimiakseen, jos niitä ei ole asetustiedostoissa

määritelty. Esimerkiksi admin salasanan puuttuminen local.conf tiedostossa ei ole ongelma, sillä scripti kysyy sen, jos sitä ei ole olemassa.

Scripti pystyy havaitsemaan sisäisesti millä käyttöjärjestelmällä sitä ajetaan ja se on kirjoitettu toimivaksi suoraan Ubuntu, Fedora ja Centos käyttöjärjestelmissä, virtuaalisissa, sekä raudan päällä.

Skriptissä oli maininta myös EPEL repositoriosta, joka selvitettiin olevan "Extra Packages for enterprise Linux". Tämä paketti lisää vakaampia ja luotettavempia paketteja linuxiin, joita OpenStack sittemmin käyttää hyväkseen.

Stack.sh skripti on kirjoitettu siten, että se antaa aina error viestin, jos jokin epäonnistuu, eikä sen pitäisi jäädä roikkumaan. Se myös importtaa TLS ja Apache funktiot tietokantojen sekä frontendin suojaukseen.

Stack.sh scripti lukee myös local.conf tiedostosta tietokannan tyyppin, esimerkissä oli vaihtoehtoina postgresql ja mysql. Näiden jälkeen skripti asentaa myös jonkinlaisen RPC (remote procedure call) palvelun.

## 5 Työn kulku

Käytimme Openstack Docs -ohjetta DevStacking asennukseen (OpenStack Docs: DevStack). Ohjeen mukaan suositelluin alusta oli Ubuntu 18.04, joten asensimme DevStackin siihen. DevStack vaati minimissään 8GB muistia, 2 prosessoriydintä ja 60GB levytilaa (OpenStack Docs: Setup DevStack.) Meidän virtuaalikoneessamme, oli 8GB muistia, 4 ydintä ja 50GB levytilaa.

### 5.1 Käyttäjän luominen

DevStackia suositellaan ajettavaksi erillisen käyttäjän alaisuudessa, jolla on sudo-oikeudet. Siispä loimme ohjeiden mukaan käyttäjän nimeltä "stack" (Kuvio 1), ja lisäsimme sille sudoaja tiedoston, jossa määrittelimme että käyttäjä ei tarvitse sudoamiseen salasanaa.



```

useradd -s /bin/bash -d /opt/stack -m stack
root@jamk:~# echo "stack ALL=(ALL) NOPASSWD: ALL" | tee /etc/sudoers.d/stack
stack ALL=(ALL) NOPASSWD: ALL
root@jamk:~# cat /etc/sudoers.d/stack
stack ALL=(ALL) NOPASSWD: ALL
root@jamk:~#

```

Kuvio 1 Käyttäjän luominen.

## 5.2 DevStacking asennus

Ensimmäiseksi siirryimme luomallemme "stack" -käyttäjälle. (Kuvio 2)

```

poke@jamk:~$ sudo su - stack
[sudo] password for poke:
stack@jamk:~$ git

```

Kuvio 2 Käyttäjän vaihtaminen.

Sitten kloonasimme DevStack-repon. (Kuvio 3)

```

stack@jamk:~$ git clone https://opendev.org/openstack/devstack
Cloning into 'devstack'...
remote: Enumerating objects: 44376, done.
remote: Counting objects: 100% (44376/44376), done.
remote: Compressing objects: 100% (13357/13357), done.
remote: Total 44376 (delta 31683), reused 42575 (delta 30334)
Receiving objects: 100% (44376/44376), 8.41 MiB | 7.46 MiB/s, done.
Resolving deltas: 100% (31683/31683), done.
stack@jamk:~$ cd devstack
stack@jamk:~/devstack$ ls
clean.sh  extras.d  functions-common  HACKING.rst  LICENSE  openrc  roles  setup.cfg  stack.sh  tox.ini
data      files     FUTURE.rst      inc          MAINTAINERS.rst  playbooks  run_tests.sh  setup.py  tests  unstack.sh
doc       functions  gate             lib          Makefile  README.rst  samples      stackrc   tools
stack@jamk:~/devstack$

```

Kuvio 3 DevStack-repon kloonaus.

Ennen asennuksen aloittamista täytyy luoda "local.conf" -asetustiedosto. Sen voi joko tehdä itse, tai kopioida kloonatun repon kansiota "samples". Ohje suositteli tekemään vähintään Kuvion 4 mukaisen configin, mutta emme saaneet DevStackia asennettua vain sillä.

```

stack@jamk:~/devstack$ nano stack.sh
stack@jamk:~/devstack$ nano local.conf
stack@jamk:~/devstack$ cat local.conf
[[local|localrc]]
ADMIN_PASSWORD=salasana
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
stack@jamk:~/devstack$

```

Kuvio 4 local.conf.

Kopioimme siis sen sijaan asetustiedoston "samples" -kansioista. Se eroaa Kuvion 4 asetustiedostosta hieman. Ensinnäkin salasanat ovat hieman eri, mutta sillä ei ole niinkään väliä. Myös asetetaan shell.sh kirjaaman lokit tiedostoon. (Kuvio 5)

```
# Logging
# -----

# By default ``stack.sh`` output only goes to the terminal where it runs. It can
# be configured to additionally log to a file by setting ``LOGFILE`` to the full
# path of the destination log file. A timestamp will be appended to the given name.
LOGFILE=$DEST/logs/stack.sh.log

# Old log files are automatically removed after 7 days to keep things neat. Change
# the number of days by setting ``LOGDAYS``.
LOGDAYS=2
```

Kuvio 5 local.conf loggaus asetus.

Asetustiedosto myös asettaa joitain Swiftin muuttujia. (Kuvio 6) Ensimmäinen, "SWIFT\_HASH" on pakollinen, mutta stack.sh skriptin pitäisi kysyä sitä jos se ei ole asetettu "local.conf" -tiedostossa. "SWIFT\_REPLICAS" -määrittää että ei tarvita kuin 1 kopio tietokannasta resurssien säästämiseksi. "SWIFT\_DATA\_DIR" -määrittää kansion mihin Swiftin data tallennetaan.

```
# Swift
# -----

# Swift is now used as the back-end for the S3-Like object store. Setting the
# hash value is required and you will be prompted for it if Swift is enabled
# so just set it to something already:
SWIFT_HASH=66a3d6b56c1f479c8b4e70ab5c2000f5

# For development purposes the default of 3 replicas is usually not required.
# Set this to 1 to save some resources:
SWIFT_REPLICAS=1

# The data for Swift is stored by default in (``$DEST/data/swift``),
# or (``$DATA_DIR/swift``) if ``DATA_DIR`` has been set, and can be
# moved by setting ``SWIFT_DATA_DIR``. The directory will be created
# if it does not exist.
SWIFT_DATA_DIR=$DEST/data
```

Kuvio 6 local.conf Swift muuttujat.

## 5.3 DevStack'in ajaminen ja hallinnointi.

DevStack on asennettu, ja toiminnassa. Kuvassa 7 näkyy terminaali tuloste DevStackin onnistuneen asennuksen jälkeen.

```

pip_install      370
apt-get          222
-----
Unaccounted time  503
=====
Total runtime    1686

This is your host IP address: 192.168.48.101
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.48.101/dashboard
Keystone is serving at http://192.168.48.101/identity/
The default users are: admin and demo
The password: nomoresecret

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

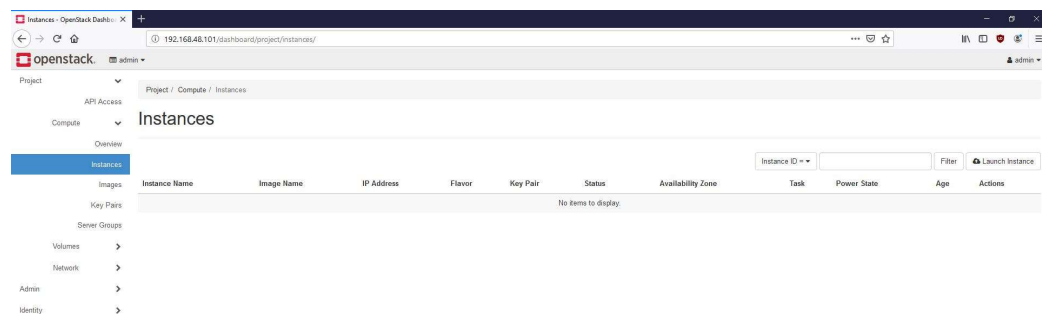
DevStack Version: ussuri
Change: 63ab664baf99716e9a5d29eb82f974f8c3456816 Merge "Revert "install LIBS_FROM_GIT using python 2
and 3 where appropriate"" 2019-11-04 20:18:13 +0000
OS Version: Ubuntu 18.04 bionic

stack@jamk:~/devstack$

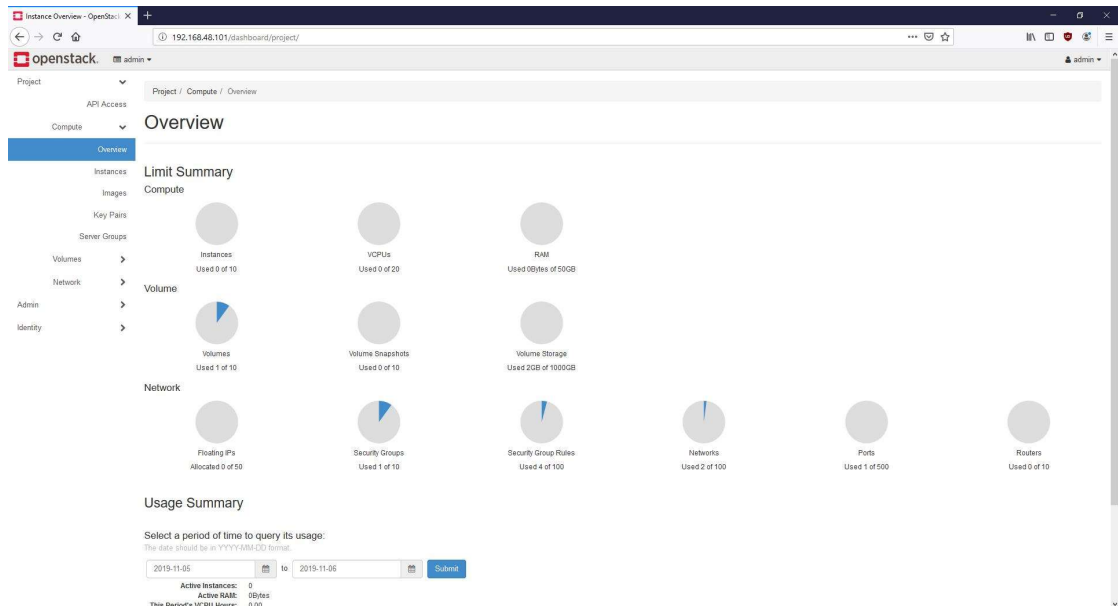
```

Kuvio 7 DevStack asennettu.

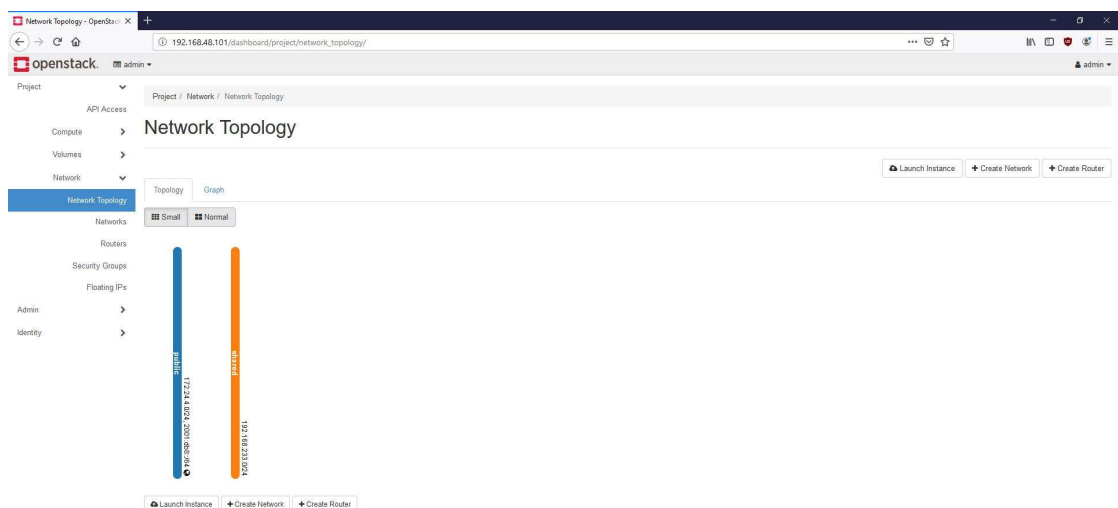
Voimme nyt mennä internet selaimella virtuaalikoneen IP -osoitteeseen, tässä tapauksessa 192.168.48.101. Varmuuden vuoksi voi vielä kirjoittaa perään `"/dashboard"` jotta varmasti pääsee OpenStackin Horizon-web-käyttöliittymään. (Kuviot 8, 9 & 10) Siellä voi hallita virtuaalikoneita sekä valvoa resurssien käyttöä yms.



Kuvio 8 OpenStackin web-käyttöliittymä.



Kuvio 9 OpenStackin web-käyttöliittymä.



Kuvio 10 OpenStackin web-käyttöliittymä.

## 5.4 Virtuaalikoneen lisääminen OpenStackissa

### 5.4.1 Levykuvan lisäys

Asennus sujui Openstack ohjeilla (OpenStack Docs: Launch and manage instances). Ensimmäiseksi avataan OpenStackin Images-välilehti, ja lisätään uusi levykuva. Images välilehdeltä valitaan Create Image ja valitaan sopivat asetukset (Kuvio 11). Sen jälkeen voimme luoda uuden virtuaalikoneen.

OpenStack:in sivuilla on tietoa levykuvista, suositellen erityisesti pilvipalvelu käyttöön suunniteltuja jakeluita. (OpenStack Docs: Get Images.) Levykuva voi olla pelkkä .iso tiedosto, josta käyttöjärjestelmä asennetaan, mutta myös virtuaalinen kovalevy, kuten .vdi tai .ova-virtualisointi template. OpenStackissa on myös enemmän vaihtoehtoja kuin tässä mainitaan.

DevStackin mukana tulee cirros testi levykuva, jolla voi testata toimivuutta. (Huom! Kuvissa Alpine Linux, mutta loppujen lopuksi käytimme Cirros testi levykuvaa. Ohjeet soveltuvat yleisesti kaikkiin.)

Kuvio 11 Create Image-ikkuna.

## 5.4.2 Instanssin luominen

Avaamme Instances sivun OpenStack web-käyttöliittymästä ja klikkaamme Launch Instance. Avautuu velho, jossa syötämme vaaditut tiedot.

Ensimmäisellä sivulla (Kuvio 12) Laitetaan nimi ja kuvaus, sekä tärkeämmin Availibity zone, sekä kuinka monta konetta luodaan. Availibity Zone on palvelun ylläpitäjän

luoma alue esimerkiksi, jos instanssin haluaa pyöriävän Euroopassa Availability Zone voisi olla "eu-west". Tässä testitapauksessa se on "nova". Luomme vain yhden koneen.

**Launch Instance**

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

**Instance Name \***  
TestAlpine

**Description**  
Alpine test VM

**Availability Zone**  
nova

**Count \***  
1

Total Instances (10 Max)  
10%

0 Current Usage  
1 Added  
9 Remaining

**Details**  
Source \*  
Flavor \*  
Networks \*  
Network Ports  
Security Groups  
Key Pair  
Configuration  
Server Groups  
Scheduler Hints  
Metadata

**Cancel** **< Back** **Next >** **Launch Instance**

Kuvio 12 Launch Instance-velhon ensimmäinen sivu.

Seuraavaksi valitsemme aikaisemmin luomamme levykuvan (Kuvio 13) ja valitsemme samalla kuinka ison tallennustilan instanssi saa.

**Launch Instance**

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

**Select Boot Source**  
Image

**Create New Volume**  
Yes No

**Volume Size (GB) \***  
1

**Delete Volume on Instance Delete**  
Yes No

**Allocated**

Name	Updated	Size	Type	Visibility
Alpine-Virtual	11/8/19 12:26 PM	35.00 MB	iso	Public

**Available 1**

Select one

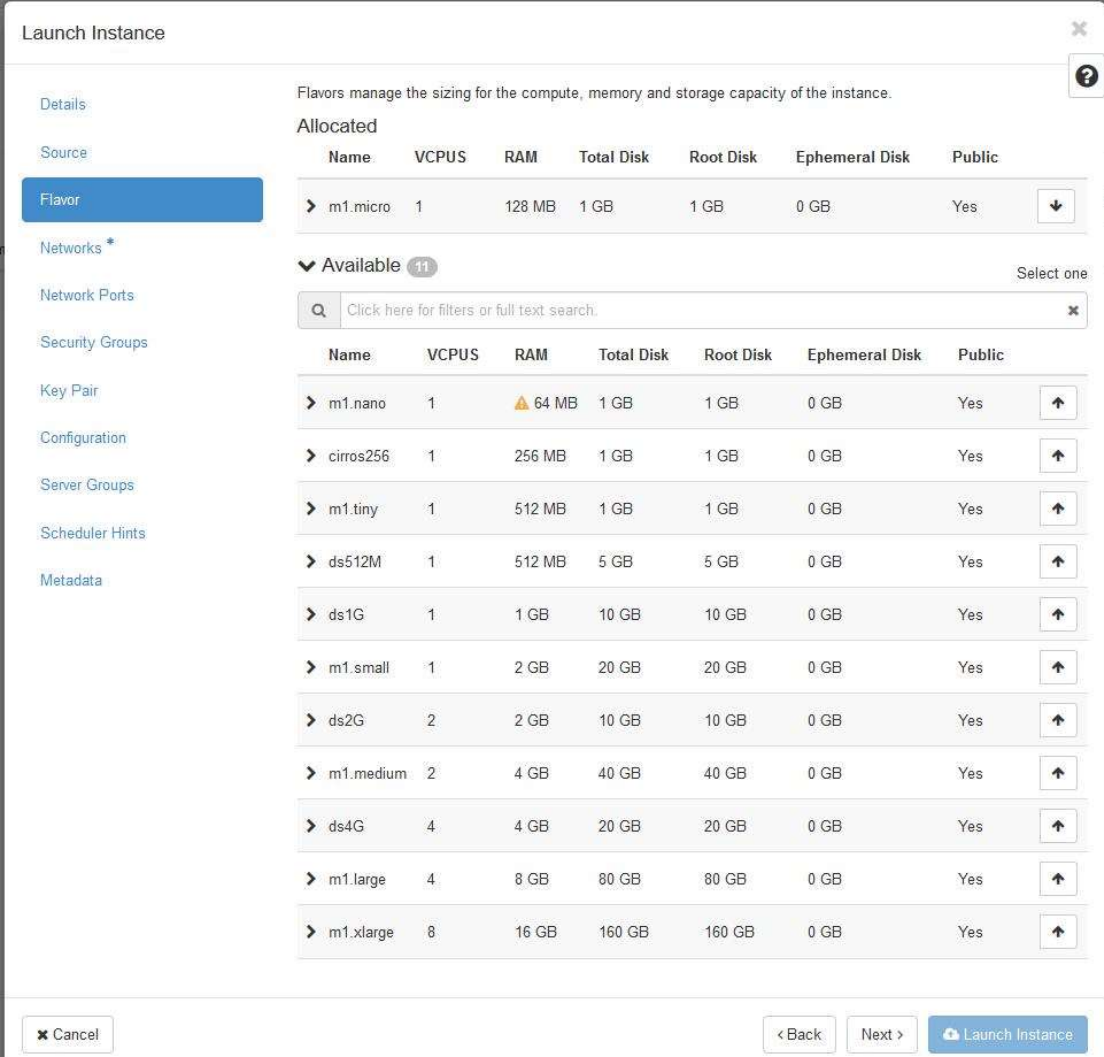
Q Click here for filters or full text search.

Name	Updated	Size	Type	Visibility
cirros-0.4.0-x86_64-disk	11/8/19 12:03 PM	12.13 MB	qcow2	Public

**Cancel** **< Back** **Next >** **Launch Instance**

Kuvio 13 Launch Instance-velhon levykuvan valinta.

Flavor nimisessä vaiheessa valitsemme esiasetetuista vaihtoehtoista kuinka paljon muistia ja prosessoritehoa instanssi käyttää (Kuvio 14).



Launch Instance

Details

Source

Flavor

Networks \*

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.micro	1	128 MB	1 GB	1 GB	0 GB	Yes

Available 11

Select one

Click here for filters or full text search.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.nano	1	64 MB	1 GB	1 GB	0 GB	Yes
cirros256	1	256 MB	1 GB	1 GB	0 GB	Yes
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes
ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes
m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes

Cancel

< Back

Next >

Launch Instance

Kuvio 14 Launch Instance-velhon koneen resurssien valinta.

Sen jälkeen valitsemme instanssin verkkoyhteydet (Kuvio 15).



**Launch Instance**

Details

Source

Flavor

**Networks**

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Networks provide the communication channels for instances in the cloud.

**Allocated** 1

Select networks from those listed below.

Network	Subnets Associated	Shared	Admin State	Status
1 shared	shared-subnet	Yes	Up	Active

**Available** 1

Select at least one network

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status
private	ipv6-private-subnet private-subnet	No	Up	Active

**Cancel** **< Back** **Next >** **Launch Instance**

Kuvio 15 Launch Instance-velhon verkon valinta.

Jotta instanssiin saa SSH -yhteyden, täytyy luoda avainpari tunnistautumista varten (Kuviot 16 ja 17).

**Launch Instance**

Details

Source

Flavor

Networks

Network Ports

Security Groups

**Key Pair**

Configuration

Server Groups

Scheduler Hints

Metadata

A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

**+ Create Key Pair** **Import Key Pair**

**Allocated**

Displaying 1 item

Name	Type	Fingerprint
alpinetest	ssh	50:27:82:9b:4c:48:fc:1d:6a:88:16:c8:91:0f:af:0e

Displaying 1 item

**Available** 0

Select one

Click here for filters or full text search.

Displaying 0 items

Name	Type	Fingerprint
No items to display.		

Displaying 0 items

**Cancel** **< Back** **Next >** **Launch Instance**

Kuvio 16 Launch Instance-velhon avainpari SSH:ta varten.



Create Key Pair

Key Pairs are how you login to your instance after it is launched. Choose a key pair name you will recognize. Names may only include alphanumeric characters, spaces, or dashes.

**Key Pair Name \***

**Key Type \***

SSH Key

**Private Key**

```
cuasw+ECgYEA5zI4+cPJaYiEkr9pw+FeOPRrAJFaXCNN9DMRfq16meixA1KpgDA7
9UYFQ3hzhH5B4/sMZ2NvTyT9f8XkvqaaZuiqK3UxqTfwm4GLJVUTEOldrKEgU40m
CTqSAkpwE9edoJ8XEnD6d4d8ep3Ij6w29eCcRW/2C6nAWG1KbkHCovUCgYEAwKb9
BNMSiUmUSgyhLPto66fNTgwFvtCkWPowl+pQE8buXhtxounGyUQkpTL2wnZp8Ef
9Gqxmcbr0Sxpl55tnnSAC6hvaVVnv3BzCPtnXNRPDpTXEdlphgINQFGYWIVkfaS
H7m6YbxmHb257qTWioHMqdMBEMqS4qRLihKj3RUCgYEAplPxxqoTizkjW6xAplQp
rRJTgxZyZvh+CN28ixUBWcEIFT9mBCbQ1nCjWHXlizquvY24M3K0DHPAsQxglbjR
14Otc8cztMVrTdQ0N0Gh3YMFONVqw+2lcDMvL9A52/WxbV0vN+IQdLmL4V63d7h9
Wh8oO8hIMd7+WDJaMwY9em0CgYBgZ0IUNFEbRitLbr9LoGbgDOOnBb477SktcHxUz
5xonCGWRMewfmqv1mxGR503tmyyR/R12sGrg8KUdKaTXblR9jOOPeCvMTcudyv8z
o0UwWvuDQ040PQGnBmOYj6gzy4xtCtH4AgF9yuFhIK7pn/dCE71hAAEFrfs5kzJ
dEJ95QKBgA+ZsJ5+XJgpumlORE0/d4fQw99usT5ZxHe+Rk9N/8xQb3bfjD9tsnwy
Piaxgu2fmr0ZE6llLckUvpciGcWPBYopujphEEIXAh2ziKitYf9CIs26i4FBZ2
J65biWlrs0CecKdxnjp6kxLpWfptZSTraqg8tMvmsADS1bmFGTQJ
-----END RSA PRIVATE KEY-----
```

Create Keypair

Copy Private Key to Clipboard

Done

Kuvio 17 Launch Instance-velhon Create Key Pair-ikkuna.

Lopputuloksena instanssi on luotu (Kuvio 18) ja instanssin pitäisi käynnistyä itsestään, jollei, sen voi käynnistää "Actions"-painikkeesta.

openstack

demo

admin

Project

API Access

Compute

Instances

Images

Key Pairs

Server Groups

Volumes

Network

Admin

Identity

Project / Compute / Instances

Instances

Instance ID

Filter

Launch Instance

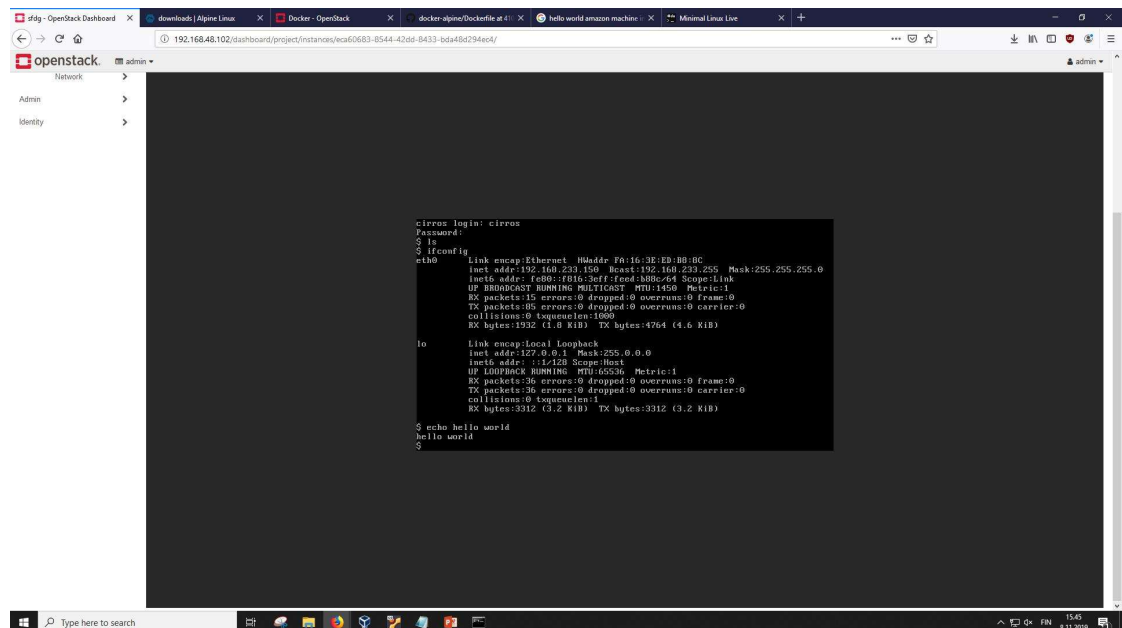
Delete Instance

More Actions

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
alpine	-		m1.nano	alpinetest	Build	nova	Scheduling	No State	0 minutes	Associate Floating IP

Kuvio 18 Instanssi on luotu.

Kun instanssi on käynnissä, mennään instanssin web-konsoliin klikkaamalla instanssin nimestä ja vaihtamalla sitten ”Console”-välilehteen (Kuvio 19), josta nähdään että instanssi on käynnissä.



Kuvio 19 Instanssin web-konsoli.

## 6 Pohdinta

Työn aloitus ja sen tekeminen kokonaisuudessaan oli hyvin sulava prosessi. Alun perin kuitenkin havaitsimme odottamattoman ongelman, kun ensimmäinen devstackin asennusyritys ei toiminut oikein (Horizon ei suostunut käynnistymään asennuksen jälkeen, todennäköisesti sen takia, että virtuaalikoneelle oli annettu vähemmän kuin 4GB muistia, ja OpenStackin minimivaatimus on 4GB) ja muutama seuraava yritys oli jäämässä jumiin asennuksen aikana jostain syystä, vaikkakin virtuaalikoneella on ollut käytössä 8GB muistia ja 2 säiettä.

## Lähteet

OpenStack: Core Components. DZone.com artikkeli OpenStackin komponenteista. Viitattu 6.11.2019. <https://dzone.com/articles/openstack-core-components>.

OpenStack Docs: DevStack. DevStacking asennusohje. Viitattu 8.11.2019. <https://docs.openstack.org/devstack/latest/>.

OpenStack Docs: Get images. OpenStackin artikkeli sopivista levykuvista. Viitattu 8.11.2019. <https://docs.openstack.org/image-guide/obtain-images.html>.

OpenStack Docs: Launch and manage instances. OpenStackin ohje instanssin luomiseksi. Viitattu 8.11.2019. <https://docs.openstack.org/horizon/latest/user/launch-instances.html>.

OpenStack Docs: Setup DevStack. DevStackin asennusohje. Viitattu 8.11.20.19. <https://docs.openstack.org/sahara/latest/contributor/devstack.html>.

OpenStack Services. OpenStack.org sivuston komponenttien esittely. Viitattu 6.11.2019. <https://www.openstack.org/software/project-navigator/openstack-components#openstack-services>.