FORGE Service Lab
OpenStack basics

Pasi Kivikangas

This material is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

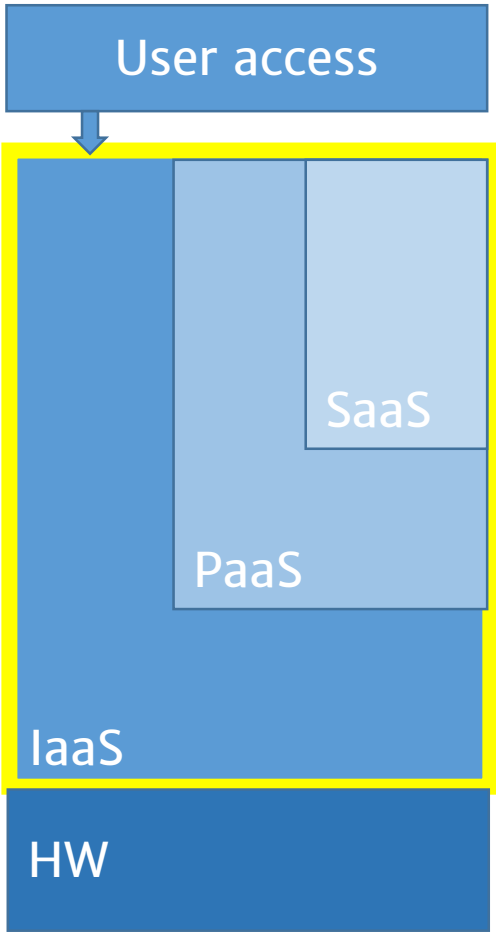
http://creativecommons.org/licenses/by-sa/3.0/

Copyright © DIGILE Ltd

http://www.digile.fi

Cloud computing service models

- SaaS (Software as a Service)
  - Provides you with access to application software. You don't have to worry about the installation, setup and running of the application. Service provider will do that for you. You just have to pay and use it through some client.
  - Examples: Google Apps, Microsoft Office 365.
- PaaS (Platform as a Service)
  - Computing platforms which typically includes operating system, programming language execution environment, database, web server etc.
  - Examples: AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos
- IaaS (Infrastructure as a Service)
  - Computing infrastructure, physical or (quite often) virtual machines and other resources like virtual-machine disk image library, block and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks etc.
  - Examples: Amazon EC2, Windows Azure, Rackspace, Google Compute Engine



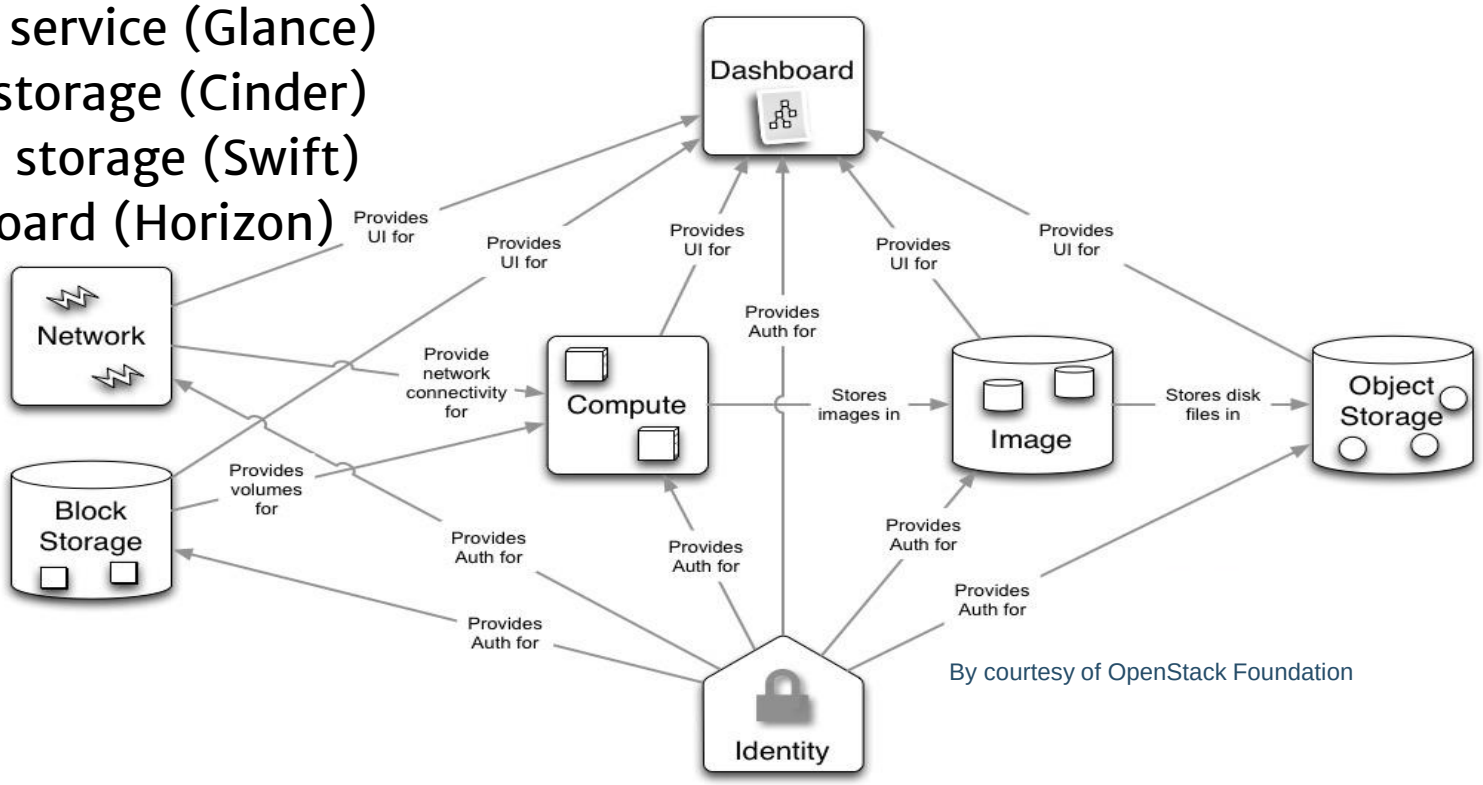
FORGE OpenStack IaaS

- FORGE uses OpenStack to provide IaaS
- OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources
  - Processing
  - Storage
  - Networking resources
  - KVM hypervisors
- Users manage resources using
  - A web-based dashboard
  - Command-line tools
  - RESTful API
- OpenStack is a free and open-source software
  - OpenStack.org release it under the terms of the Apache License 2.0

https://cloud.forgeservicelab.fi

OpenStack

- OpenStack consists of several integrated projects
  - Compute (Nova)
  - Identity (Keystone)
  - Networking (Neutron)
  - Image service (Glance)
  - Block storage (Cinder)
  - Object storage (Swift)
  - Dashboard (Horizon)



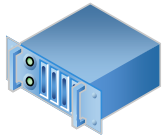
By courtesy of OpenStack Foundation

IaaS usage illustrated

User

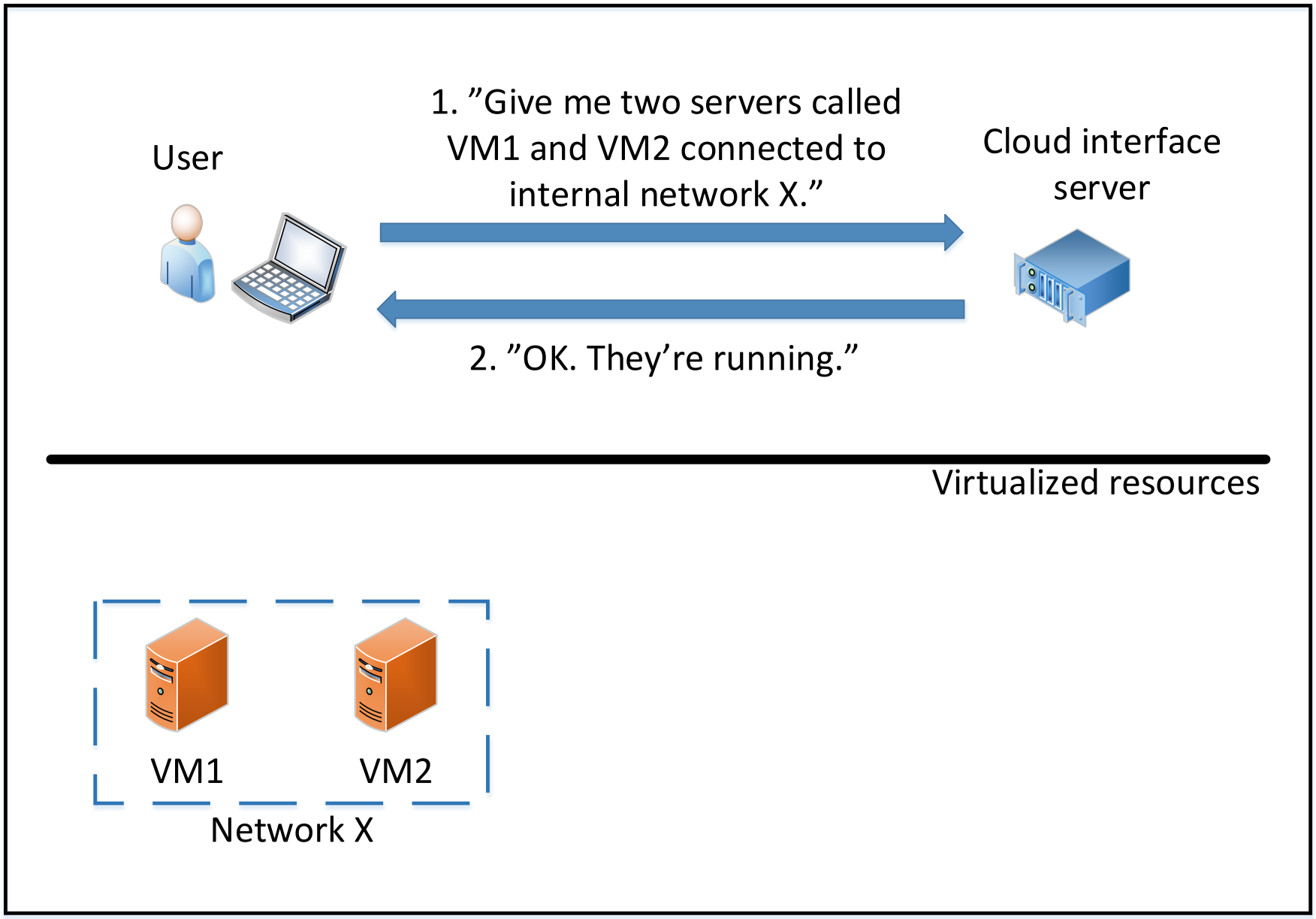
1. ”Give me two servers called VM1 and VM2 connected to internal network X.”

Cloud interface server

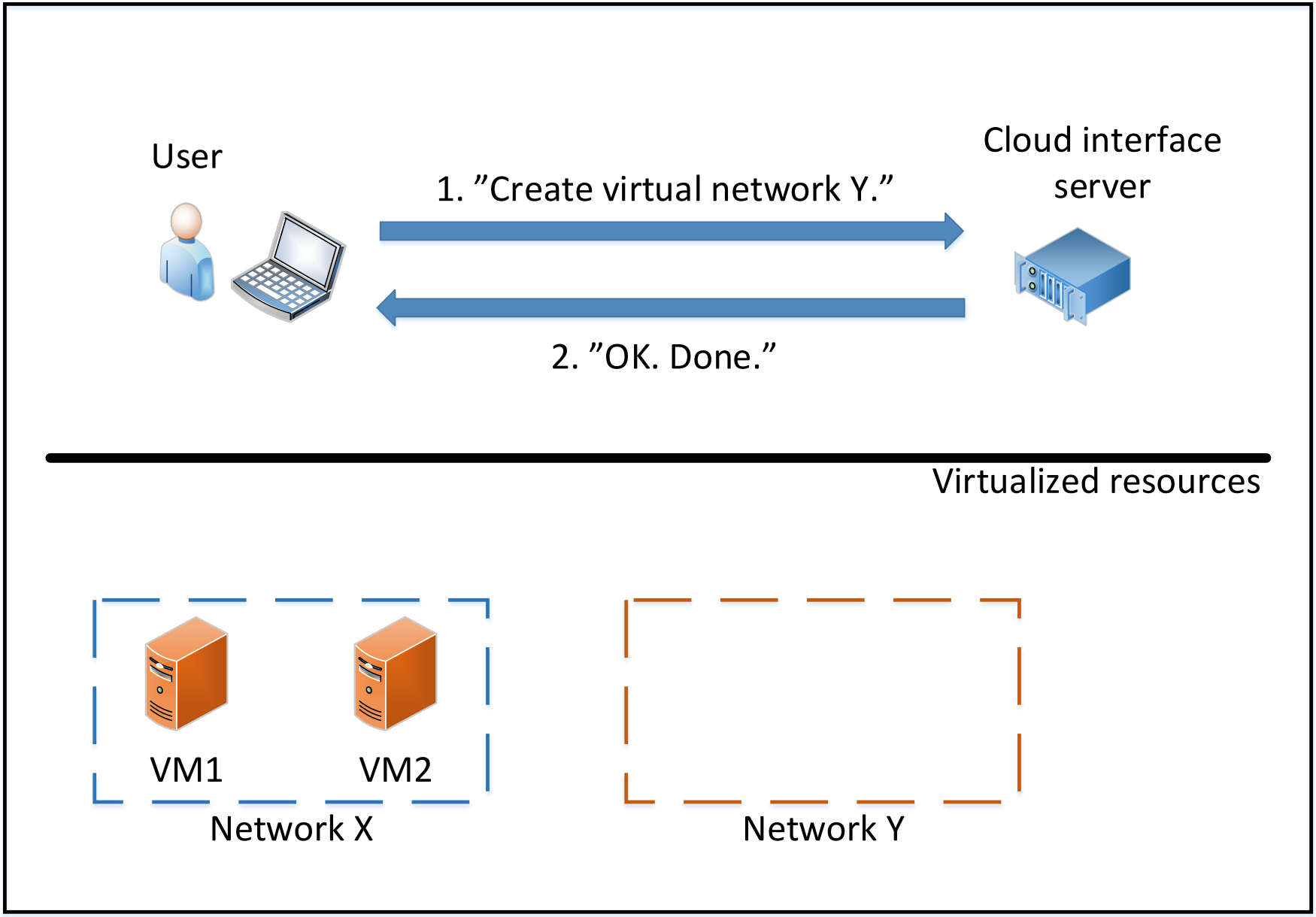


Virtualized resources

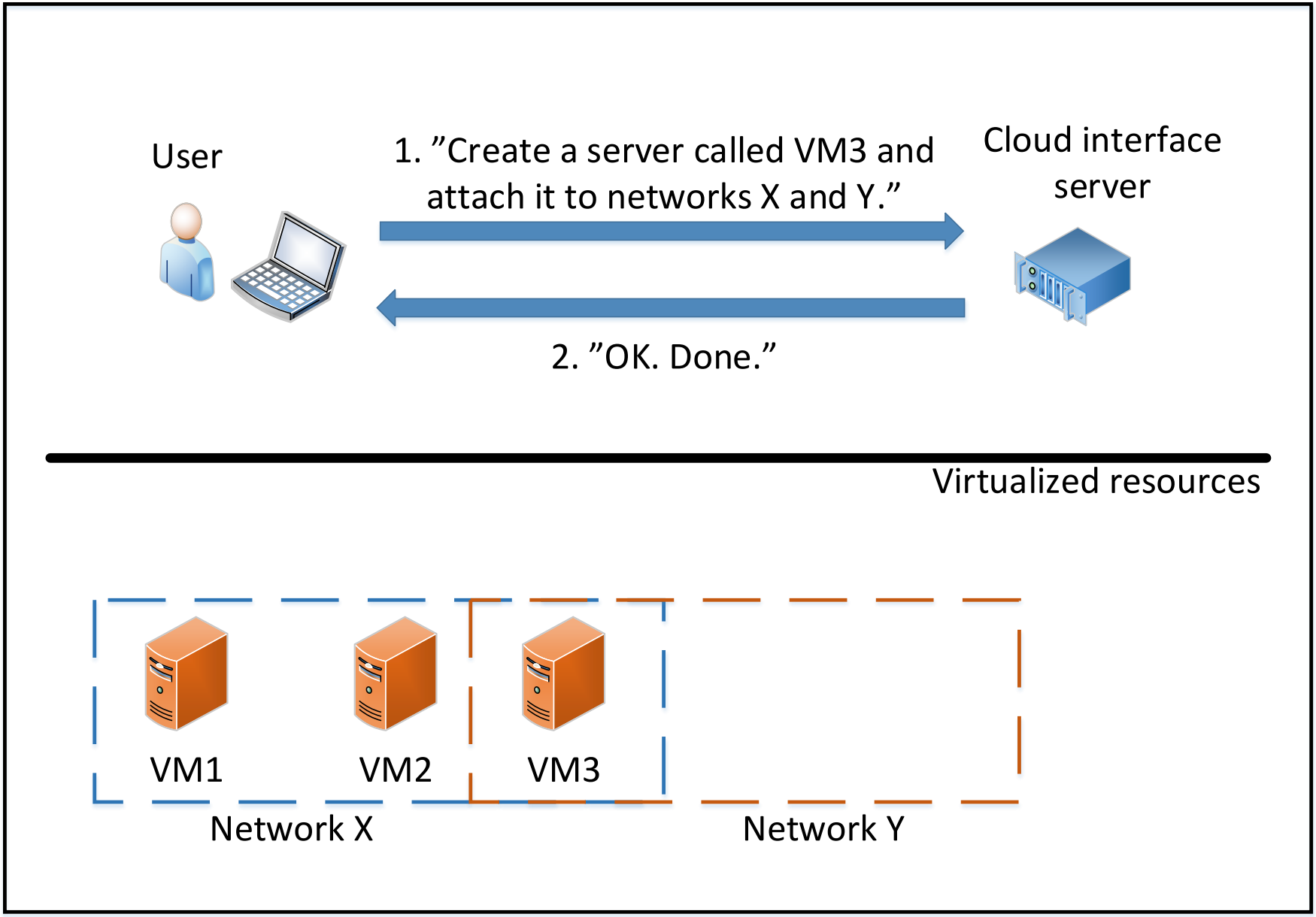
By courtesy of Risto Laurikainen, CSC



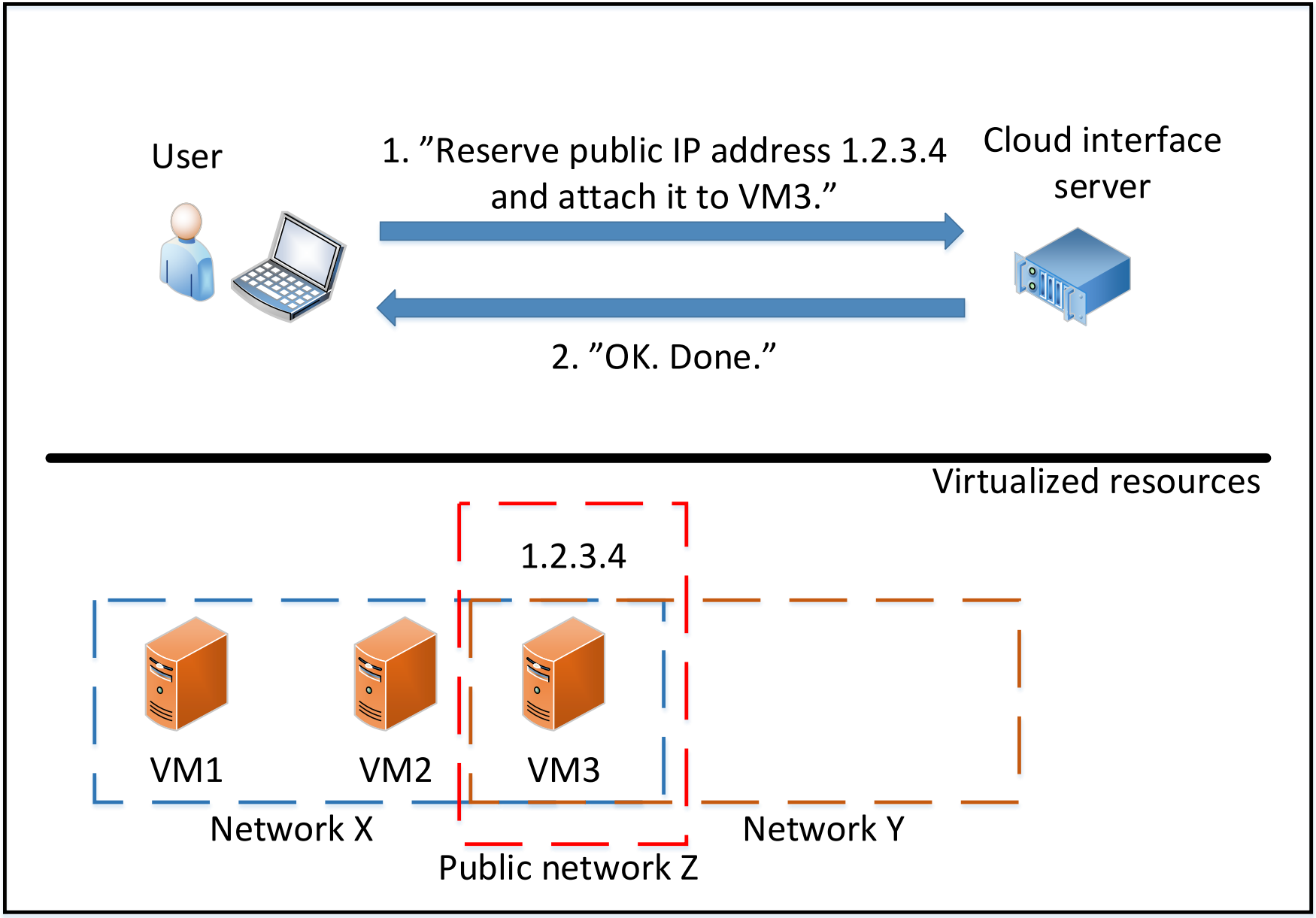
By courtesy of Risto Laurikainen, CSC



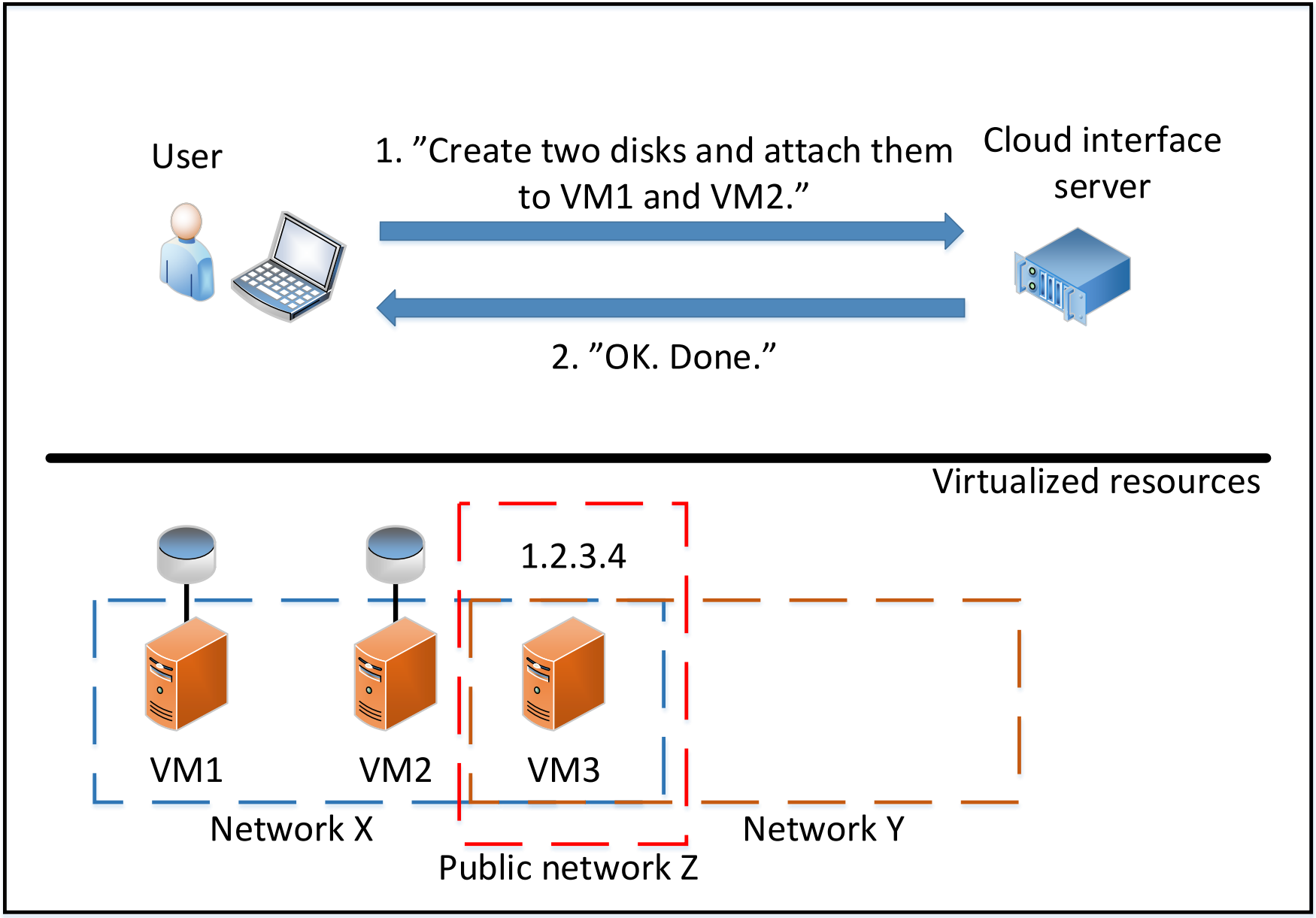
By courtesy of Risto Laurikainen, CSC



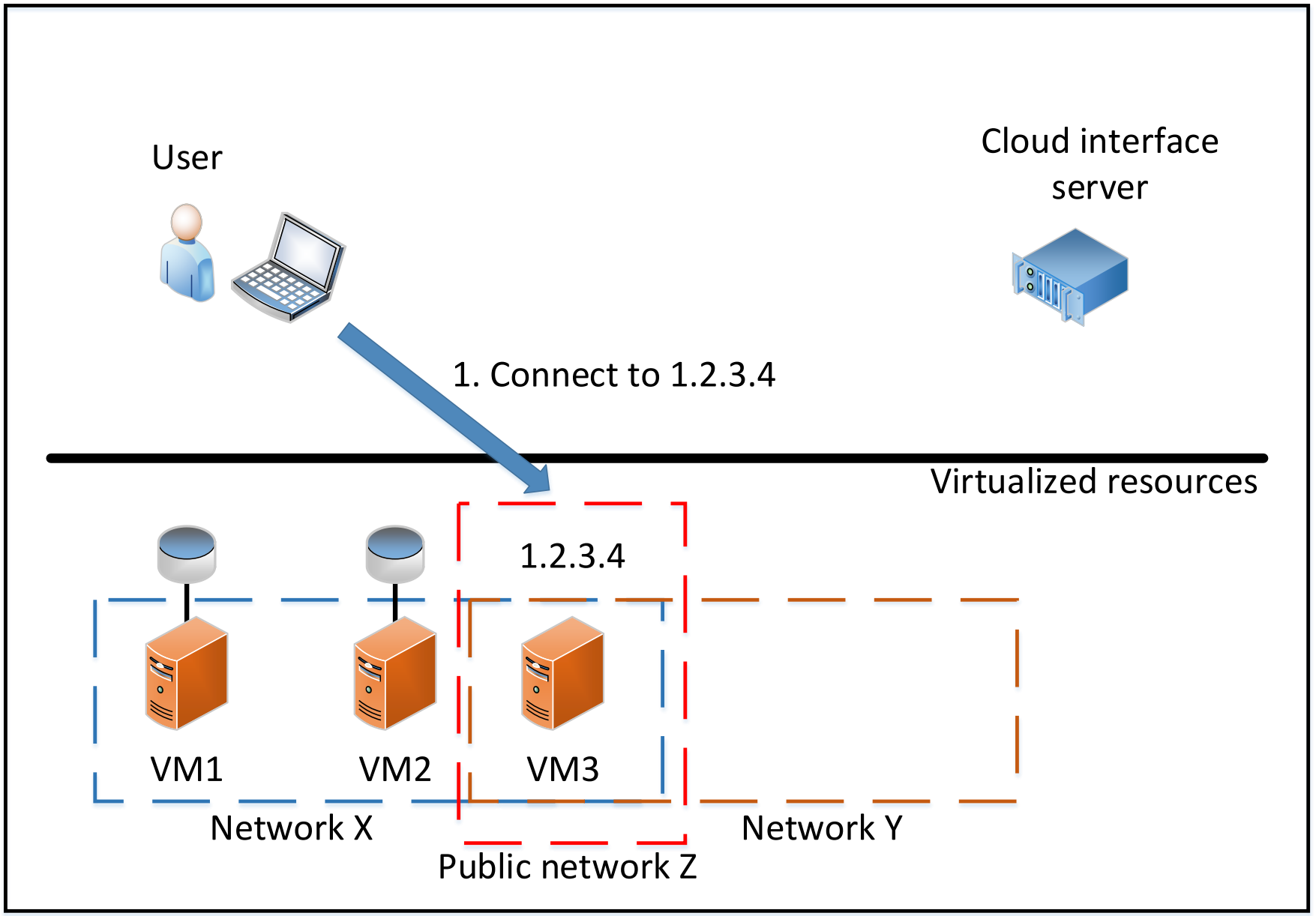
By courtesy of Risto Laurikainen, CSC



By courtesy of Risto Laurikainen, CSC



By courtesy of Risto Laurikainen, CSC

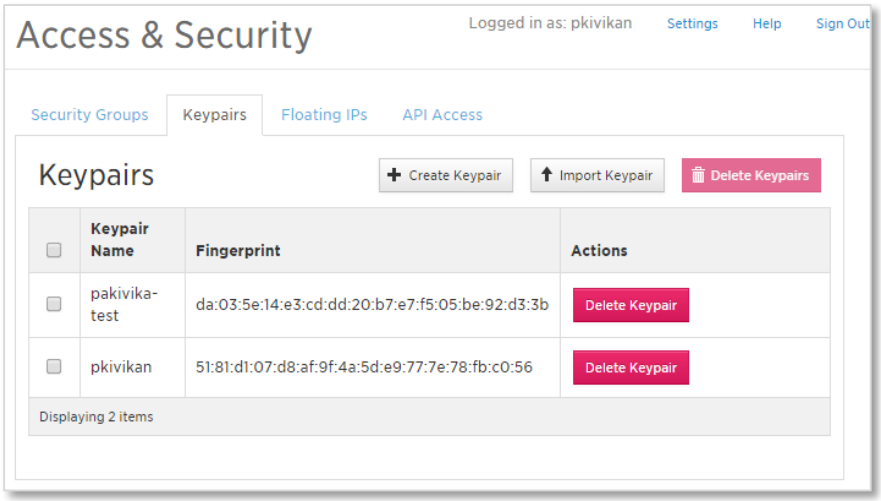


By courtesy of Risto Laurikainen, CSC

IaaS usage

Preparations – SSH keys

- Using SSH keys is the default way to access new virtual machines
- You only need to set up your SSH keys once
- You can either upload your existing SSH key or create a new key using Dashboard (Horizon)



Upload existing key

- In the web interface go to **Access & Security** and from **Keypairs** select **Import Keypair**
- Name your key, and paste your public key (starts with something like "ssh-rsa AAFAA...." or "ssh-dss AFAFA...") into the other box.

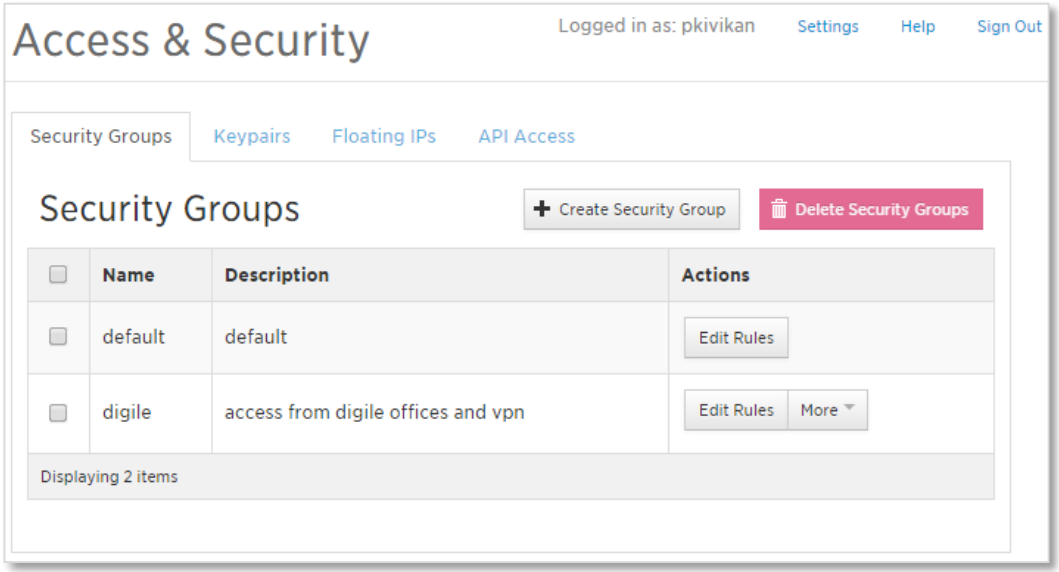
Create a new key

- If you have not used SSH keypairs before, you need to create one.
- The dashboard can take care of this for you.
- Go to **Access & Security** and from **Keypairs** select **Create Keypair**. Give your key a name and click **create**. Now you will get a "*keyname.pem*" to save. Save it in your home directory and set permissions to 400

Preparations – Security groups

- Security groups are sets of firewall rules which limit access to your machines
- Security groups of OpenStack can be configured using Dashboard (Horizon)
- You may have additional firewall rules within your virtual machine
- Security groups are easy to manage by creating a dedicated group for a particular purpose and avoiding using “Default” security group

- All external incoming traffic is denied by default and you can allow connections from a set of IP addresses and ports
- Security groups can be created and edited in Horizon **Access & Security** page



Exercise 1 – Create a SSH key

EXERCISE

- Search instructions from wiki about how to create a SSH key for you
  - Login to https://support.forgeservicelab.fi
  - Tip: Search for ”Getting started with OpenStack”
- Create a SSH key for yourself using Horizon, give your key a name, download pem file to your home directory and set the right permissions for the file

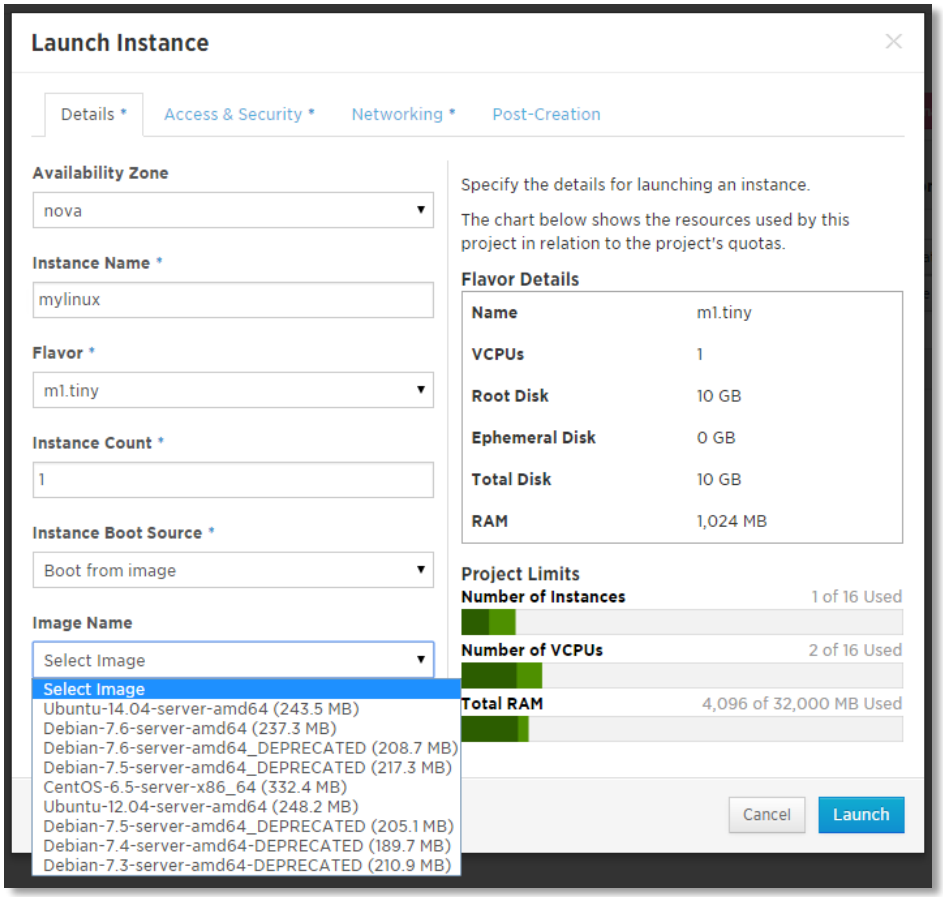
Exercise 2 – Create a security rule

EXERCISE

- Search instructions from wiki about how to create a security group
  - Login to https://support.forgeservicelab.fi
  - Tip: Search for ”Getting started with OpenStack”
- Create a new security group using Horizon, give it a name
- Then edit the security group and add a rule to allow incoming IPv4 traffic from your current IP address to SSH port

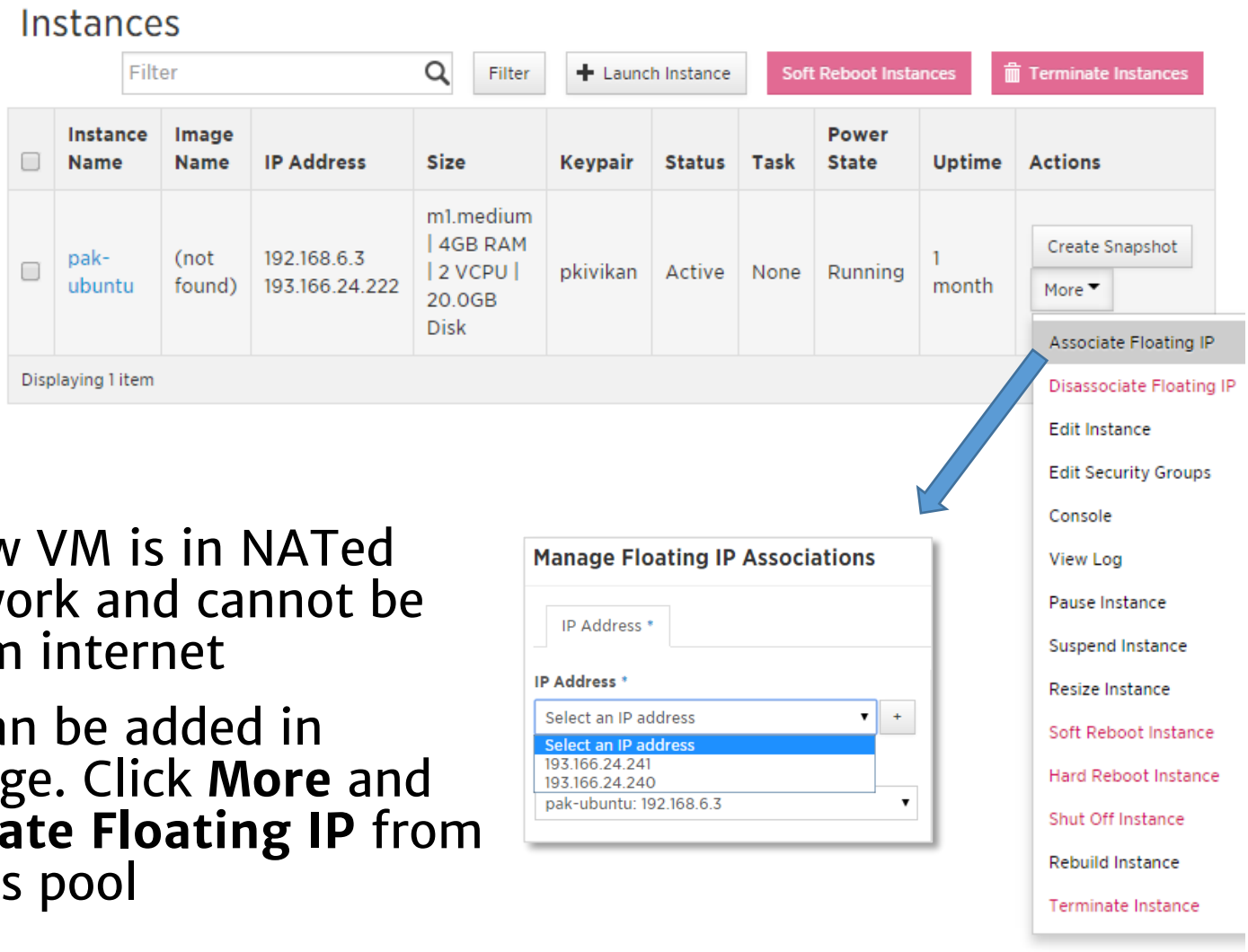
Instances – Launching virtual machines

- Virtual machines are launched from Horizon **Instances** page
- VM properties can be defined after selecting **Launch**
- SSH key and security group for the VM is selected in **Access & Security** tab
- Flavors determine the sizing for the virtual machines



ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public
1	m1.tiny	1024	10	0		1	1.0	True
2	m1.small	2048	10	0		1	1.0	True
3	m1.medium	4096	20	0		2	1.0	True
4	m1.large	8192	40	0		4	1.0	True
5	m1.x-large	16384	80	0		8	1.0	True

Instances – Associating public IP



- Initially a new VM is in NATed internal network and cannot be accessed from internet
- A public IP can be added in **Instances** page. Click **More** and select **Associate Floating IP** from the IP address pool

Exercise 3 – Launch Linux image

EXERCISE

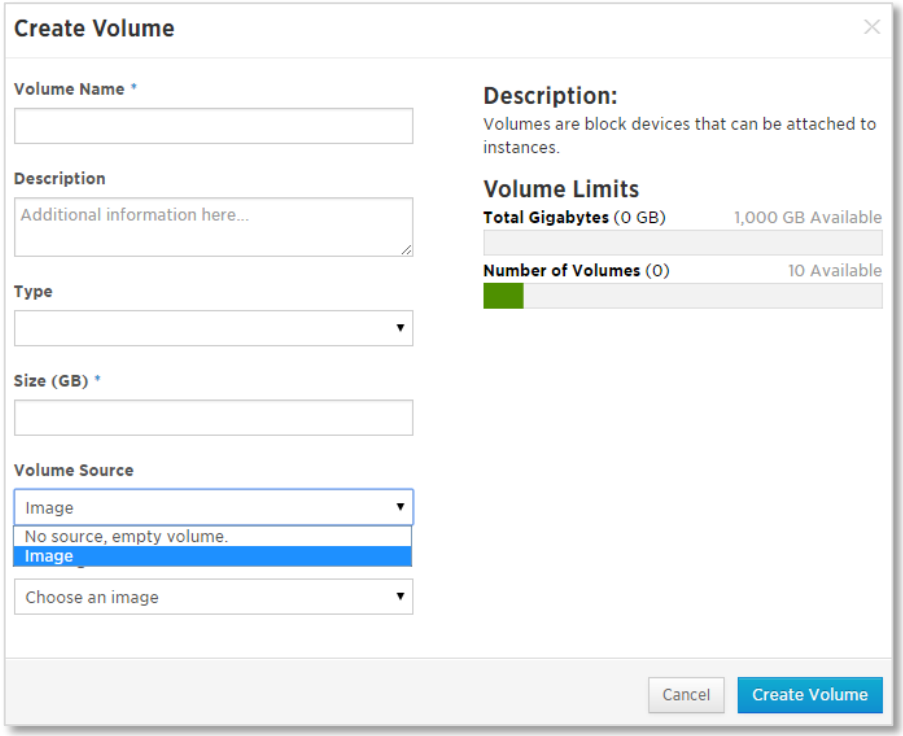
- Search instructions from wiki about how to launch a Linux image using Horizon GUI
  - Login to https://support.forgeservicelab.fi
  - Tip: Search for ”Launch Linux image”
- Launch Ubuntu 14.04
  - Login to https://cloud.forgeservicelab.fi
  - Launch Ubuntu 14.04 server image from the list of available images
  - Use the SSH key and security group you have created earlier
  - Associate a public IP address for the VM
  - Make a SSH connection by issuing a command
    - `ssh ubuntu@`*`public-ip`* `-i` *`key.pem`*

Persistent storage

- Virtual machines have storage space (instance storage) in the root filesystem as defined by the flavor that was used when the instance was created
- Terminating the instance also terminates the instance storage
- Persistent volumes can be used to retain data even when instances are removed
- Persistent volumes can be attached or detached from instances while they are running
- The behavior is quite similar to mounting e.g. an USB stick to Linux on the fly

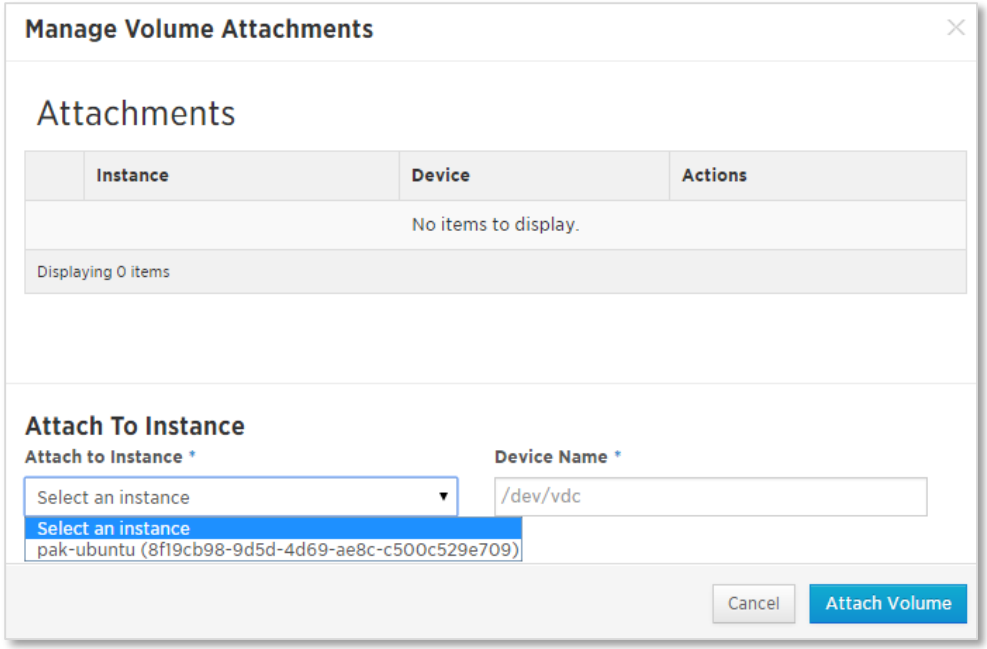
Persistent storage –Volume creation

- Persistent volumes are created in the Horizon **Volumes** page
- **Create volume** button creates a new volume with desired parameters
- Name, a size for the volume (1GB is the minimum)
- You can also use existing filesystem image as volume



Persistent storage –Volume attachment

- After the volume was created it can be attached to a running virtual machine (one at the time)
- Attachment can be made in Horizon **Volumes** page by selecting **Edit attachments**
- The virtual machine for the volume attachment should be selected
- The valid device name must be provided even when it will be ignored



Persistent storage –Volume use

- A newly created volume needs to be initialized first time it is being used
- The name of the volume can be determined and partition created by issuing fdisk commands inside the instance
- The filesystem for the partition can be created by issuing mke2fs command after the partition was created
- The filesystem can then be taken into use by mounting it
- Before detaching it the volume should be unmounted

```
pakivika@pak-ubuntu:~$ sudo fdisk /dev/vdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xb3f385a0.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): p

Disk /dev/vdb: 2154 MB, 2154823680 bytes
16 heads, 63 sectors/track, 4175 cylinders, total 4208640 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xb3f385a0

   Device Boot      Start         End      Blocks   Id  System

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4208639, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-4208639, default 4208639):
Using default value 4208639

Command (m for help): w
```

```
pakivika@pak-ubuntu:~$ sudo mke2fs /dev/vdb1
```

```
pakivika@pak-ubuntu:~$ sudo mount /dev/vdb1 /mnt/
pakivika@pak-ubuntu:~$ ls /mnt/
lost+found
pakivika@pak-ubuntu:~$
```

Exercise 4 – Use persistent storage

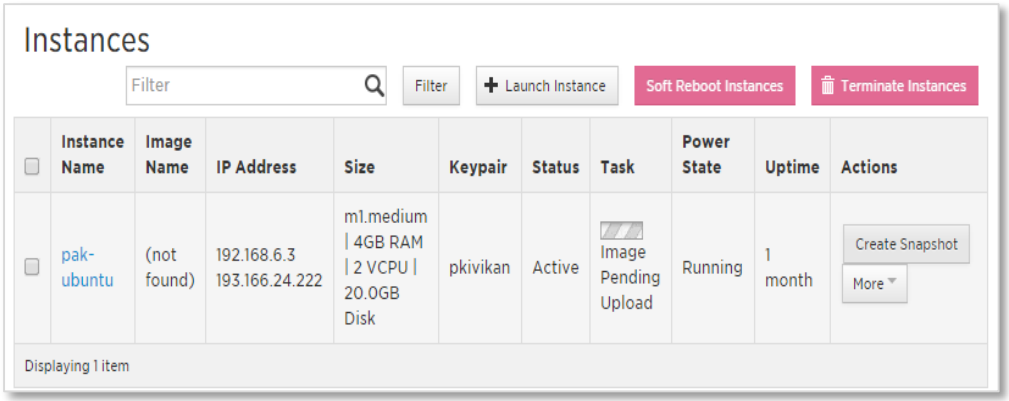
EXERCISE

- Search instructions from wiki about how to use persistent storage
  - Login to https://support.forgeservicelab.fi
  - Tip: Search for ”Using persistent storage”
- Use volume
  - Login to https://cloud.forgeservicelab.fi
  - Create a new 2GB volume
  - Attach it to your running instance
  - Login to your instance
    - `ssh ubuntu@public-ip -i key.pem`
  - List disks and identify newly attached disk based on the size
    - `sudo fdisk -l`
  - Create a new partition and filesystem for it then mount it
    - `sudo fdisk -l`
    - `sudo mke2fs /dev/vdX1`
    - `sudo mount /dev/vdX1 /mnt`
  - List the files of the new filesystem and see what you got there

Instances – Snapshots

- The state of the instance can be backed up by using snapshots
- A snapshot captures the state of the file system, but not the state of the memory
- Buffers should be flushed and filesystem state frozen using util-linux tools to ensure the snapshot contains desired data
- A snapshot can be taken by using Horizon **Instances** page and by selecting **Create Snapshot**
- A snapshot can be downloaded by using glance CLI tools. Note! CLI tools is the subject for another training

```
pakivika@pak-ubuntu:~$ sync
pakivika@pak-ubuntu:~$ sudo apt-get install util-linux
Reading package lists... Done
Building dependency tree
Reading state information... Done
util-linux is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 50 not upgraded.
pakivika@pak-ubuntu:~$ sudo fsfreeze -f /mnt/vdb
```




```
[pkivikan@forgeadmin ~]$ glance image-download --file snapshot.raw f30b204e-1ce6-40e7-b8d9-b353d4d84e7d
```

Exercise 5 – Remove unused resources

EXERCISE

- Login to https://cloud.forgeservicelab.fi
- Check the overview of your computing resource usage
- Login to your running instance
  - `ssh ubuntu@`*`public-ip`* `-i` *`key.pem`*
- Unmount the volume using unmount command inside the instance, then logout
- Terminate the instance using Horizon
- Check if volume still persists using Horizon
- Delete the volume using Horizon
- Delete the security group using Horizon
- Delete the snapshot (if you created one)
- Check the overview of your computing resource usage

DIGILE

FORGE Service Lab

DIGILE in a Nutshell

- **DIGILE** is the Center for Science, Technology and Innovation (SHOK) focusing on Internet economy and related technologies and business
- **Mission:** DIGILE creates Internet economy competencies to enable new global business and job growth for DIGILE's stakeholders and partners
- **Three main services**:
  - **Research:** Cooperative national and international research programs to create new technological and business innovations
  - **Solutions:** Facilitation of business ecosystems and lead solution creation to explore new global business opportunities
  - **Digital service creation:** FORGE Service Lab for fast digital service creation and competence scaling
- **Core enablers**:
  - International networking
  - Operative excellence
  - Co-creation leadership

FORGE Service Lab
WHAT, WHY, WHO, FOR WHOM

- **WHAT:** FORGE Service Lab is a laboratory for creating digital services in the Internet-era. It is intended as a tool to accelerate the creation of digital services in Finland - *from an idea to a scalable implementation*.

- **WHY:** Internet economy will grow stronger and digitalisation spreads across all industries. Most of the value is being created via digital services. As a result, digital services know-how needs to become one of the nation's core competencies.

- **WHO:** DIGILE, CSC-IT Center for Science, Kainuun Etu Oy with the Ministry of Traffic and Communication, the financing partner for the ramp-up

- **FOR WHOM**: To all who are interested in developing digital services e.g. businesses, educational institutions, business development teams, the public sector – all industries and government sectors are included.

FORGE Service Lab – Offering

Legal & Contract framework for each stakeholder: service developers and partners

Offers wide development framework for service projects where multiple stakeholders and partners can share openly the knowledge and develop efficiently globally recognisible successfull services

Partner network from multidisciplinary perspectivie: eg. Business development, Service Design, Technical development

Reference model for the creation of digital services, from the idea to the scalable implementation

Crowdsourcing methods ad tools which enables to create as meaningful and successfull service as possible from the end users perspective

Cloud computing platform for agile and fast ways to develop and test the services

Guidance and support for the project during the service creation path in order to manage the big picture

More information

- Documentation
  - https://support.forgeservicelab.fi/redmine/projects/forge-support/wiki
- Support tickets
  - https://support.forgeservicelab.fi/redmine/projects/forge-support/issues/new
- XMPP
  - forge-support@xmpp.forgeservicelab.fi
- Email
  - support@forgeservicelab.fi

THANK YOU