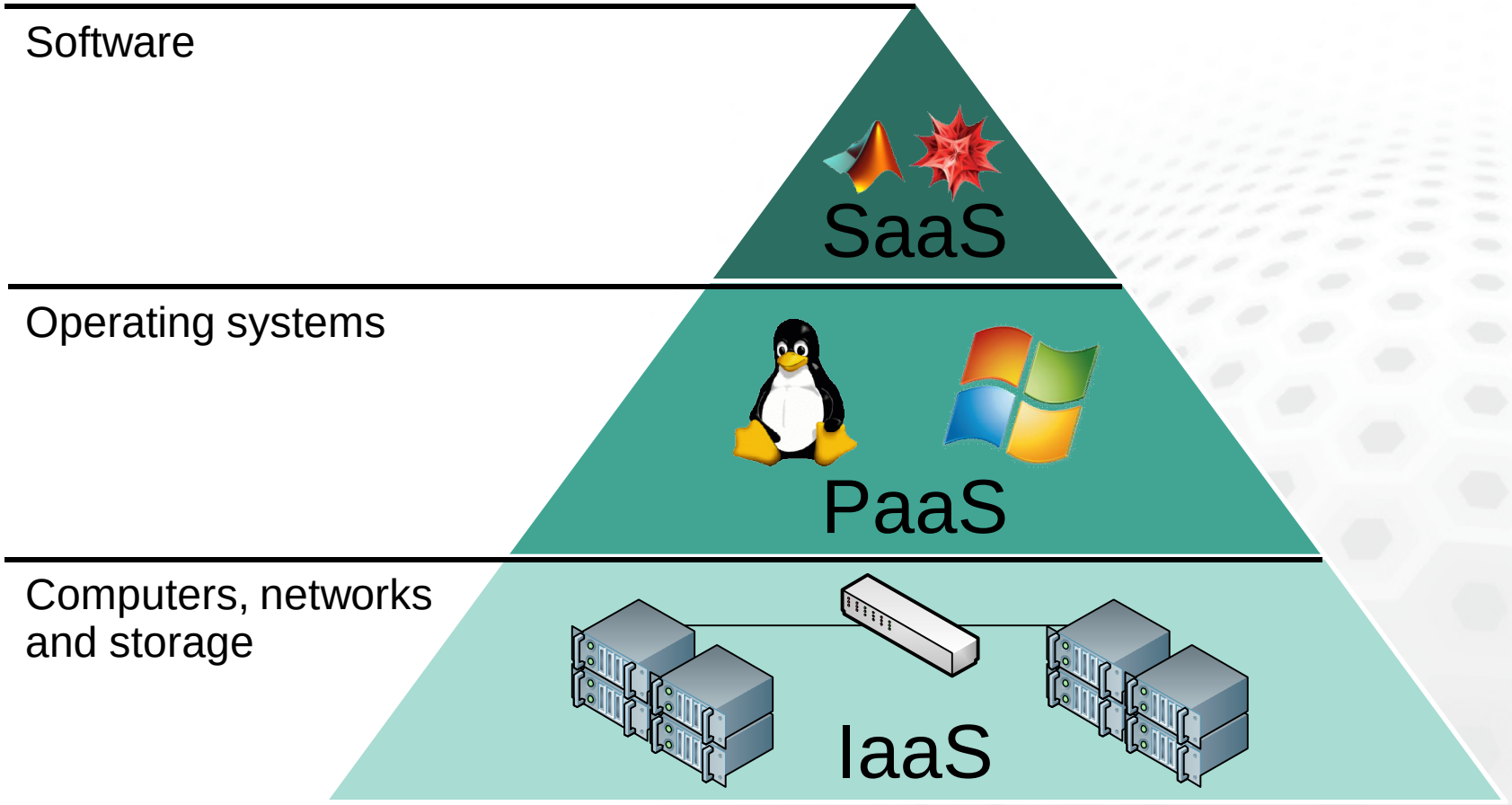
CSC Pouta Cloud Course

Risto Laurikainen

What cloud?

- Terminology overload, used to mean e.g.:
  - Storage services (Dropbox)
  - Virtual server hosting (Amazon Web Services)
  - Software platforms (Google App Engine)
  - Pretty much any web service
  - The Internet as a whole
- Self-service and automation are the common features

Cloud computing: three service models



The topic of this course: IaaS

- The user manages their own
  - Servers
  - Networks
  - Storage
- The resources are typically virtualized
- The user has full admin access to their own virtual resources

Traditional HPC vs. IaaS

	Traditional HPC environment	Cloud environment Virtual Machine
Operating system	Same for all: CSC's cluster OS	Chosen by the user
Software installation	Done by cluster administrators Customers can only install software to their own directories, no administrative rights	Installed by the user The user has admin rights
User accounts	Managed by CSC's user administrator	Managed by the user
Security e.g. software patches	CSC administrators manage the common software and the OS	User has more responsibility: e.g. patching of running machines
Running jobs	Jobs need to be sent via the cluster's Batch Scheduling System (BSS)	The user is free to use or not use a BSS
Environment changes	Changes to SW (libraries, compilers) happen.	The user can decide on versions.
Snapshot of the environment	Not possible	Can save as a Virtual Machine image
Performance	Performs well for a variety of tasks	Very small virtualization overhead for most tasks, heavily I/O bound

Cloud service development in 2014

- **Pouta** = CSC cloud service

  https://research.csc.fi/cloud-computing

  – cPouta (in production): ”Amazon-type” cloud for research communities and organisations

  – ePouta (in development): Enterprise virtual hosting with a focus on security

  – Both are based on OpenStack

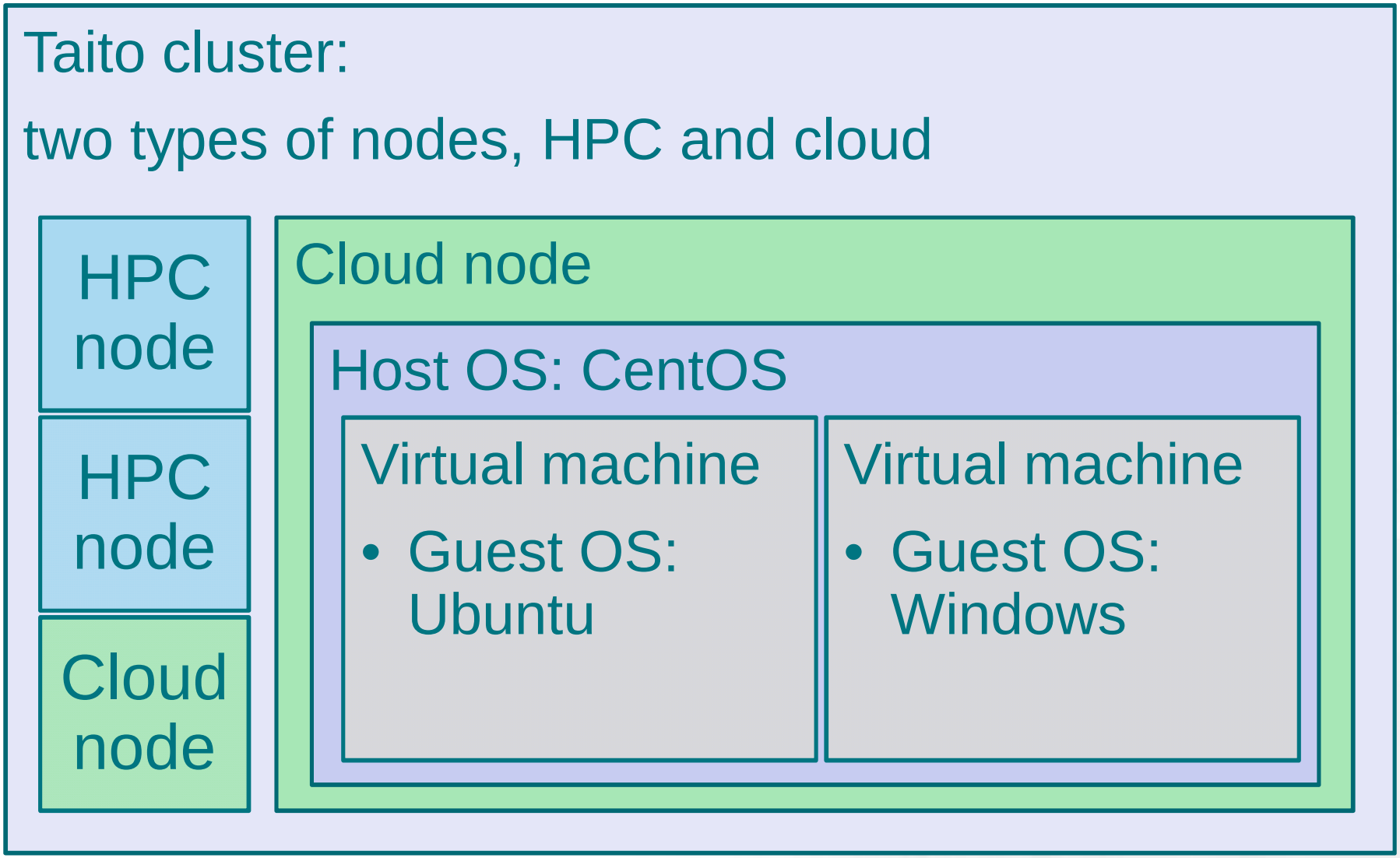
Pouta's use cases

- Enhanced security – isolated virtual machines
- Advanced users – able to manage servers
- Difficult workflows – can't run on Taito
- Complex software stacks
- Ready made virtual machine images
- Deploying tools with web interfaces
- "We need root access"

If you can run on Taito – run on Taito
If not – Pouta might be for you

- Pouta user guide: *https://research.csc.fi/pouta-user-guide*

Virtualization in Taito



Virtual machine flavors in Pouta

Name	Cores	Memory (GB)	Local disk (total, GB)	RAM/core
tiny	1	1	120	1
small	4	15	230	4
medium	8	30	450	4
large	12	45	670	4
fullnode	16	60	910	4

ePouta

- Renewing the cloud cluster equipment in Espoo in 2015
  - Changes to OpenStack cloud middleware (autumn 2014)
  - Focus on secure computing and service for organisations
  - Idea: seamless scaling of local resources using a trusted compute center in Finland
  - Requires local IT admin contact
  - Funding model and resource allocation policy is still under debate, supported by ELIXIR Finland

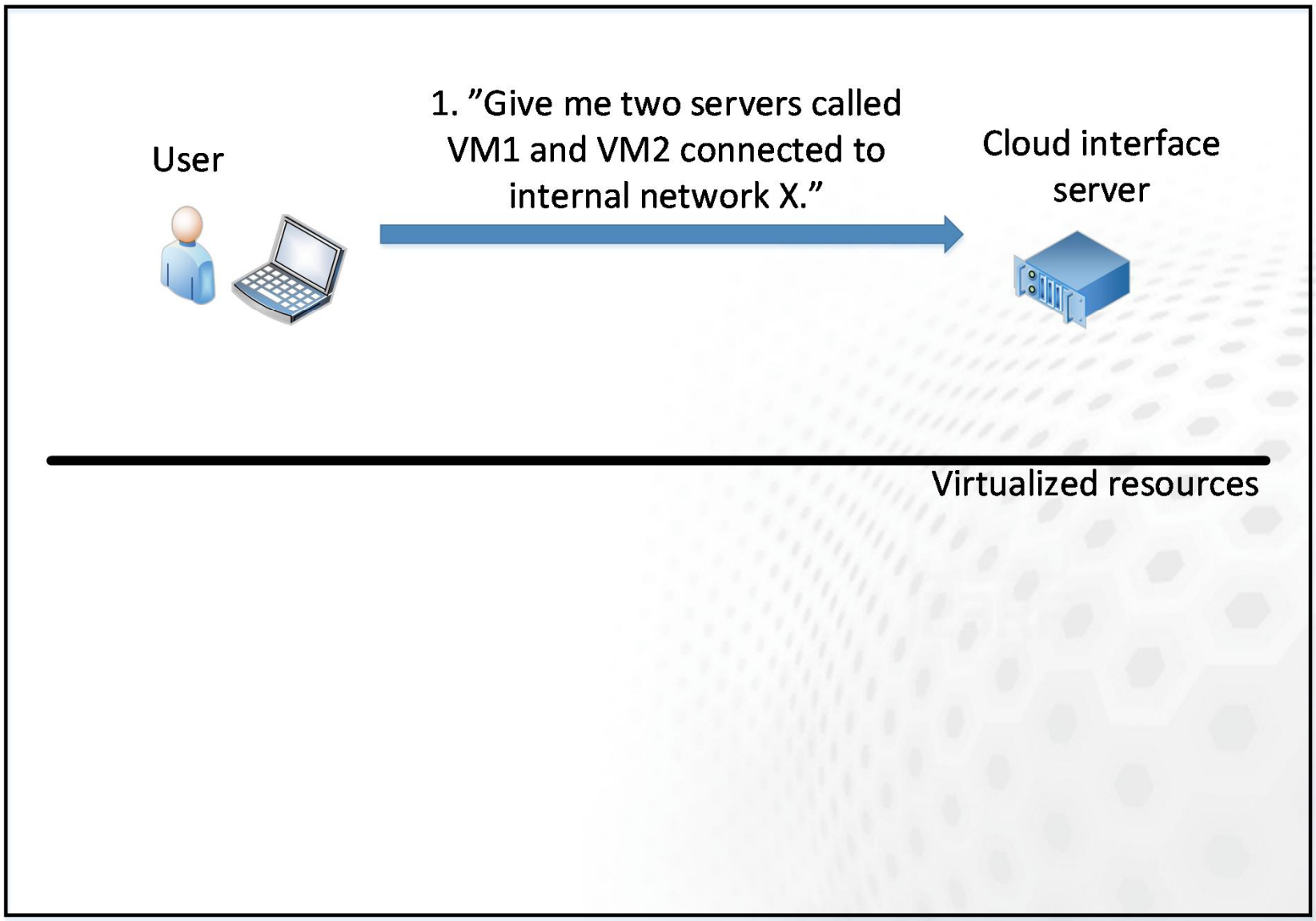
Cloud contact information

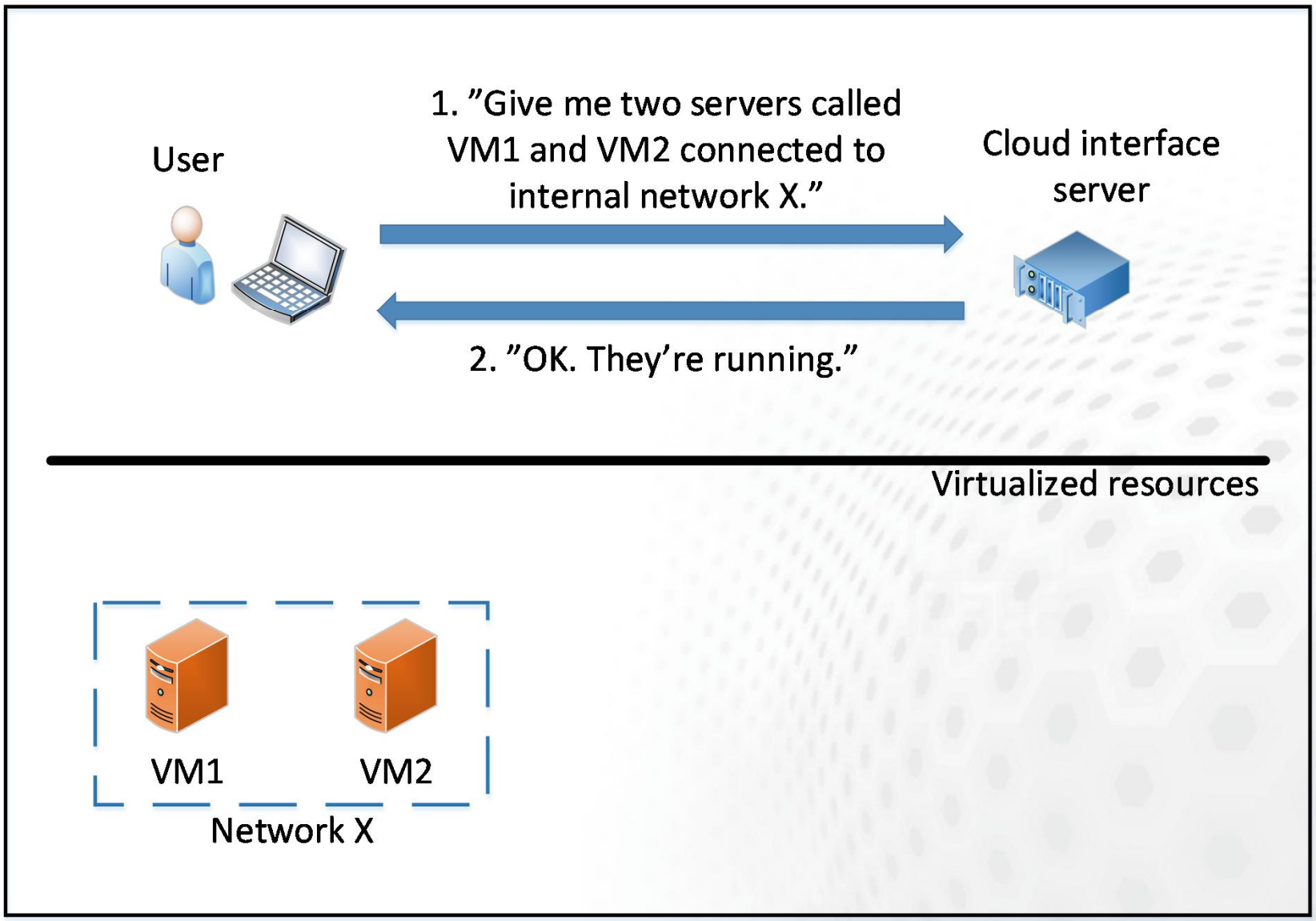
- Support: cloud-support@csc.fi
- Documentation: https://research.csc.fi/pouta-user-guide
- Contacts (firstname.lastname@csc.fi):
  - Project manager: Peter Jenkins
  - Technical contacts
    - Risto Laurikainen
    - Kalle Happonen

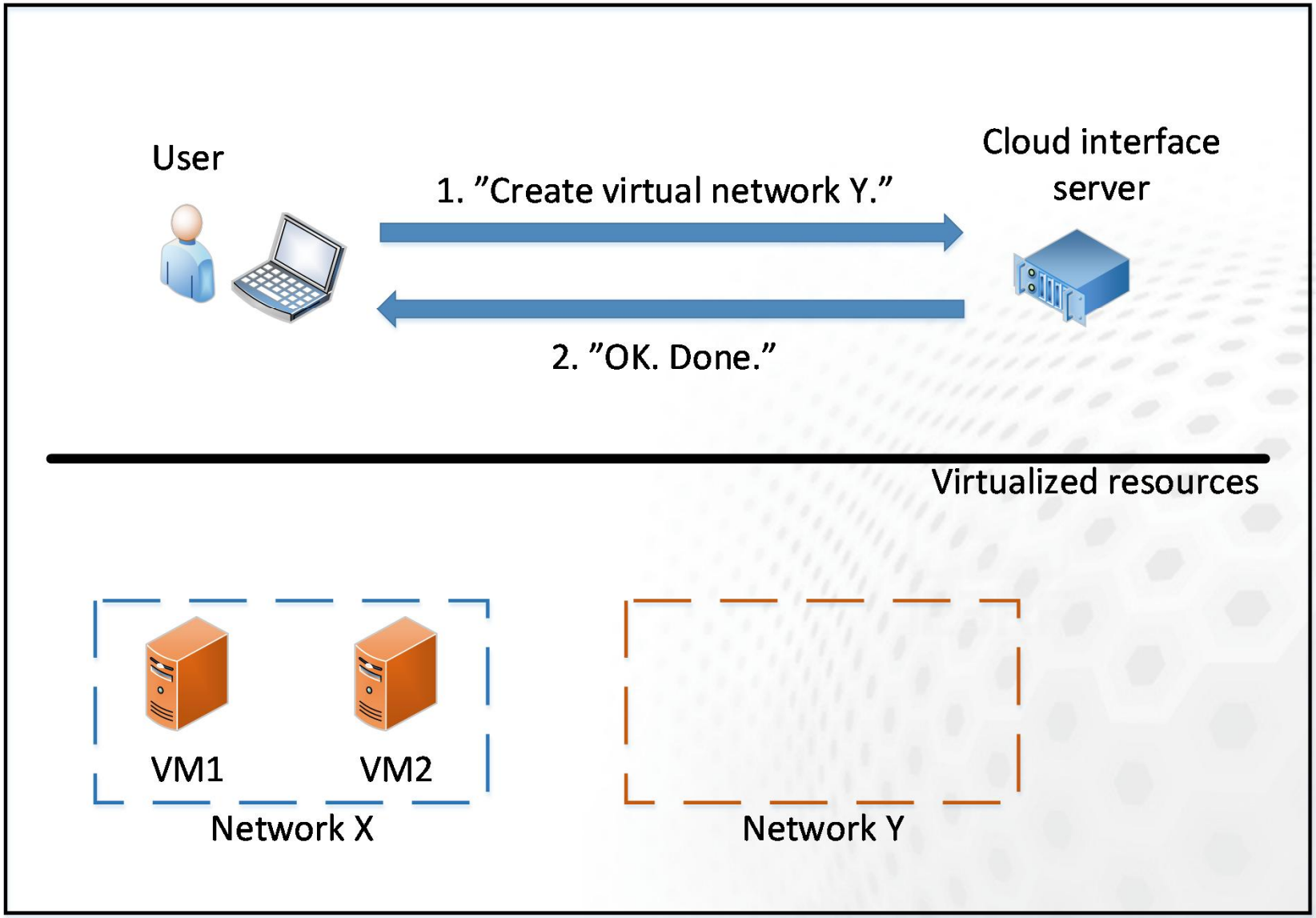
OpenStack

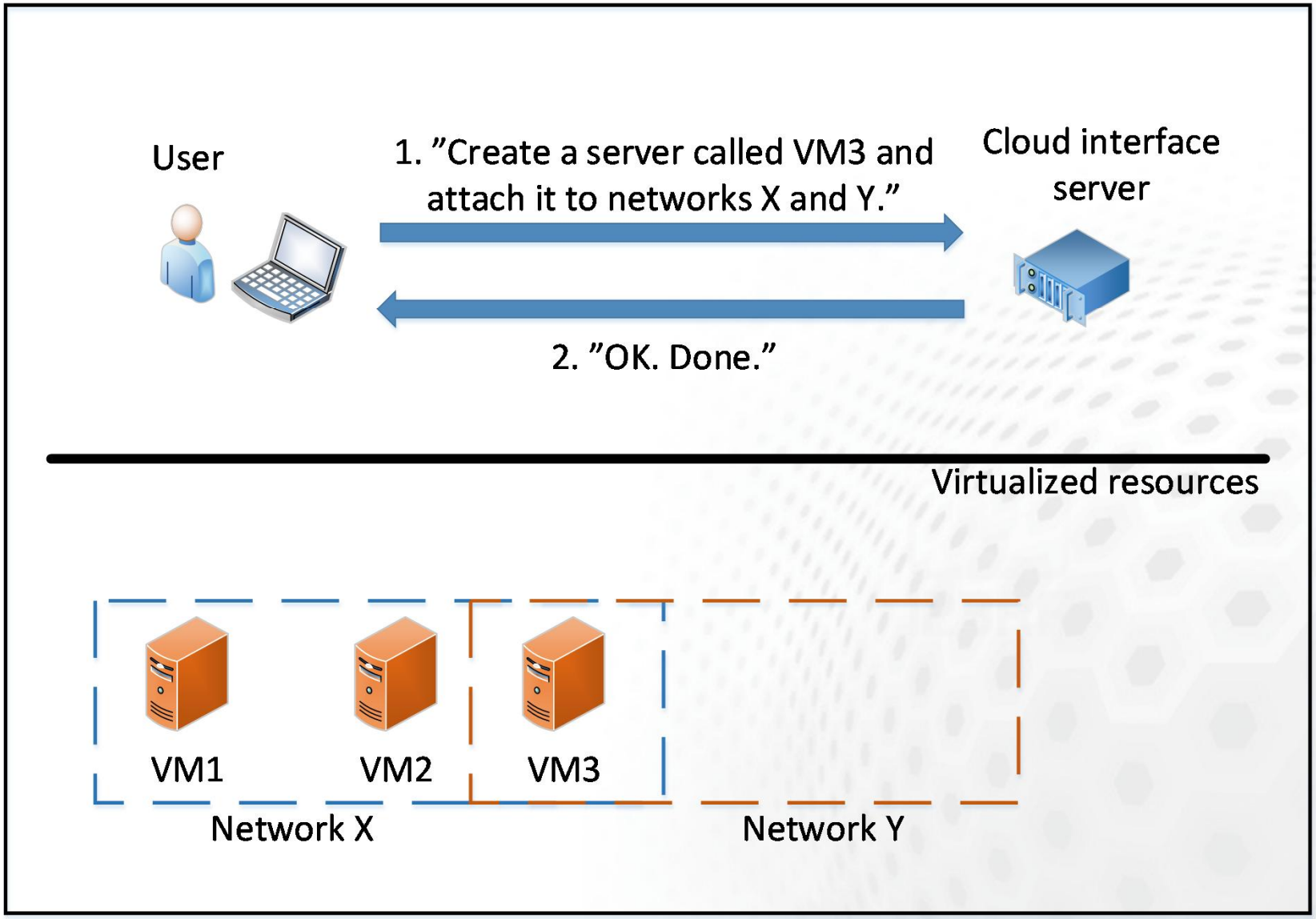
What OpenStack?

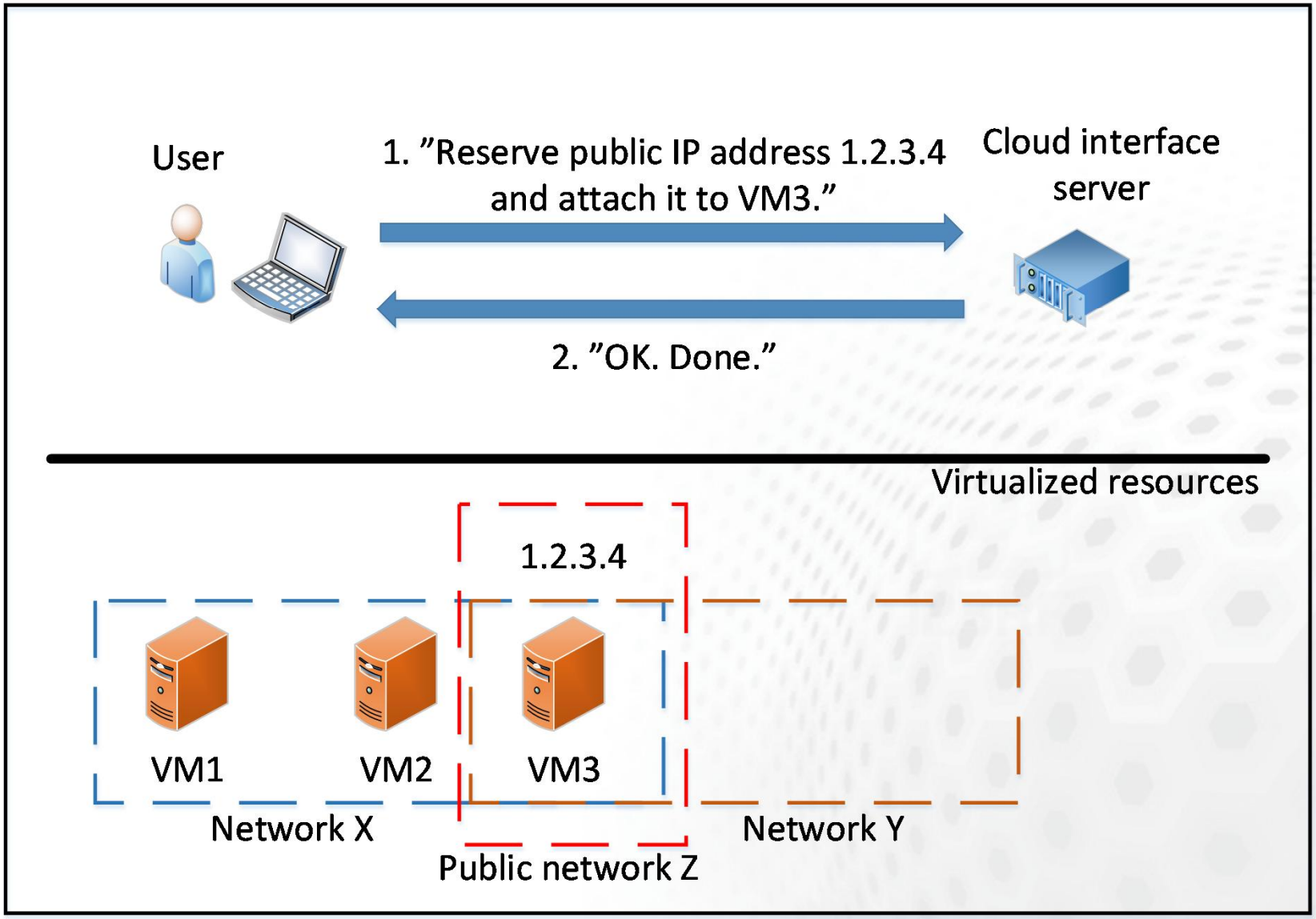
- Set of tools to build an IaaS cloud for creating virtualized
  - servers
  - networks
  - storage
- OpenStack is to the datacenter what Linux is to a server - an operating system
- Just like there are many Linux distributions, there are many OpenStack distributions

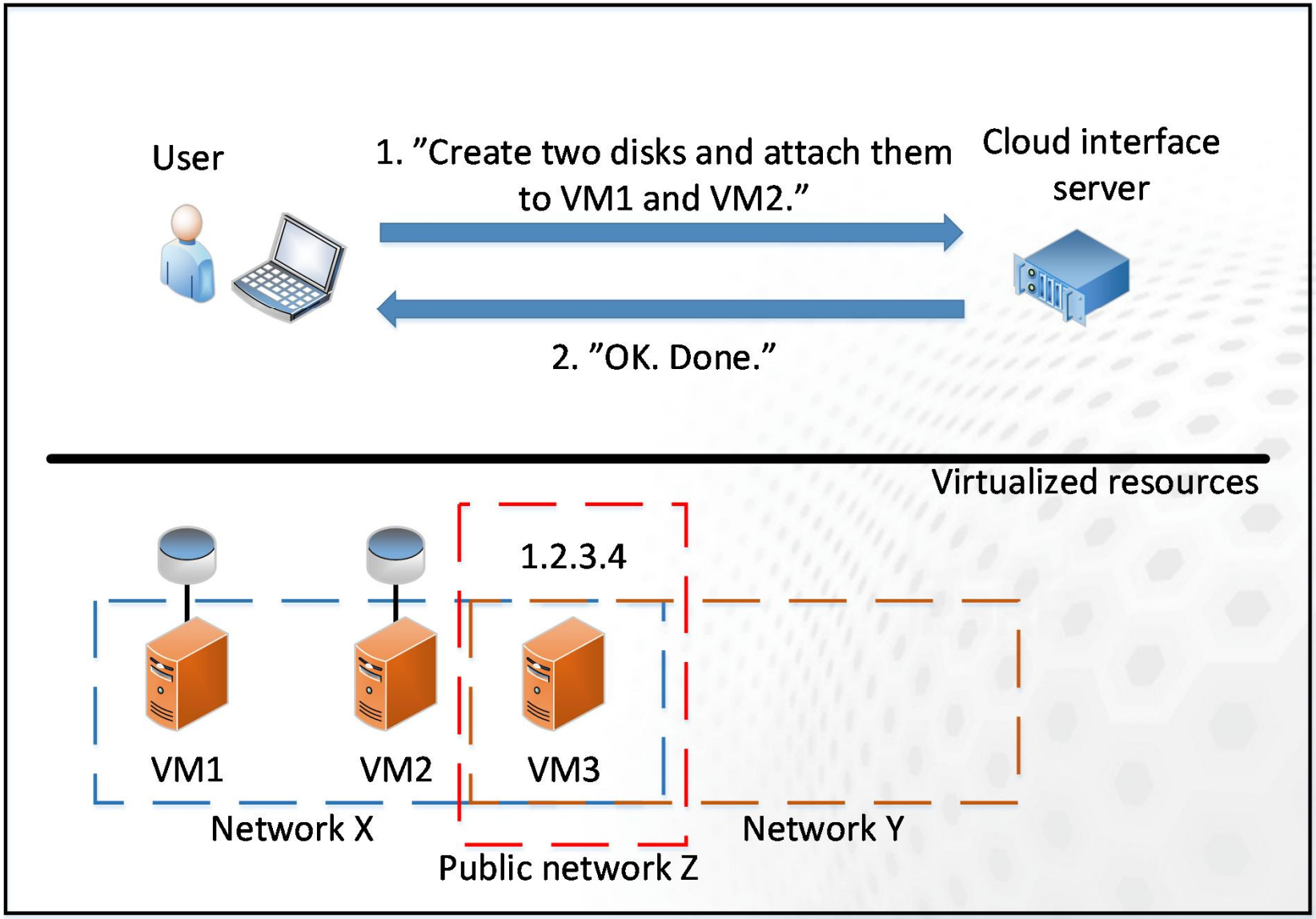

User
1. ”Give me two servers called VM1 and VM2 connected to internal network X.”
Cloud interface server
Virtualized resources

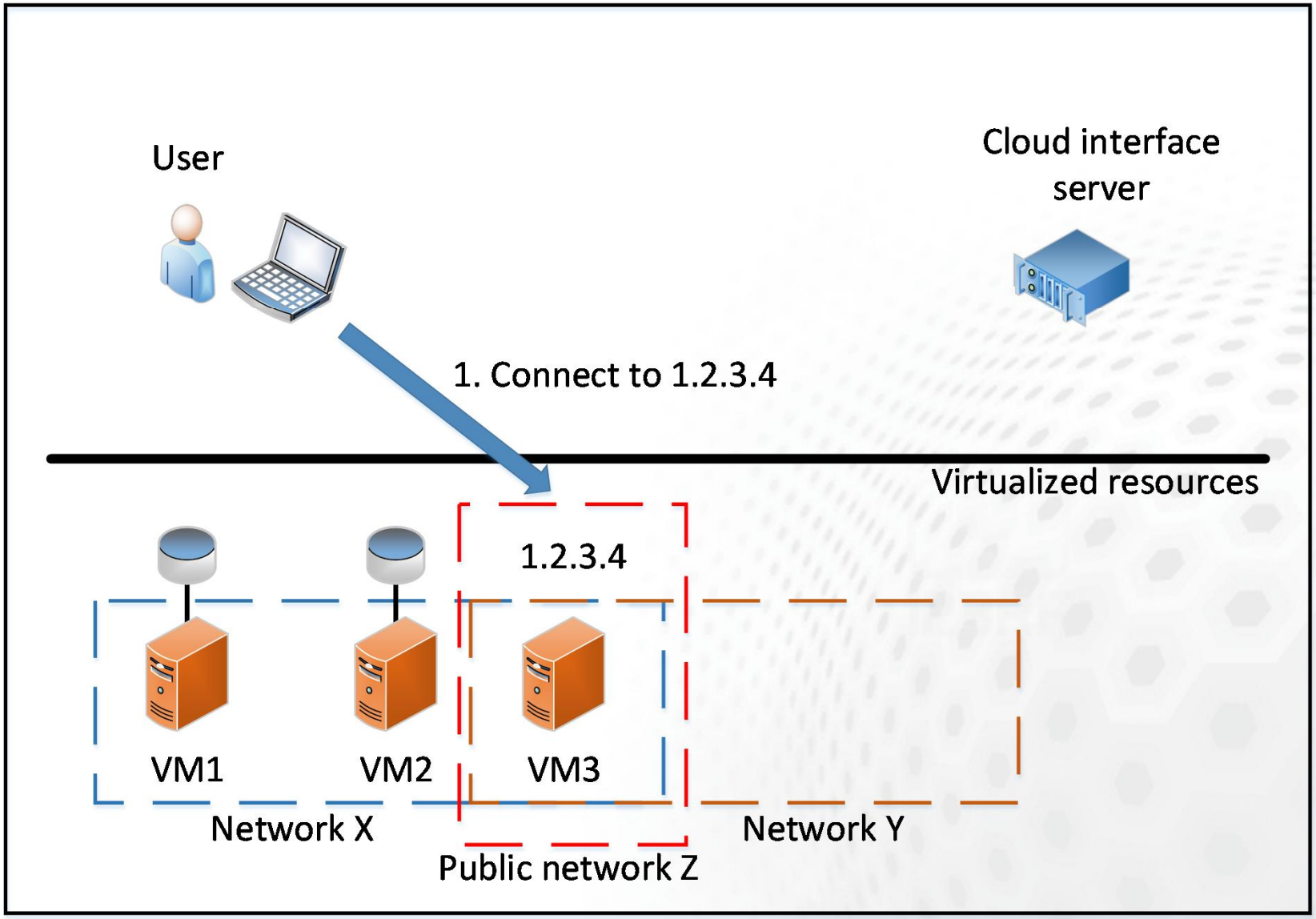

User
1. ”Give me two servers called VM1 and VM2 connected to internal network X.”
Cloud interface server
2. ”OK. They’re running.”
Virtualized resources
VM1
VM2
Network X


User
1. ”Create virtual network Y.”
Cloud interface server
2. ”OK. Done.”
Virtualized resources
VM1
VM2
Network X
Network Y


User
1. ”Create a server called VM3 and attach it to networks X and Y.”
Cloud interface server
2. ”OK. Done.”
Virtualized resources
VM1
VM2
VM3
Network X
Network Y


User
1. ”Reserve public IP address 1.2.3.4 and attach it to VM3.”
Cloud interface server
2. ”OK. Done.”
Virtualized resources
1.2.3.4
VM1
VM2
VM3
Network X
Network Y
Public network Z


User
1. ”Create two disks and attach them to VM1 and VM2.”
Cloud interface server
2. ”OK. Done.”
Virtualized resources
1.2.3.4
VM1
VM2
VM3
Network X
Network Y
Public network Z


User
Cloud interface server
1. Connect to 1.2.3.4
Virtualized resources
1.2.3.4
VM1
VM2
VM3
Network X
Network Y
Public network Z

Interfaces

- Web
  - Works from any modern browser
  - Launch, list, terminate servers
  - Server console in the browser
  - Manage storage and networks
- Command line
  - Can do all the same things as the web interface and more
- API
  - Management through a programmable interface

Project | Admin

CURRENT PROJECT
csc

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

Manage Network

- Networks
- Routers
- Network Topology

Overview

Logged in as: rlaurika | Settings | Help | Sign Out

Quota Summary

Used 26 of 501 Available Instances

Used 170 of 256 Available vCPUs

Used 623,200 MB of 1,024,000 MB Available RAM

Used 16 of 100 Available volumes

Used 3,541 GB of 10,000 GB Available volume storage

Select a month to query its usage:

January | 2014 | Submit

Active Instances: 25 **Active RAM:** 563GB **This Month's VCPU-Hours:** 16232.76 **This Month's GB-Hours:** 5223808.40

Usage Summary

Download CSV Summary

Instance Name	VCPUs	Disk	RAM	Uptime
Galaxy-test	16	210	58GB	6 months
karan-ceph-admin	1	10	1GB	3 months
karan-ceph-mon2	4	230	15GB	3 months

Instances

Logged in as: rlaurika Settings Help Sign Out

Project Admin

CURRENT PROJECT
csc

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

Manage Network

- Networks
- Routers
- Network Topology

Instances

+ Launch Instance | Terminate Instances

Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions
olli_test3	192.168.1.19 86.50.168.30	medium | 30GB RAM | 8 VCPU | 10GB Disk	olli_bombay	Active	None	Running	Create Snapshot More
kalletest	192.168.1.22	tiny | 1GB RAM | 1 VCPU | 10GB Disk	kalle	Active	None	Running	Create Snapshot More
lalves_test	192.168.1.21	tiny | 1GB RAM | 1 VCPU | 10GB Disk	lalves	Active	None	Running	Create Snapshot More
pj-ubuntu	192.168.1.2 86.50.168.10	small | 15GB RAM | 4 VCPU | 10GB Disk	pj-keys	Active	None	Running	Create Snapshot More
HarriPerformanceTests_1_4	192.168.1.29 86.50.168.26	tiny | 1GB RAM | 1 VCPU | 10GB Disk	keypair-harri	Active	None	Running	Create Snapshot More
HarriPerformanceTests_1_3	192.168.1.26 86.50.168.22	tiny | 1GB RAM | 1 VCPU | 10GB Disk	keypair-harri	Active	None	Running	Create Snapshot More

Instance Detail: olli_test3

Logged in as: rlaurika Settings Help Sign Out

Project Admin

CURRENT PROJECT
csc

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

Manage Network

- Networks
- Routers
- Network Topology

Overview Log Console

Instance Console

If console is not responding to keyboard input: click the grey status bar below. Click here to show only console

Connected (encrypted) to: QEMU (instance-000017bd) Send CtrlAltDel

```
CentOS release 6.5 (Final)
Kernel 2.6.32-431.3.1.el6.x86_64 on an x86_64

192-168-1-19 login: _
```

```
rlaurika@pilkkasiipi ~ $ nova list
+--------------------------------------+---------------------------+-----------+-------------------------------+
| ID                                   | Name                      | Status    | Networks                      |
+--------------------------------------+---------------------------+-----------+-------------------------------+
| ca02b61d-656e-479f-bde2-7bb8b58add0f | Galaxy-test               | SUSPENDED | csc=192.168.1.18              |
| b1884673-3844-4c22-825e-4a0567ed8b3b | HarriPerformanceTests_1_3 | ACTIVE    | csc=192.168.1.26, 86.50.168.22 |
| e37bb795-57de-4e88-93bc-e91b65a5f77a | HarriPerformanceTests_1_4 | ACTIVE    | csc=192.168.1.29, 86.50.168.26 |
| a4d94367-971f-4ec4-af8d-18384dd9bb84 | Image builder             | ACTIVE    | csc=192.168.1.25, 86.50.168.6  |
| 393c4f74-0964-4029-94f9-f871e8ace721 | JarnoTest                 | ACTIVE    | csc=192.168.1.20, 86.50.168.64 |
| df5dee5f-76df-4f91-b3c4-1c6112f9dfec | TestUserTest              | SHUTOFF   | csc=192.168.1.44              |
| 82fd0a09-9fc7-449f-843b-1298cad52bbe | ceph-node4                | ACTIVE    | csc=192.168.1.45              |
| 37efdee1-11a6-4b0a-9297-2682ced4f681 | ceph-node5                | ACTIVE    | csc=192.168.1.46              |
| 7ca99cd8-c9ec-45bc-ad6c-09013049b8cd | ceph-node6                | ACTIVE    | csc=192.168.1.47              |
| 71adc582-6d4c-4bb2-ae03-a3feec5213c3 | chipster-test             | ACTIVE    | csc=192.168.1.5, 86.50.168.39  |
| 6f3c82af-7f42-40dd-98da-db4ba12b960c | chipster-test-from-image  | ERROR     |                               |
| 9300313c-338a-4327-890e-4d02d1821bf2 | fail2ban-test Johan       | ACTIVE    | csc=192.168.1.27, 86.50.168.8  |
| c250a4ee-6323-4246-a18e-5b5ab1f1882d | kalletest                 | ACTIVE    | csc=192.168.1.22              |
| 51c0a65d-1aba-4cc0-b565-ca85bd19c61b | karan-RDO                 | ACTIVE    | csc=192.168.1.49, 86.50.168.33 |
| 1e1999c5-b6fe-44f8-8960-91d47e300727 | karan-ceph-admin          | ACTIVE    | csc=192.168.1.28, 86.50.168.70 |
| 2373fa4f-95e9-4bb2-b80e-feb2b0379415 | karan-ceph-client1        | ACTIVE    | csc=192.168.1.40              |
| 0db4f304-3cd2-4337-b0e2-934ff74082fd | karan-ceph-mon1           | ACTIVE    | csc=192.168.1.38              |
| 5825e7f5-fe91-4889-a32d-298b24168c20 | karan-ceph-mon2           | ACTIVE    | csc=192.168.1.33              |
| aea288a5-2042-4311-9010-dbf686070246 | karan-ceph-mon3           | ACTIVE    | csc=192.168.1.31              |
| b88b0360-9557-4942-8a2d-e2a597f93f9f | karan-ceph-node1          | SUSPENDED | csc=192.168.1.34              |
| 28242b30-8b64-4186-9dc3-6834e5037d84 | karan-ceph-node2          | ACTIVE    | csc=192.168.1.37              |
| 618e23e5-f8d0-49a0-89da-2b89c4a008de | karan-ceph-node3          | ACTIVE    | csc=192.168.1.41              |
| 2b8164d8-cbe2-4143-85f6-67f1598ccdce | karan-ceph-puppetmaster   | ACTIVE    | csc=192.168.1.4               |
| 20dd95ad-4178-4e5b-9097-912a396bc6bd | lalves_test               | ACTIVE    | csc=192.168.1.21              |
| c7eb0d54-12b9-4124-baf5-7cf2459320d4 | olli_test3                | ACTIVE    | csc=192.168.1.19, 86.50.168.30 |
| fade122d-d763-4354-ae7d-79e235421baf | pj-ubuntu                 | ACTIVE    | csc=192.168.1.2, 86.50.168.10  |
+--------------------------------------+---------------------------+-----------+-------------------------------+
rlaurika@pilkkasiipi ~ $
```

Storage types in OpenStack

- OS image
  - The root disk of the VM
  - Usually not very large for efficiency reasons
- Ephemeral disk = scratch
  - Throw-away scratch disk
  - Disappears when VM instance is deleted
- Volumes = persistent block storage
  - Persistent disk for storing hot data
  - Can be attached and detached to/from a running VM
- Swift = reliable object storage
  - Replicated storage for cold data
  - Accessed over HTTP
- Still missing: shared file system (CIFS,NFS,..)

Tips for the efficient use of IaaS

The most obvious workflow when using a cloud

1. Start a virtual machine
2. Login
3. Configure some software using the command line
   - Install some packages
   - Edit a few configuration files
   - Make a few changes to the firewall
   - Start some services
4. Done!

What needs to fail for this workflow to fail? Just one of these:

Some recommendations

- Automate as much as possible
- Separate configuration from state

Automate as much as possible

- If something goes wrong, manual recovery may be difficult or impossible
- Make it easy to recreate your VMs from scratch
- Configuration management helps. Some tools for that:
  - Ansible
  - Puppet
  - Chef

Separate configuration from state

- Configuration is installed software, configuration files, firewall rules etc.
- State is e.g. data in a database or data produced by a computation
- Where to store each:
  - Configuration: VM's local filesystem
  - State: persistent volume (like a virtual hard drive attached to the VM)
- You should have a backup of both your state and your configuration

Ansible (http://www.ansible.com)

- Free and open source software for automating configuration tasks
- Easy to use
- No need to install anything on the machine to be configured - SSH is enough

Hands on exercises

Documentation:

https://research.csc.fi/pouta-user-guide

1. Setup prerequisites
   - SSH key
   - Security group
2. Launch a virtual machine (use CentOS 6.5)
3. Assign a floating IP to the VM
4. Login to the VM
5. Create a snapshot of the VM
6. Attach block storage