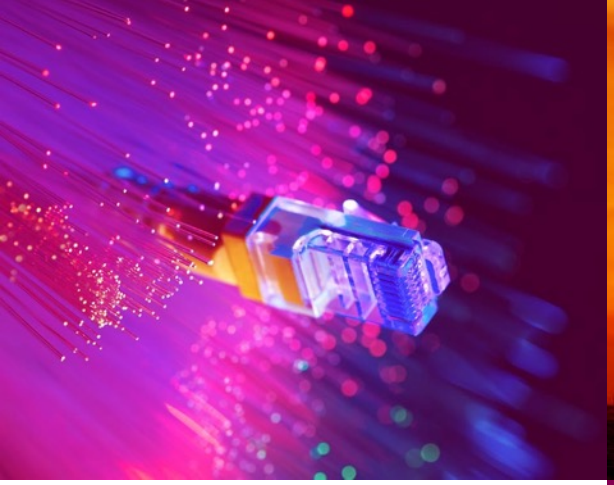


Pouta Cloud Course, Autumn 2019

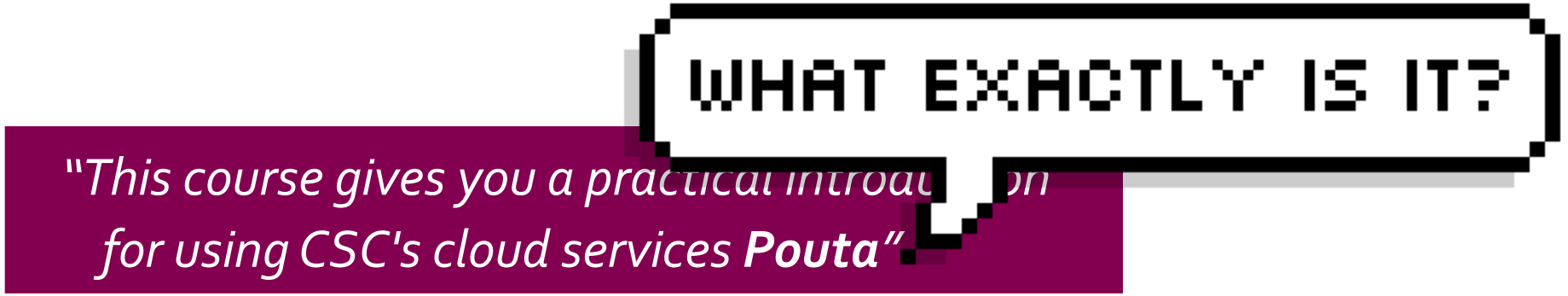
- Shubham Kapoor, Cloud System Specialist
- Juhani Kataja, Application Specialist

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Syllabus

- **9:00 - 10:30** Course introduction
  - Cloud computing basics
  - Cloud service landscape
  - Pouta virtual resource basics
- **10:30 - 10:45** Coffee break
- **10:45 - 12:00** Exercises
  - Creating virtual resources via Web User Interface
  - Use Security Groups to secure your virtual resources
- **12:00 - 13:00** Lunch
- **13:00 - 14:30** Theory
  - cPouta vs. ePouta
  - Using Object Storage & Persistent Volumes for safely storing your data
  - Pouta: Managing Project, Billing, VM lifecycles…
  - Building Application stack on your Pouta VM
- **14:30 - 14:45** Coffee break
- **14:45 - 16:00** Exercises
  - Quiz
  - Application stack development on your VM
  - Object Storage/Persistent Volume configurations and best practices
  - Snapshot of VM, OpenStack CLI tools…

Course Introduction



- CSC's Infrastructure as a Service (IaaS) Cloud offering based on OpenStack
- Allows running Virtual Machines (VMs) on CSC's Data Center infrastructure
- Grants users full control of OS, middleware & run time environments
  - On the flipside, users must manage and secure their VMs
- Provides VMs direct connection to the Internet/Intranet, allowing for new collaboration.
- Provides an IaaS cloud environment for your sensitive data processing (ePouta).

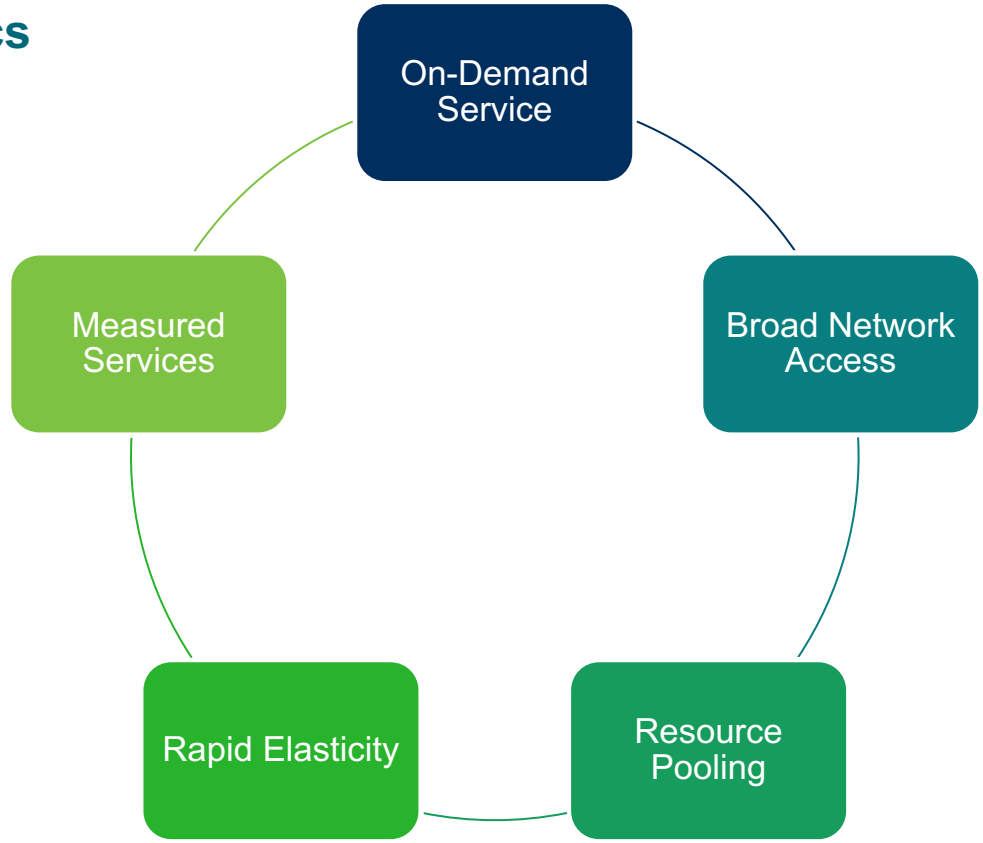
Cloud Computing

"Cloud Computing refers to on-demand delivery of computing services – servers, storage, databases, networking, software, analytics and more— over the network."

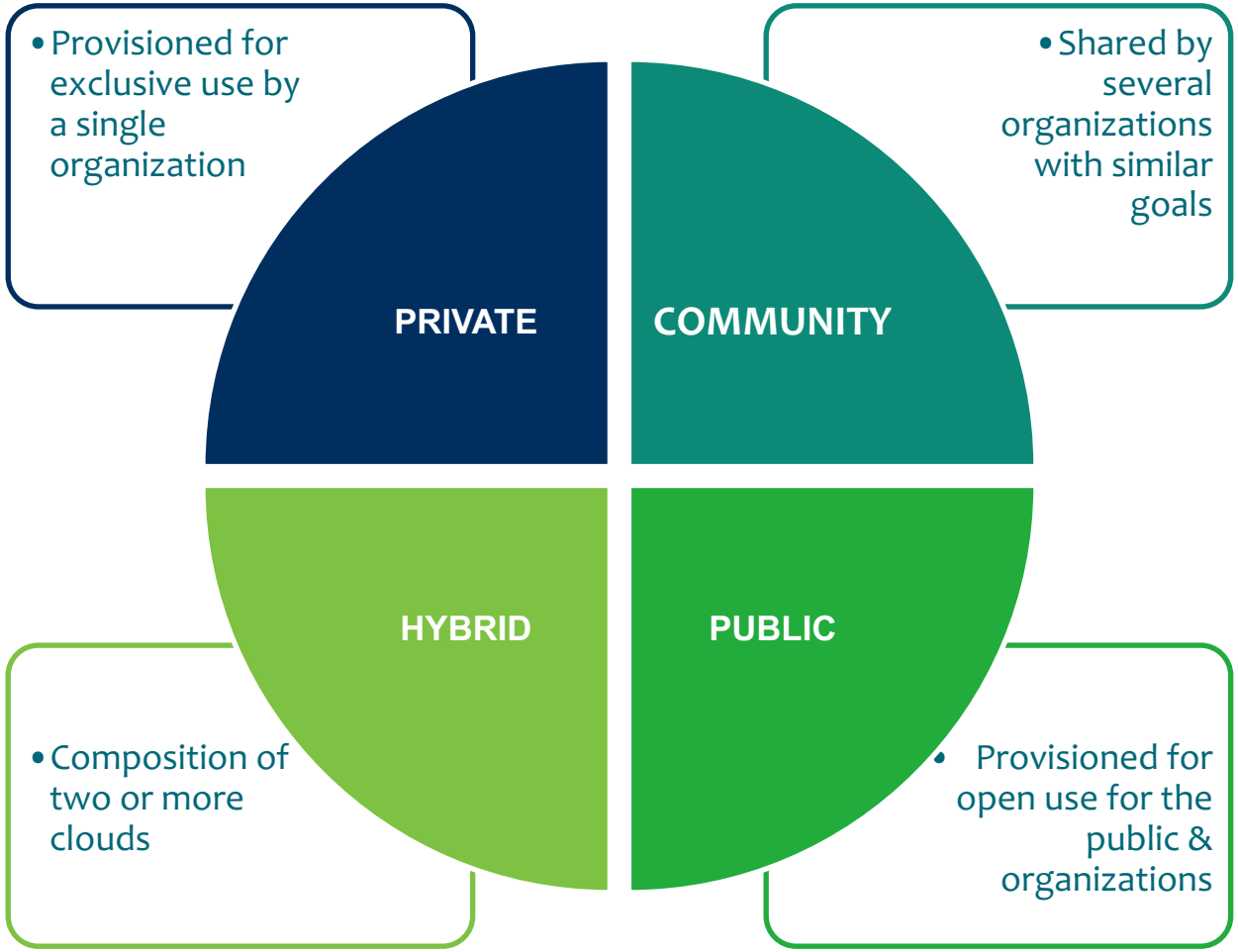
"A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

Cloud Computing Characteristics

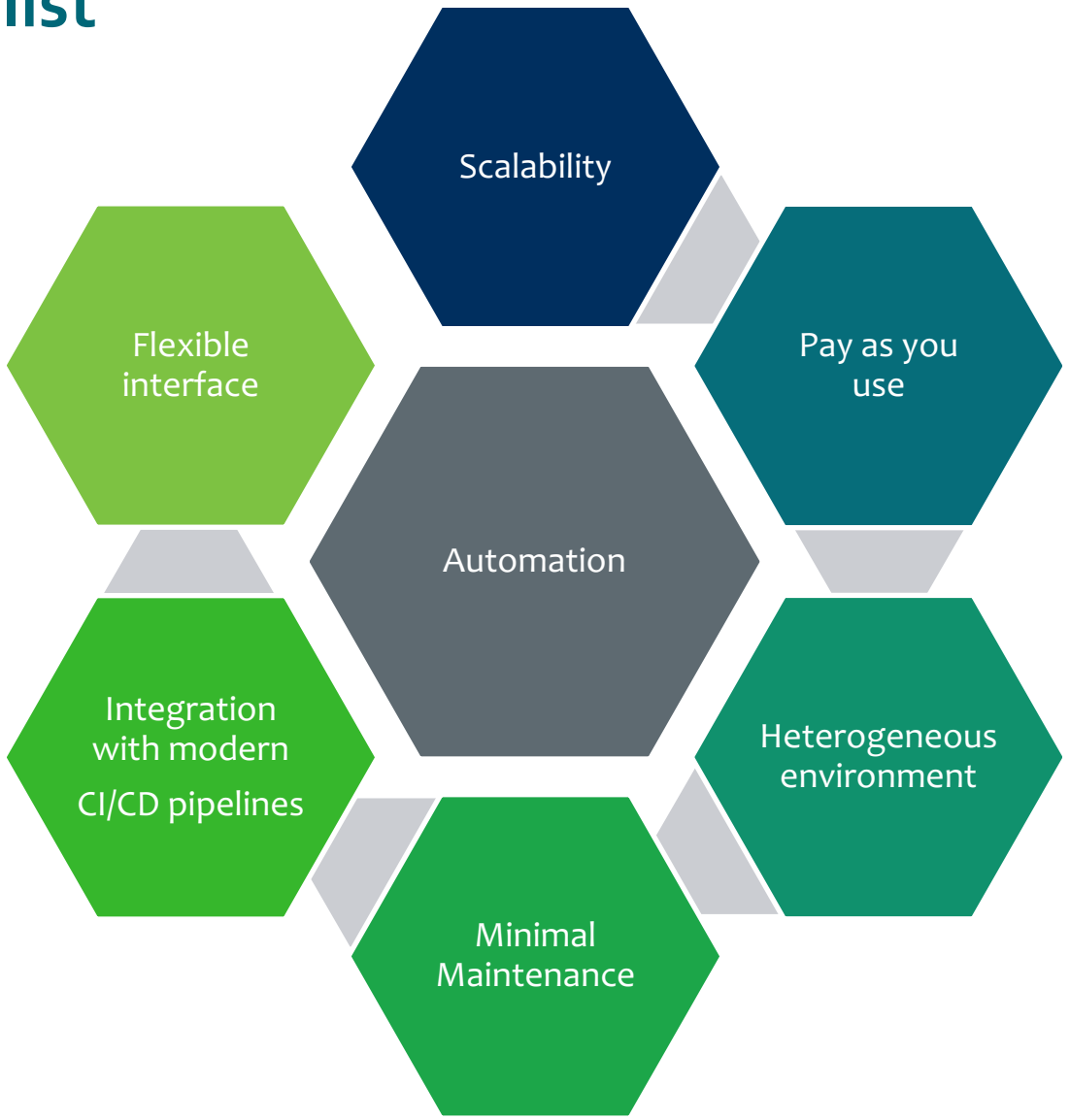
- **Cloud Computing has 5 essential characterstics**
  - On-demand self-service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured Service



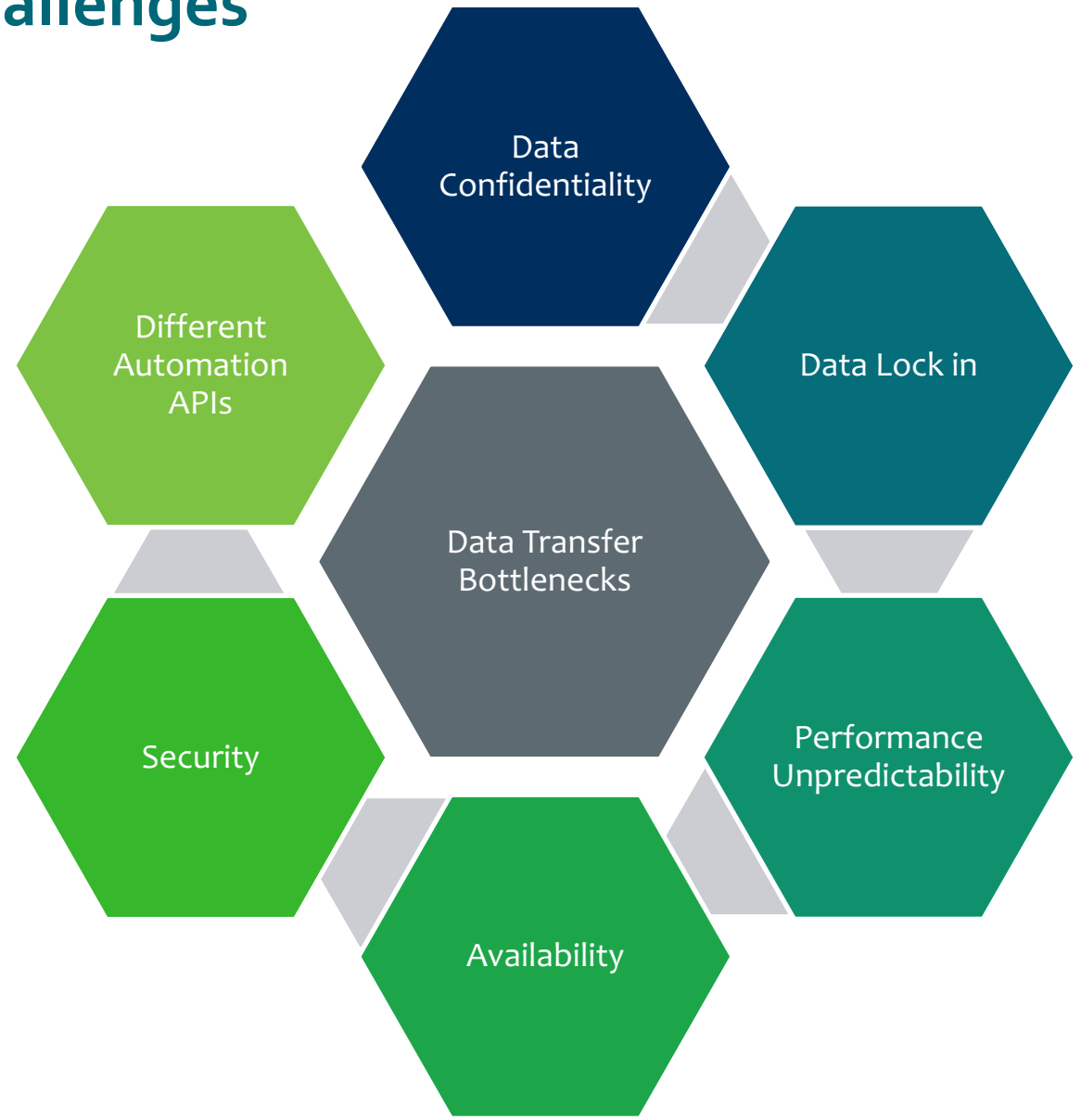
Cloud Deployment Models



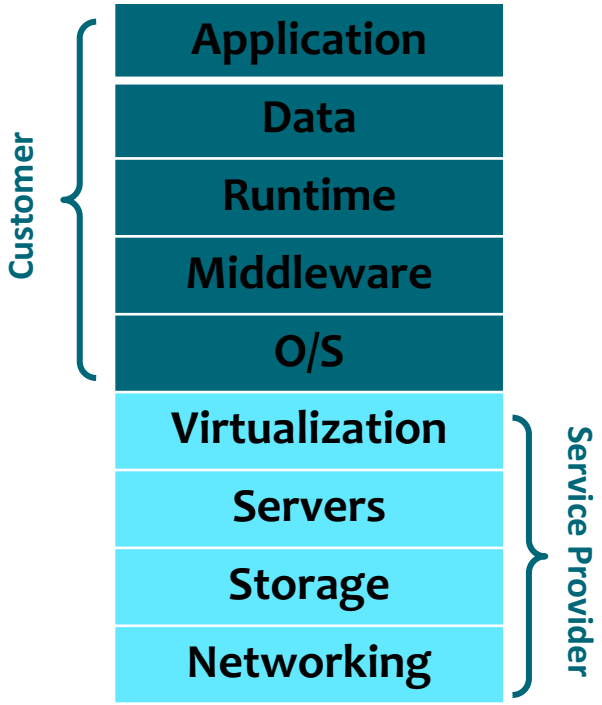
Cloud Computing Wishlist



Cloud Computing Challenges

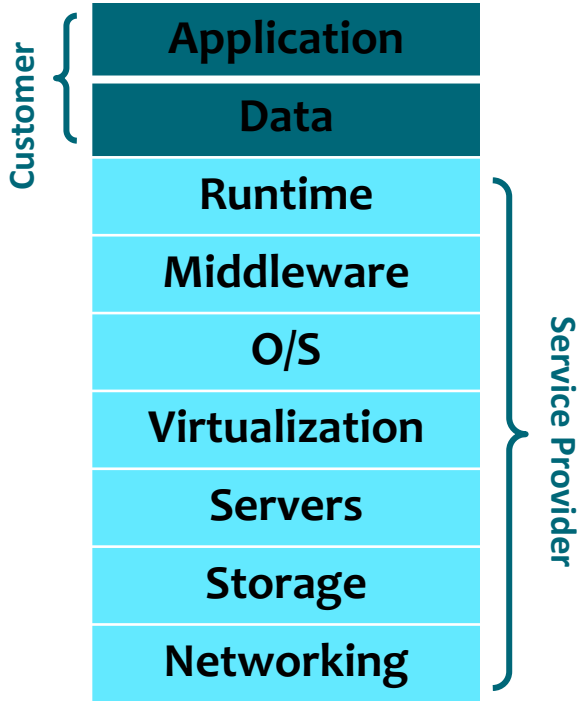


Cloud Computing Landscape



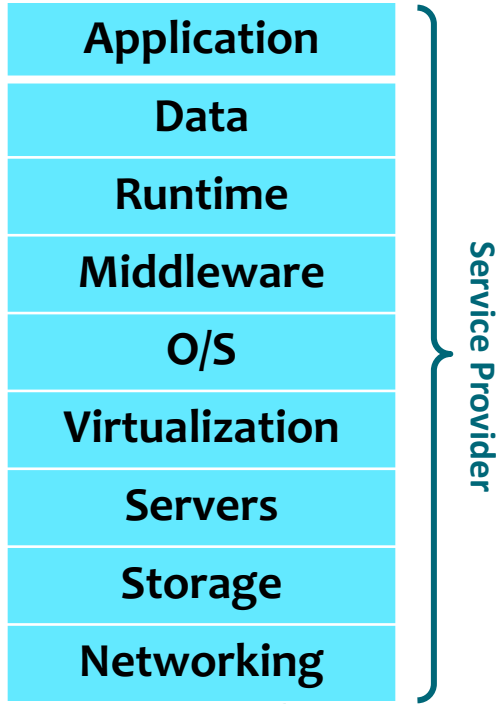
Infrastructure as a Service (IaaS)

CSC's ePouta/cPouta
Amazon EC2,
Microsoft Azure….



Platform as a Service (PaaS)

CSC's RAHTI
CSC's notebook.csc.fi
Google AppEngine,
Heroku,…



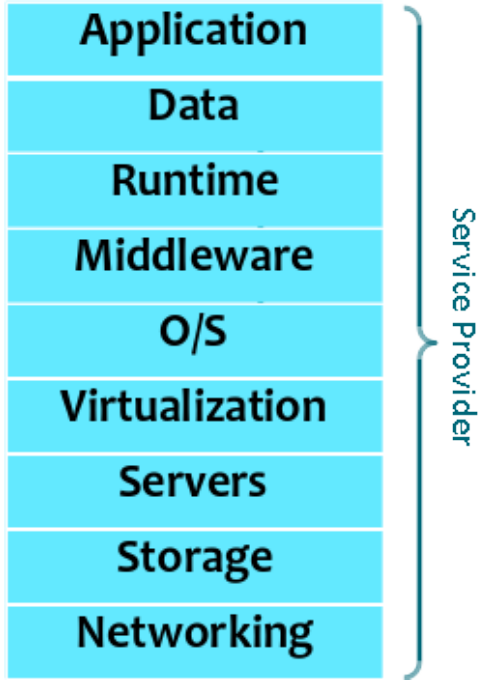
Software as a Service (SaaS)

CSC's Chipster,
Google Web Apps,
Microsoft Web Apps,,..

Cloud Service Landscape

- "Cloud" is a **very** overloaded term
  - And for good reason - almost any kind of workload can be crammed in there!
  - Most often cloud is confused to be SaaS and nothing more
- Chances are that on a daily basis you encounter tens of services which are using a cloud backend for storing both **content** and **state**
  - Anything from your employer's electronic access control system to a public transport information service to a music service you pick up the latest grooves from

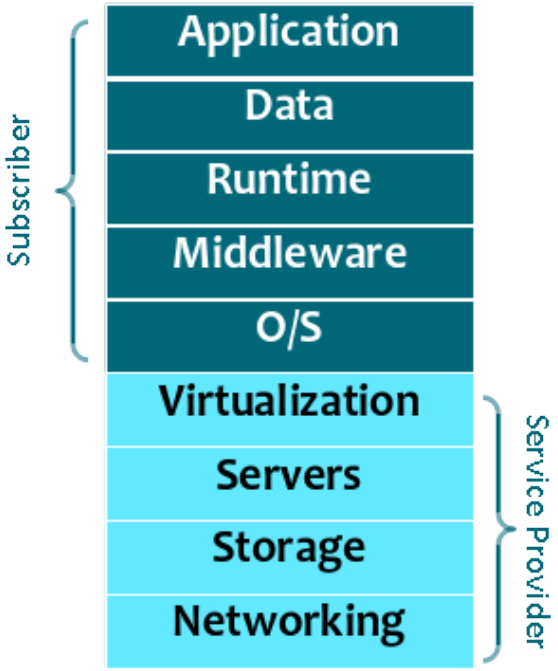
Software as a Service (SaaS)



For example:
chipster.csc.fi, Google Web Apps, Microsoft Web Apps, Dropbox, Spotify

Cloud Service Landscape

Infrastructure as a Service (IaaS)

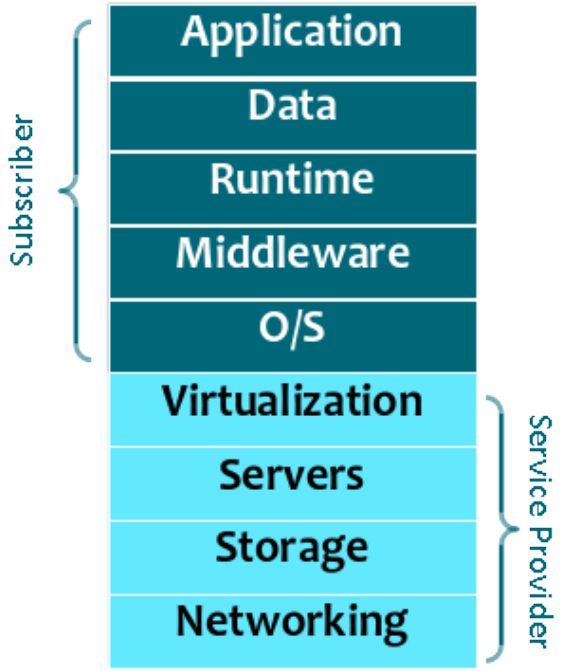


For example:
pouta.csc.fi, Amazon EC2, Microsoft Azure, Google Compute Engine

- This course is strictly about IaaS cloud
- However, as in the universe at large, also in the ICT field things are in constant flux and nothing is permanent
- Yesterday's services in a Virtual Machine are today's containerized microservices; today's containerized microservices are tomorrow's serverless architectures
- IaaS probably has some evolutionary cycles left, too
- Let's proceed to look at how IaaS is setup!

Cloud Service Landscape

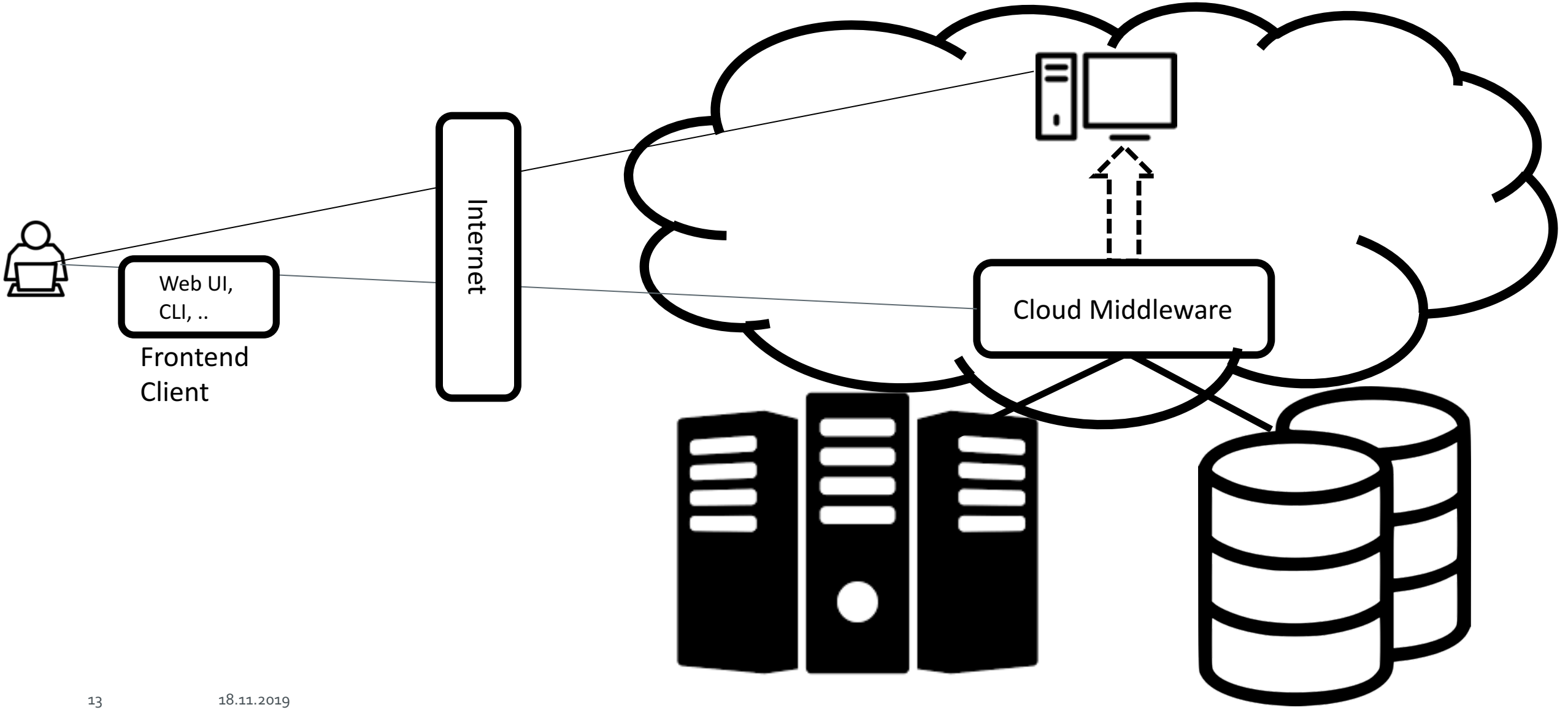
Infrastructure as a Service (IaaS)



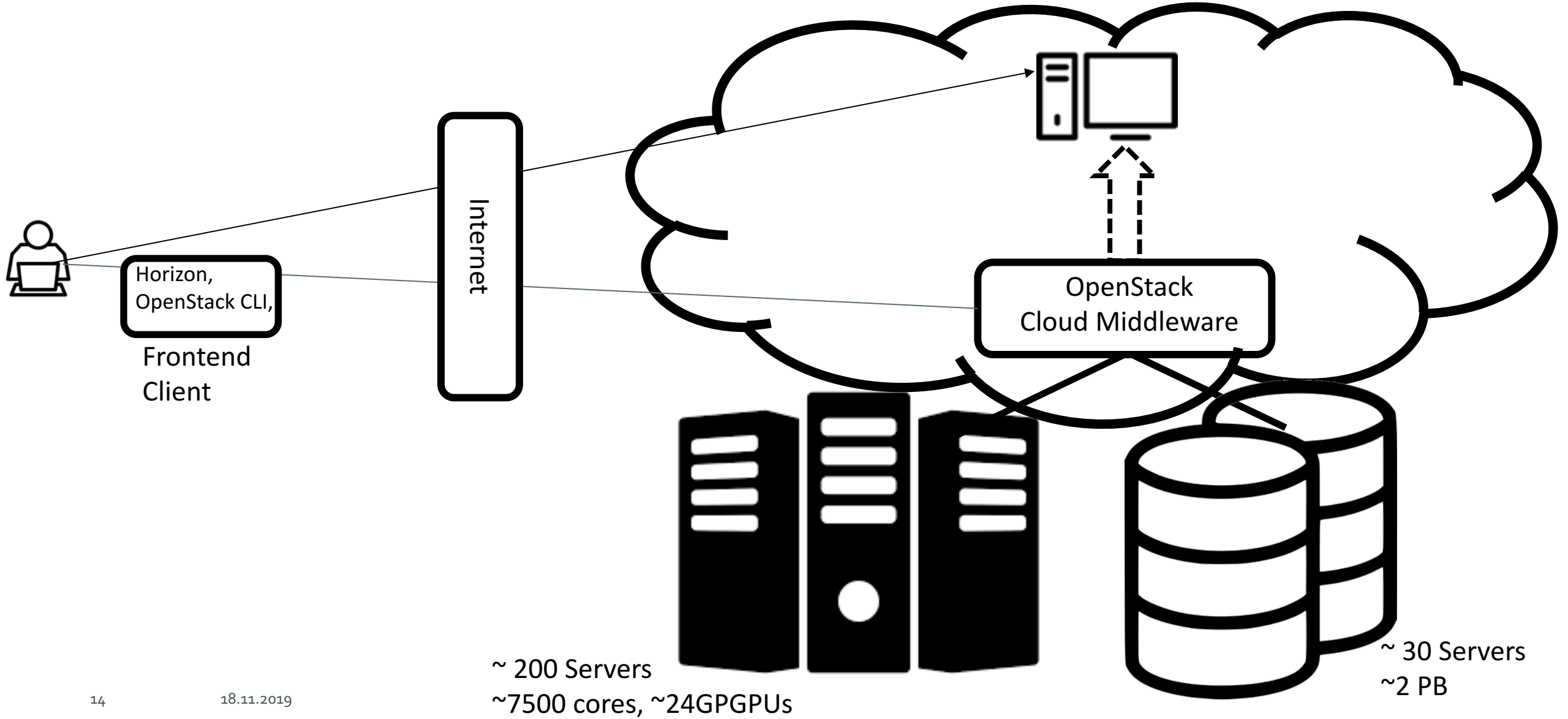
For example:
pouta.csc.fi, Amazon EC2, Microsoft Azure, Google Compute Engine

- This course is strictly about IaaS cloud
- However, as in the universe at large, also in the ICT field things are in constant fluxx and nothing is permanent
- Yesterday's services in a Virtual Machine are today's containerized microservices; today's containerized microservices are tomorrow's serverless architectures
- IaaS probably has some evolutionary cycles left, too
- Let's proceed to look at how IaaS is setup!

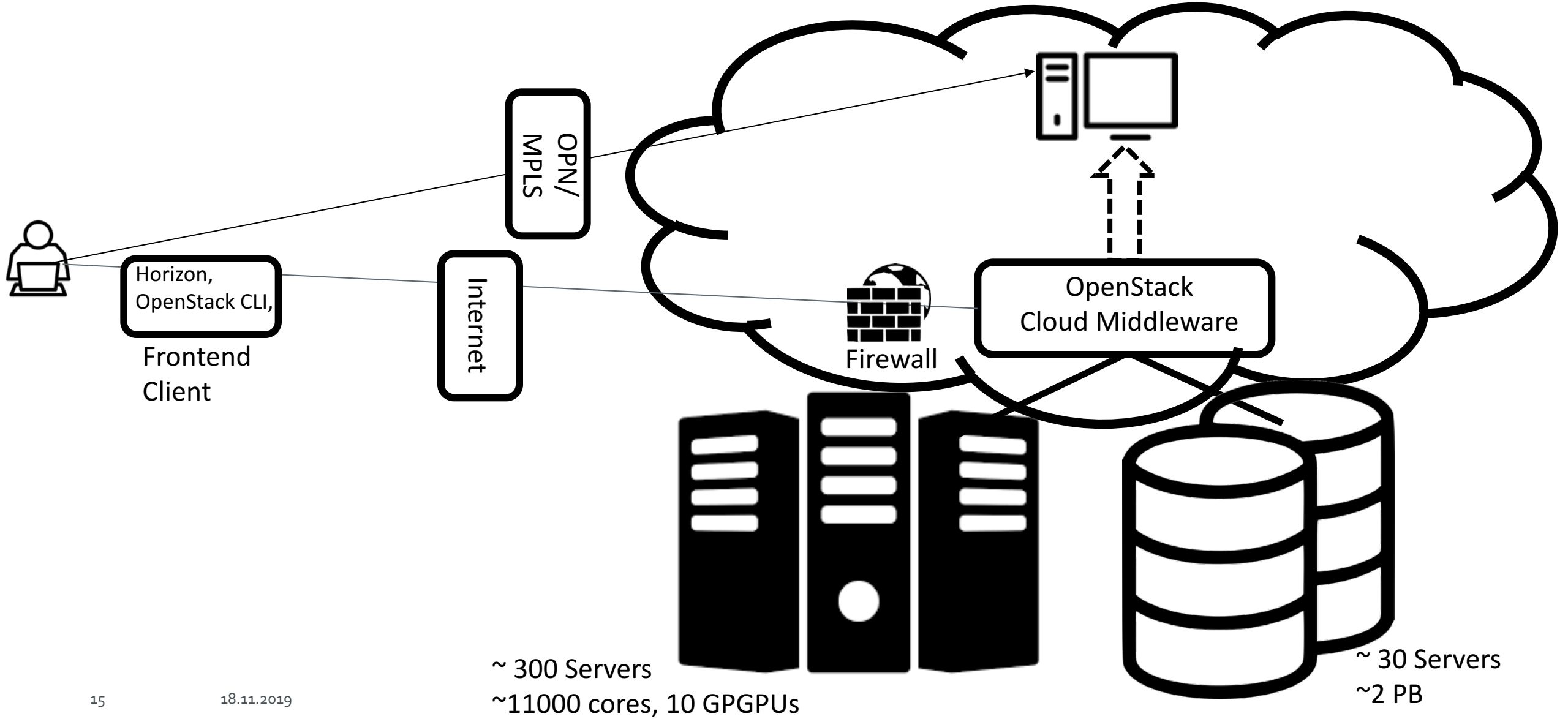
Typical IaaS Cloud Setup



cPouta Cloud Setup



ePouta Cloud Setup



Creating virtual resources in Pouta - User Interfaces

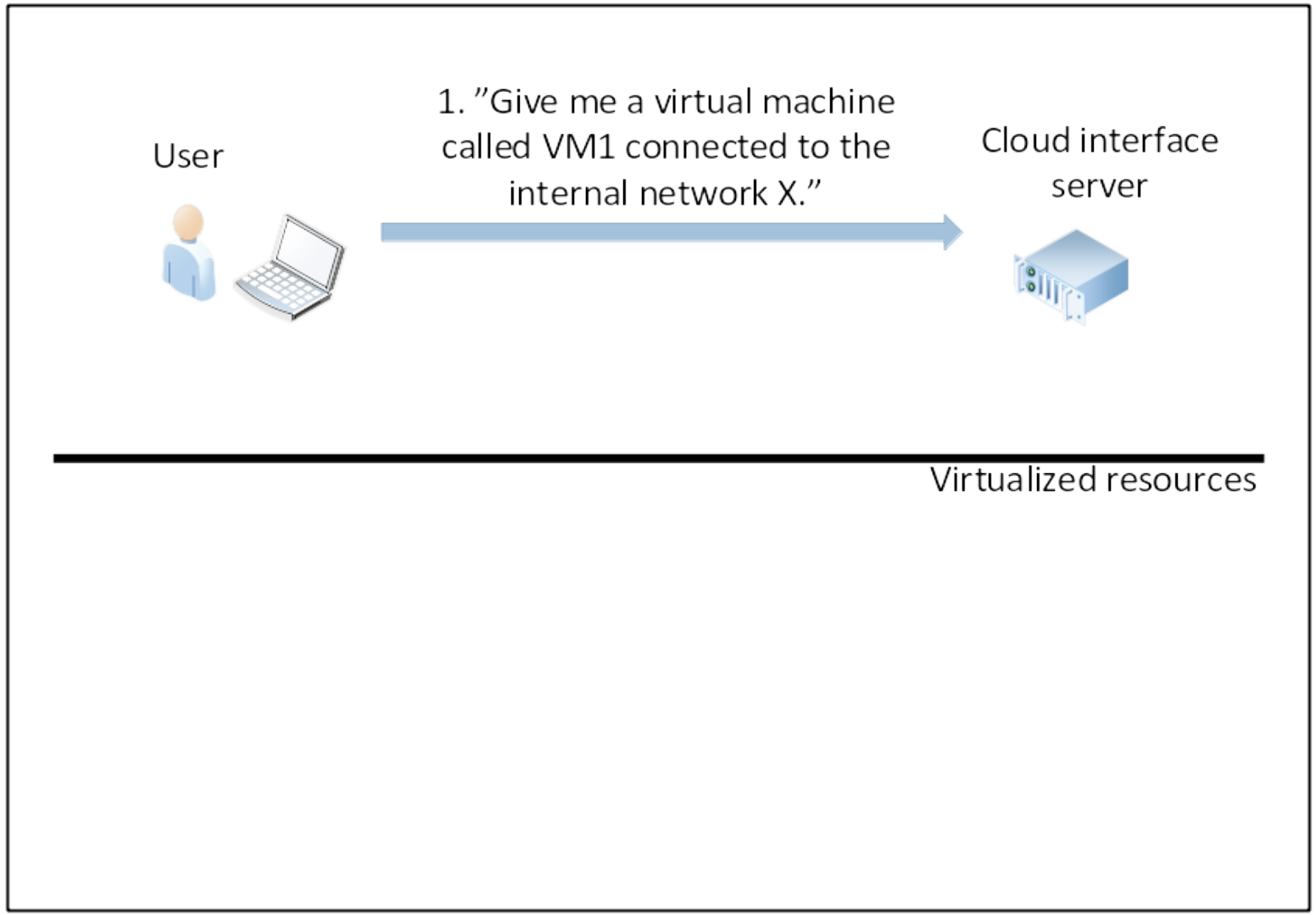
- Web User Interface -
  - Suitable for administering individual VMs, keys, images, volumes…
  - The only UI to support Haka federated login

Today's MO

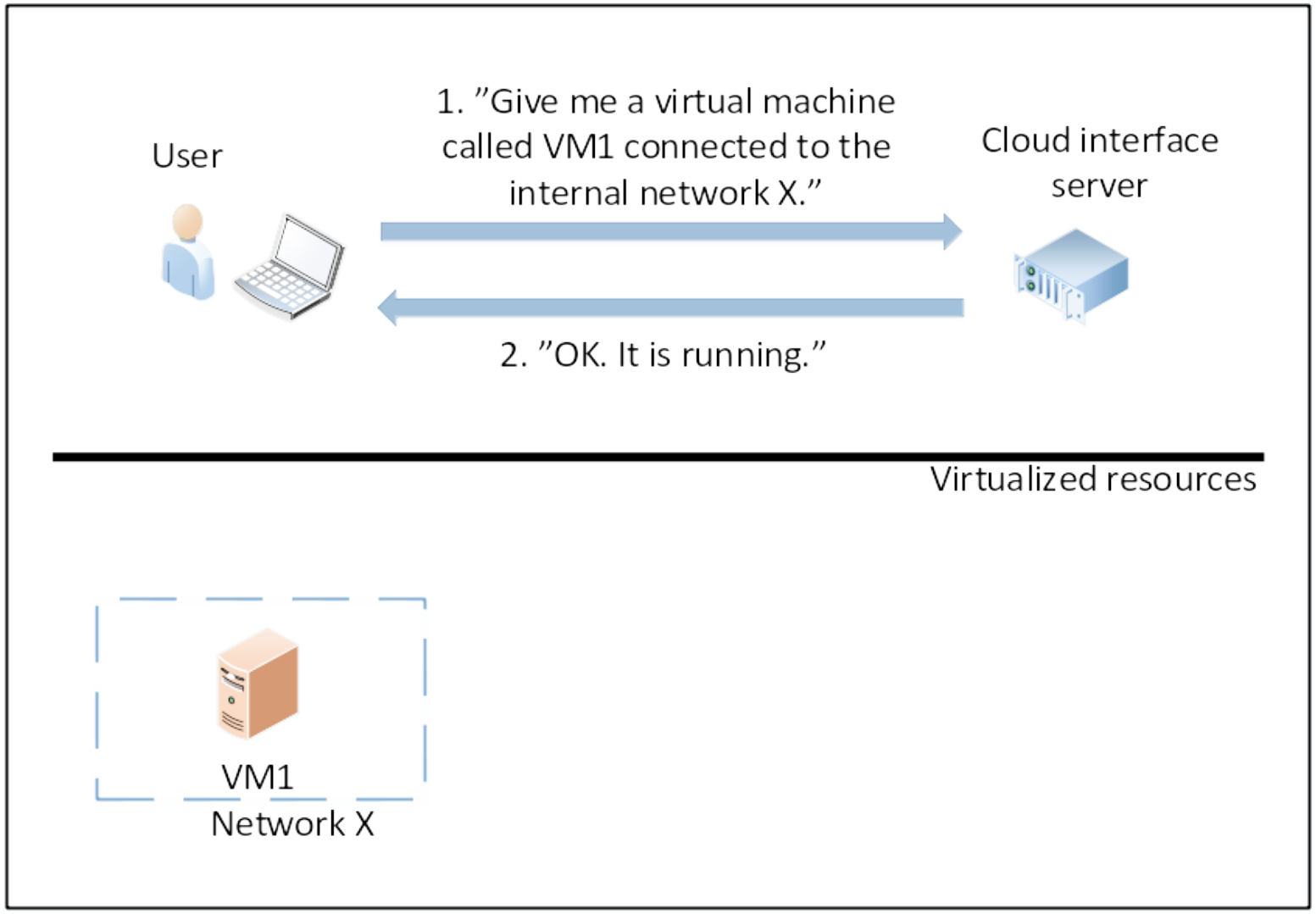
- CLI tools
  - Suitable for more elaborate resource provisioning and possibly some lightweight (scripted) software integrations
  - More info at ***https://research.csc.fi/pouta-install-client***

- Programming APIs
  - Suitable for building very large systems and stacks
  - Support from individual services (compute, storage) to full-fledged orchestration
  - List of APIs available at ***https://pouta.csc.fi/dashboard/project/access_and_security***

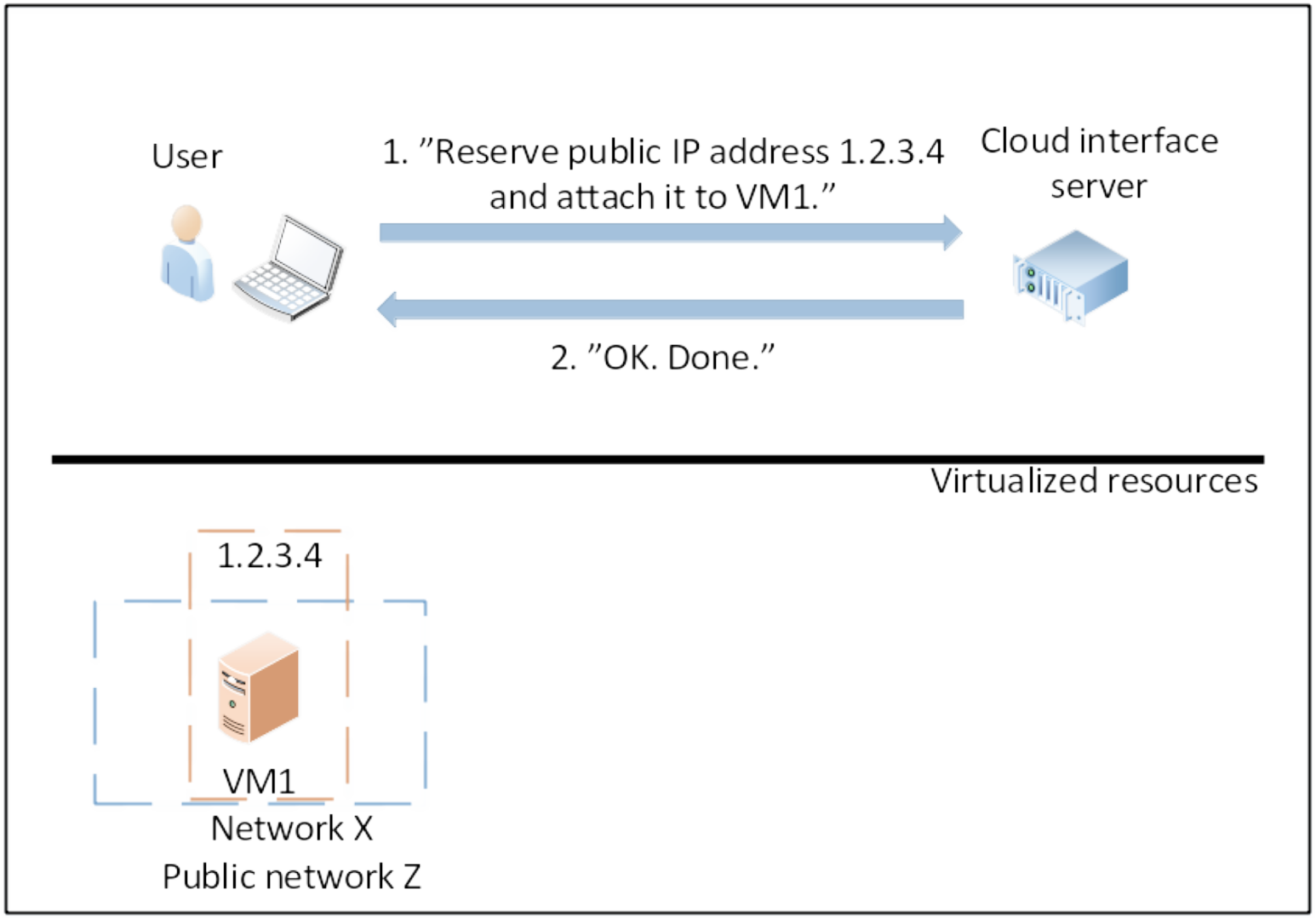
Workflow For Creating Resources



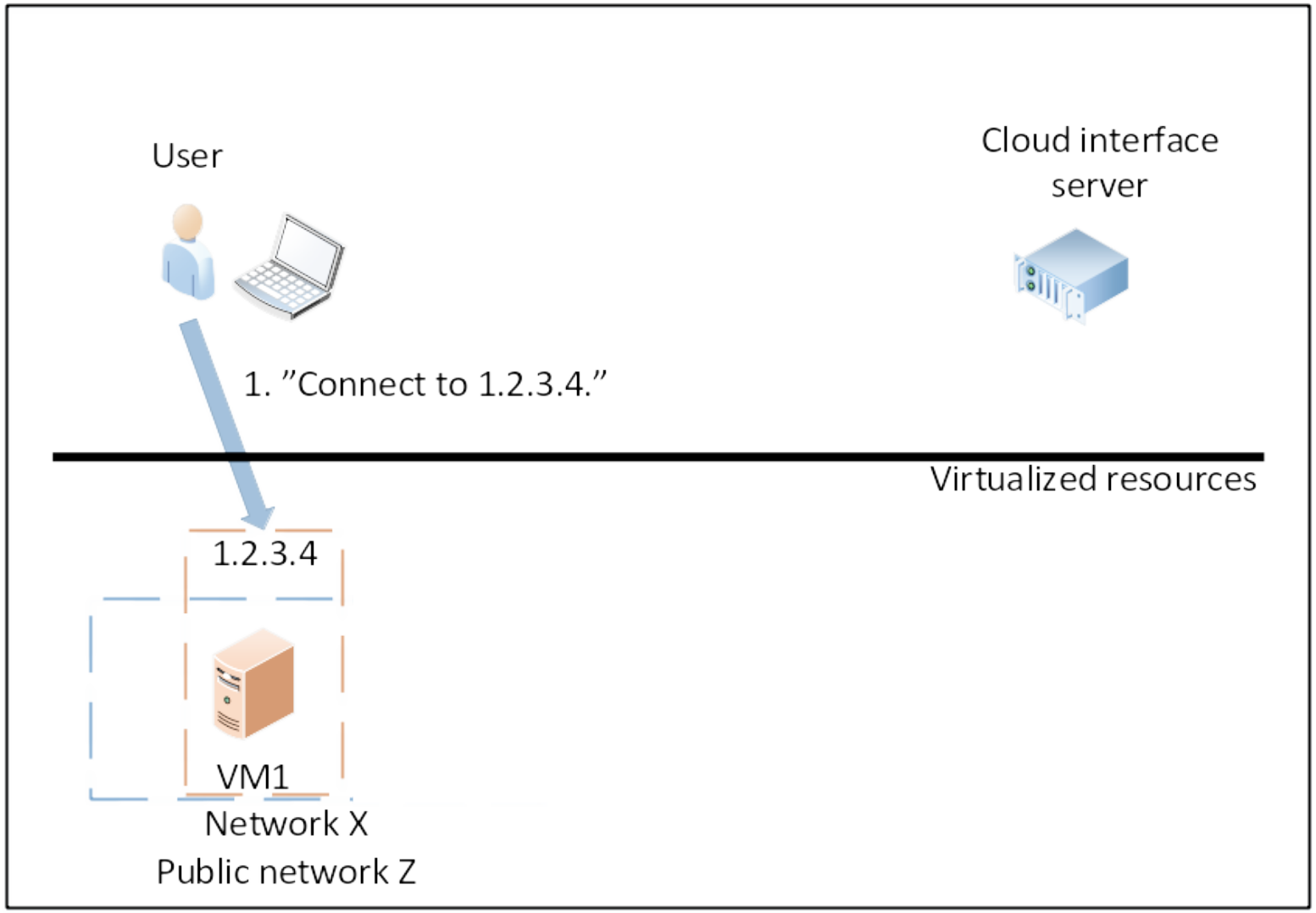
Workflow For Creating Resources



Workflow For Creating Resources



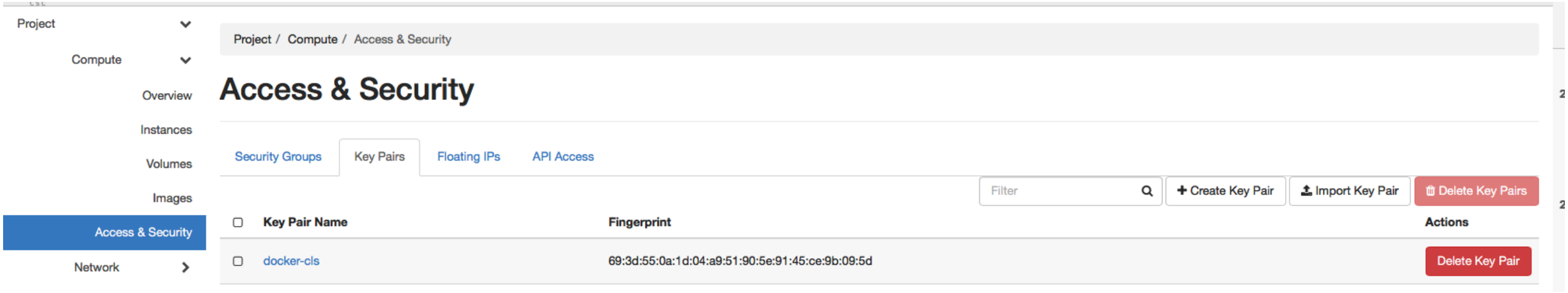
Workflow For Creating Resources



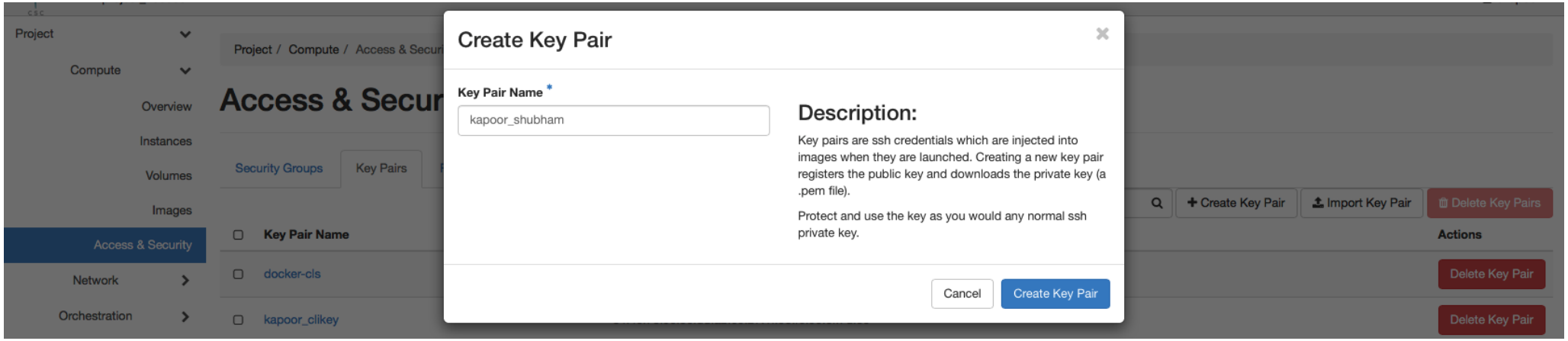
Things needed to create and access a VM in cPouta

- Access to Pouta Web UI
- One IPv4 address - a public "**Floating IP**"
- **Security Group** permitting access from User's computer
- Identity
  - Password (not recommended, but acceptable for one-time test)
  - **SSH Key**-Based Authentication (recommended)
- **SSH** client software
- Internet access

Creating a Key pair



Navigate to **Compute>Access and Security>Key Pairs**



Click on create Key Pair, name key as **lastname_firstname**

Storing a Key pair

Linux and Mac OS X

- Create .ssh directory in ~ if its not there already

```
mkdir -p  .ssh
chmod 700 .ssh
```

- Move key pair to .ssh directory

```
cd .ssh
mv ../Downloads/yourkey.pem .
```

- Protect key with passphrase (Optional)

```
ssh-keygen -p -f yourkey.pem
```

- Make key unreadable by other users

```
chmod 400 yourkey.pem
```

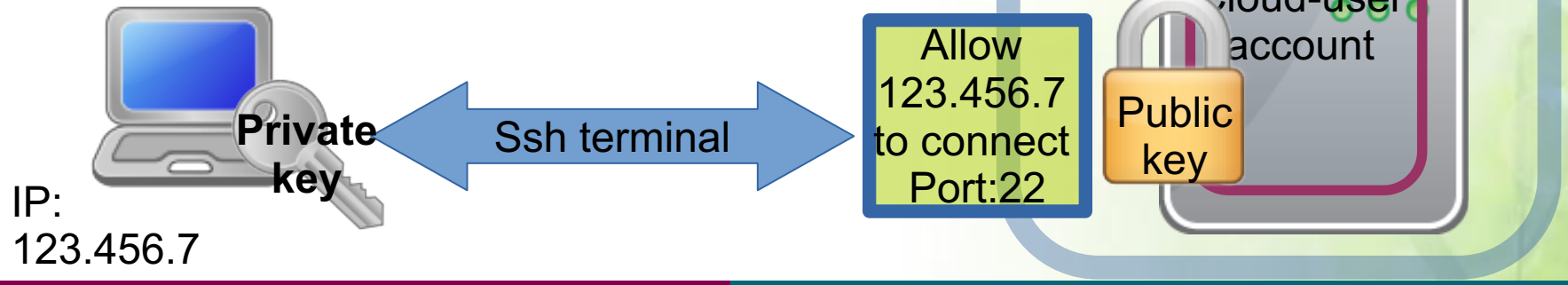
Windows

- Download Putty and Puttygen tools if you don't have them

- Load your **private key (yourkey.pem)** into **puttygen** and change it to .ppk format
- Open Putty, load .ppk file under **Connection | SSH | Auth | Private key file for authentication**
  - Provide user name **cloud-user**
  - Provide password which you added to Puttygen (Optional)

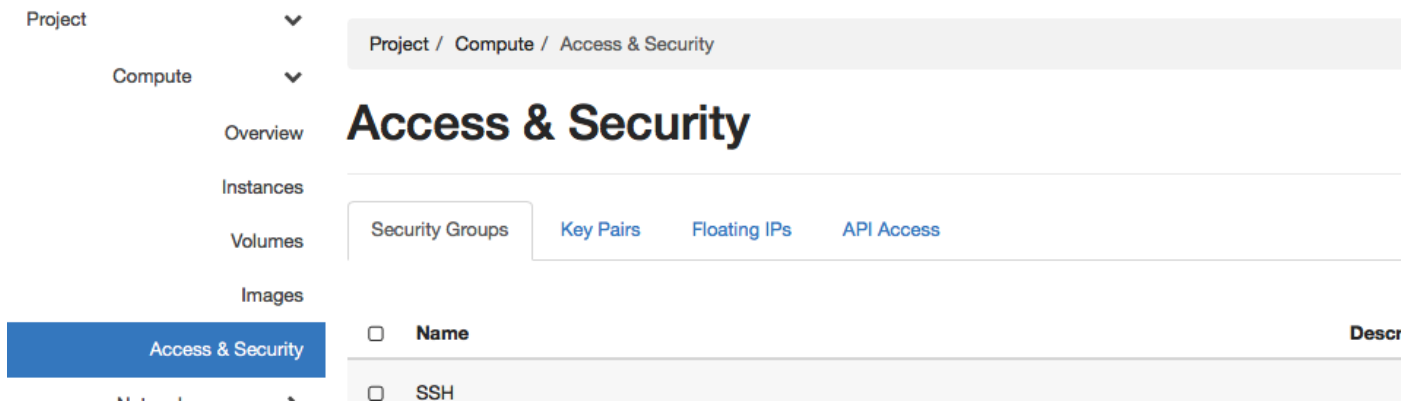
Opening VM to internet with security groups

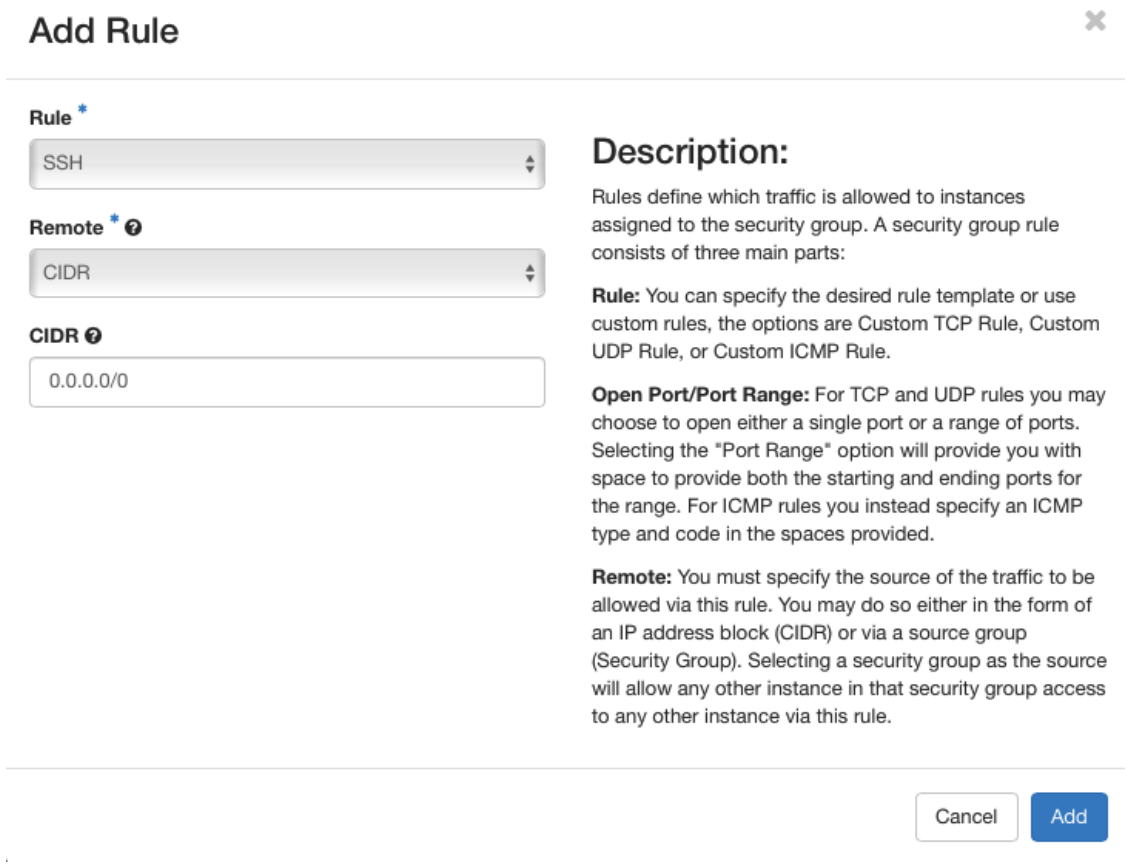
- A Security Group defines a set of cloud level firewall rules for filtering traffic, typically inbound
- By default Security Groups blocks all incoming connections to your VM
- Security groups define combinations of ports and IP addresses for which the incoming connections are permitted
- Security groups can be created in the web interface and then applied to virtual machines
- One security group and include several “rules” and be used by several virtual machine
- One machine can use several security groups and vice versa
- You can additionally have some VM level firewall rules in conjunction to security group for better security.
- Typical case: allow connections from the IP address of your own computer to port 22 (SSH port).



Security Groups

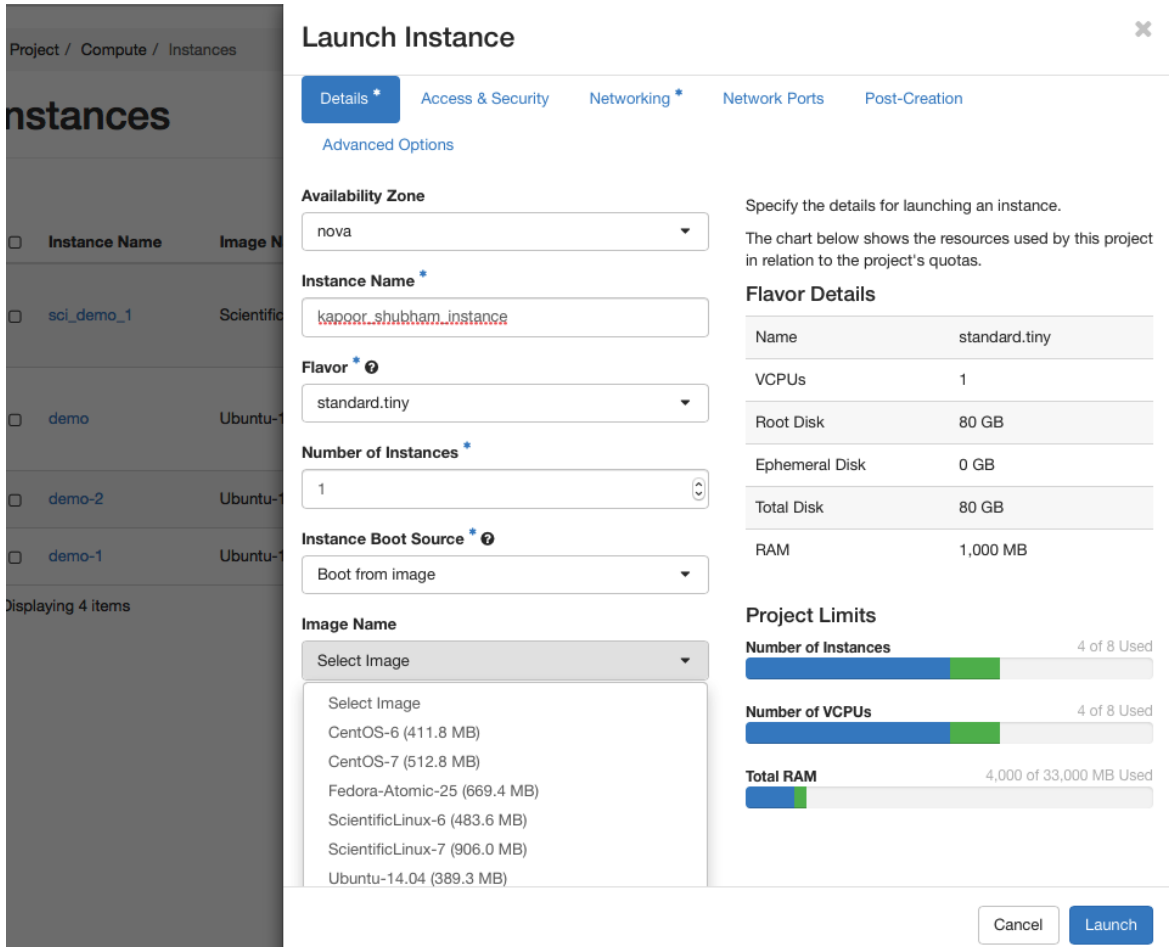
- Created by navigating to **Compute>Access and Security>Create Security Group**
  - Several predefined rule sets are available, such as for SSH
  - At bare minimum you need to select the Source IP for the traffic
  - Modify the **CIDR** field to allow SSH connections only from specific IPs





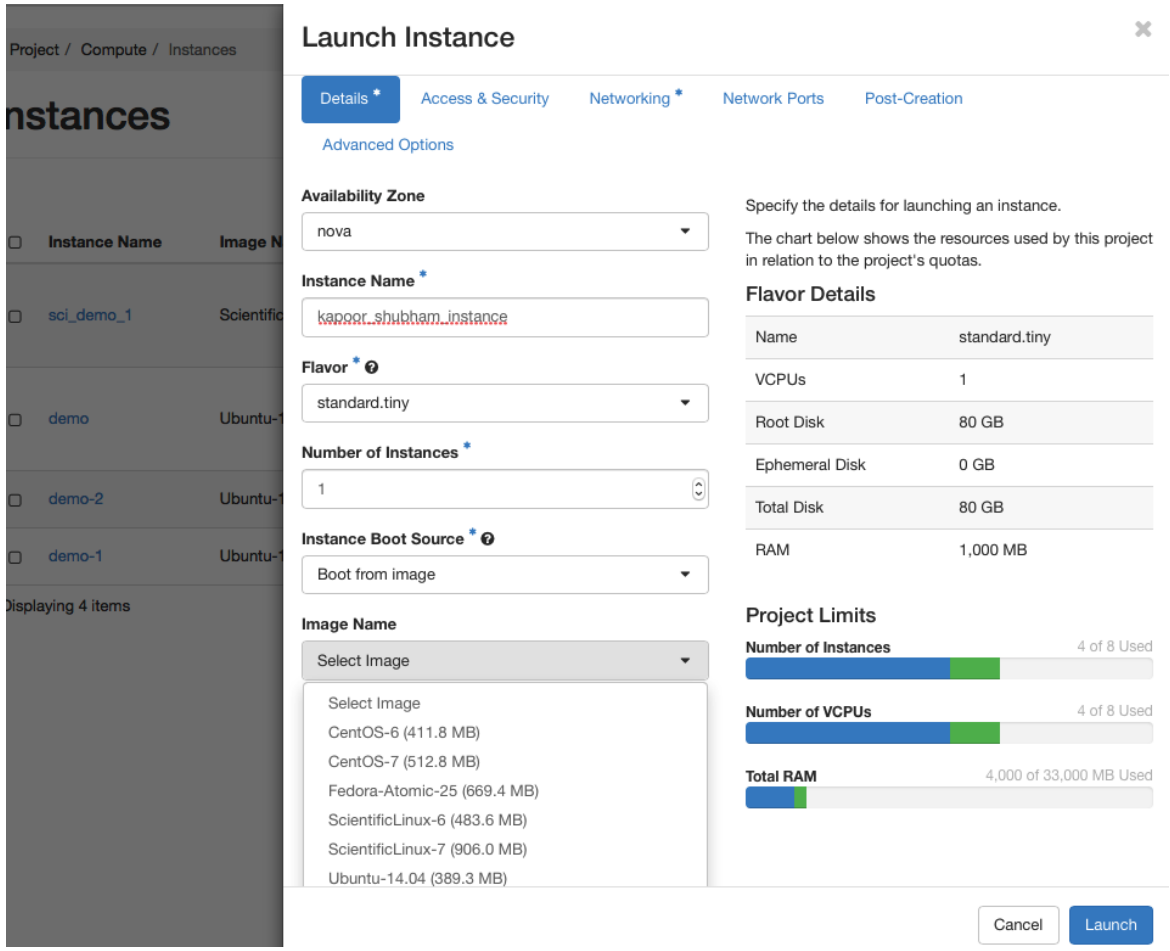
Creating an Instance

- Navigate to **Compute>Instances** and **Launch Instance**
- Give Instance name as **lastname_firstname_instance**
- Select a Flavor of your choice (**standard.tiny** is a good first choice)
- Select Instance Boot Source as **Boot from image**
- Pick an image - any image
- Navigate to **Access & Security** in same popup. Make sure that the "**SSH - World"** Security Group is selected.
- Populate the Post-Creation script as per



Creating an Instance - Redux!

- Add SSH key pair to Web UI
- Create instance as before
- In **Access & Security**, make sure that the SSH key pair is selected
- When connecting to the instance, designate the private key into the session or pre-populate it into an SSH agent prior to making a connection



Coffee break

Web UI Login

- From your web browser, browse ***https://pouta.csc.fi***
- Log in using provided training "*trainingXX*" username and password
  - All set? Great!
  - Issues logging in? Please let Shubham or Juhani know and we'll help
- Everybody should be able to log in to the cPouta Web UI before we start exercises

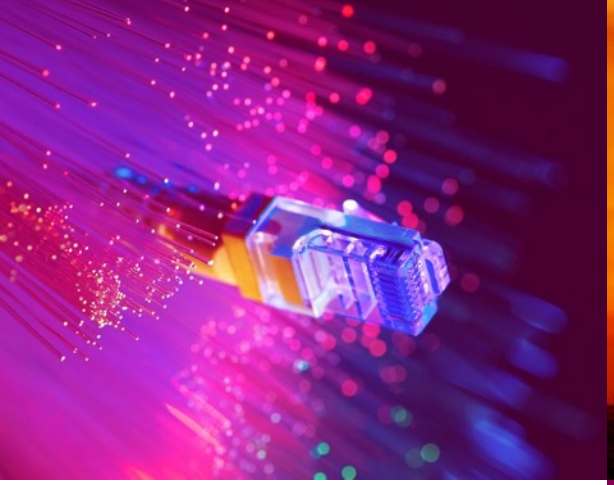
Exercise Set 1: Creating & Securing Virtual Resources

- **Exercise 1 - Creating a temporary Virtual Machine for testing login**
  - Log in to Cloud Dashboard at https://pouta.csc.fi/
  - Create your own Virtual Machine Instance with disposable password in post creation section
  - Associate Floating IP to Virtual Machine Instance
  - Log in to your VM using SSH or Putty
  - Exit and delete the VM
- **Exercise 2 - Creating an SSH key pair for secure login to an Instance**
  - Create an SSH key pair, storing the private key in a safe place
  - Create new VM Instance using this key pair
  - Associate Floating IP address to VM
  - Log in
  - Declare victory! Watch Star Wars!!

Exercise Set 1: Creating & Securing Virtual Resources

- **Exercise 3 – Create your own Security Group for securing your virtual resources**
  - Create your own Security Group for SSH traffic
  - Start by creating a wrong Security Group rule
  - Attach it to your VM
  - You would not be able to access your VM
  - Modify Security Group again, this time with correct Security Group rule
  - Connect to your VM

Lunch



Pouta: CSC's IaaS Cloud Services

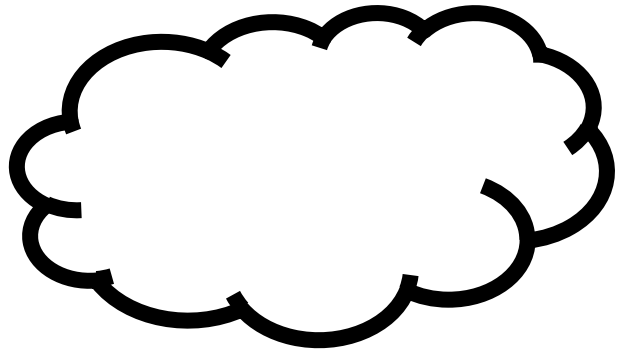
CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Pouta: CSC's IaaS offering

- CSC provides two Infrastructure as a Service (IaaS) cloud services for research and education:
  - **cPouta**
  - **ePouta**
- Powered by OpenStack.
  - Current version Newton.
- Management processes ISO27001 and VAHTI 2010 certified.
- True IaaS experience
  - Deploy your own Virtual Machine
  - Decide your own OS, Middleware, Runtime environment & Storage.
  - Design your own Network based upon your project needs.
- Simple to create and modify VMs: Web UI, CLI and REST API interfaces supported.

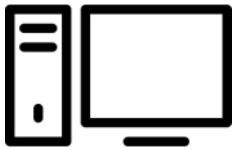
cPouta

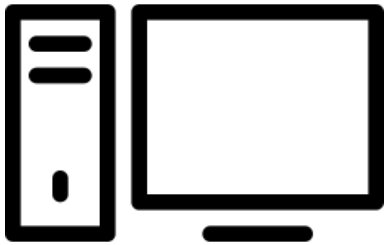
- OpenStack based cloud.
- Serving cloud computing needs of Finnish research institutes & universities since 2013.
- VMs and Control plane can be accessed via public internet.
- Customers may decide access to VMs by creating firewall rules at OpenStack level known as “Security Groups”.
- ISO27001 certified .
- Can support modern DevOps, Agile, CI/CD etc. environments
- Could be used for hosting:
  - Scientific applications,
  - Custom services such as Web servers, File servers, load balancer etc.,
  - Virtual Computer class,
  - Research Data Sharing etc.



cPouta VM Flavors

Flavor	Cores	Memory	Disk (root)	Disk (ephemeral)	Disk (total)	Memory/core	Billing Units/h
standard.tiny	1	1000 MB	80 GB	0 GB	80 GB	1000 MB	0.5
standard.small	2	2000 MB	80 GB	0 GB	80 GB	1000 MB	1
standard.medium	3	4000 MB	80 GB	0 GB	80 GB	1333 MB	2
hpc-gen1.1core	1	3750 MB	80 GB	0 GB	80 GB	3750 MB	2
hpc-gen2.2core	2	10000 MB	80 GB	0 GB	80 GB	5000 MB	4
io.70GB	2	10000 MB	20 GB	70 GB	90 GB	5000 MB	5
hpc-gen1.4core	4	15000 MB	80 GB	0 GB	80 GB	3750 MB	8
hpc-gen2.48core	48	240	80 GB (RAID0)	0 GB	80 GB	5000 MB	90





Full list of flavor available at https://research.csc.fi/pouta-flavours

ePouta

- OpenStack based cloud.

- Serving cloud computing needs of Finnish research institutes & universities which involves **Sensitive Data**.

- Complete Isolation of VMs from rest of the world and other ePouta customers.

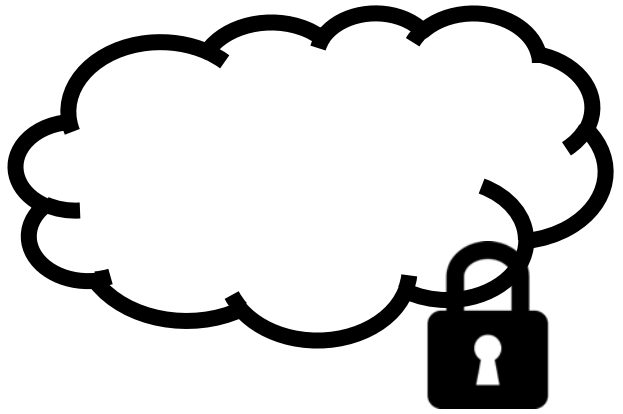
- VMs accessible only from customer network.

- VPC : Optical Private Network(OPN) or MPLS VPN connection between the end customer and ePouta VM instances.

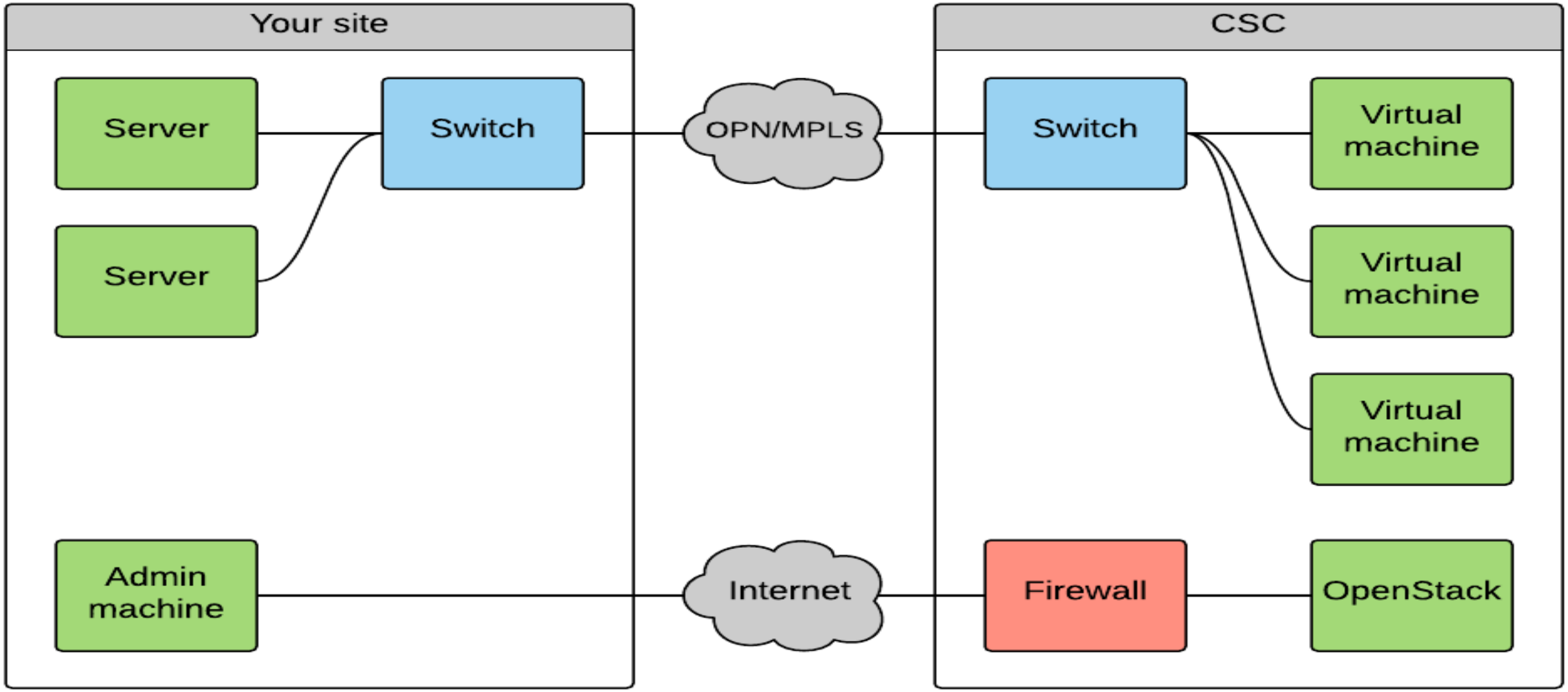
- ISO 27001 certified.

- Can support modern DevOps, Agile, CI/CD etc. environments.

- Could be used for hosting:
  - Scientific applications dealing with sensitive data,
  - Sensitive Data Sharing, Archiving etc.



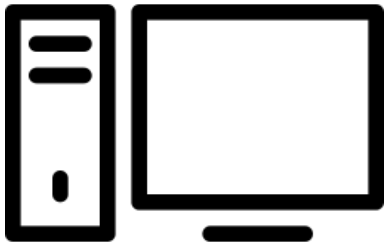
ePouta : Connection



Typical VM connections between ePouta and customer's network. Such connections are normally coordinated between CSC's cloud team, Funet and customers IT department for initial setup.

ePouta VM Flavors

Flavor	Cores	Memory (MB)	Disk (root)	Disk (ephemeral)	Billing Units/h
hpc.mini	2	3600	80	0	5
hpc.medium.westmere	8	14400	80	0	8
hpc.small	4	7200	80	0	10
io.haswell.8core	8	40000	20 (SSD/RAID0)	350 (SSD/RAID0)	25
io.haswell.16core	16	80000	20 (SSD/RAID0)	700 (SSD/RAID0)	50
hpc.xlarge.haswell	32	160000	80	0	80
hpc.fullnode.haswell	46	248000	80	0	120
tb.3.1470RAM	80	1470000	80 GB (SSD/RAID0)	2500 GB (NVMe/RAID0)	600



Full list of flavor available at https://research.csc.fi/pouta-flavours

Pouta Usecases

- Scientific Modelling in the areas of natural sciences
- Advanced Computing and Software Development
- Big Data Analytics
- Digital Education
- Research Data Sharing & Archiving

Typical Resources You get from Pouta Clouds

- VMs
- Oversubscribed or dedicated CPUs
- GPUs

Compute

- Volume Storage
- Object Storage*

Storage

- 10 GbE or 40 GbE

Private VLAN

- With or Without * NAT

IPv4

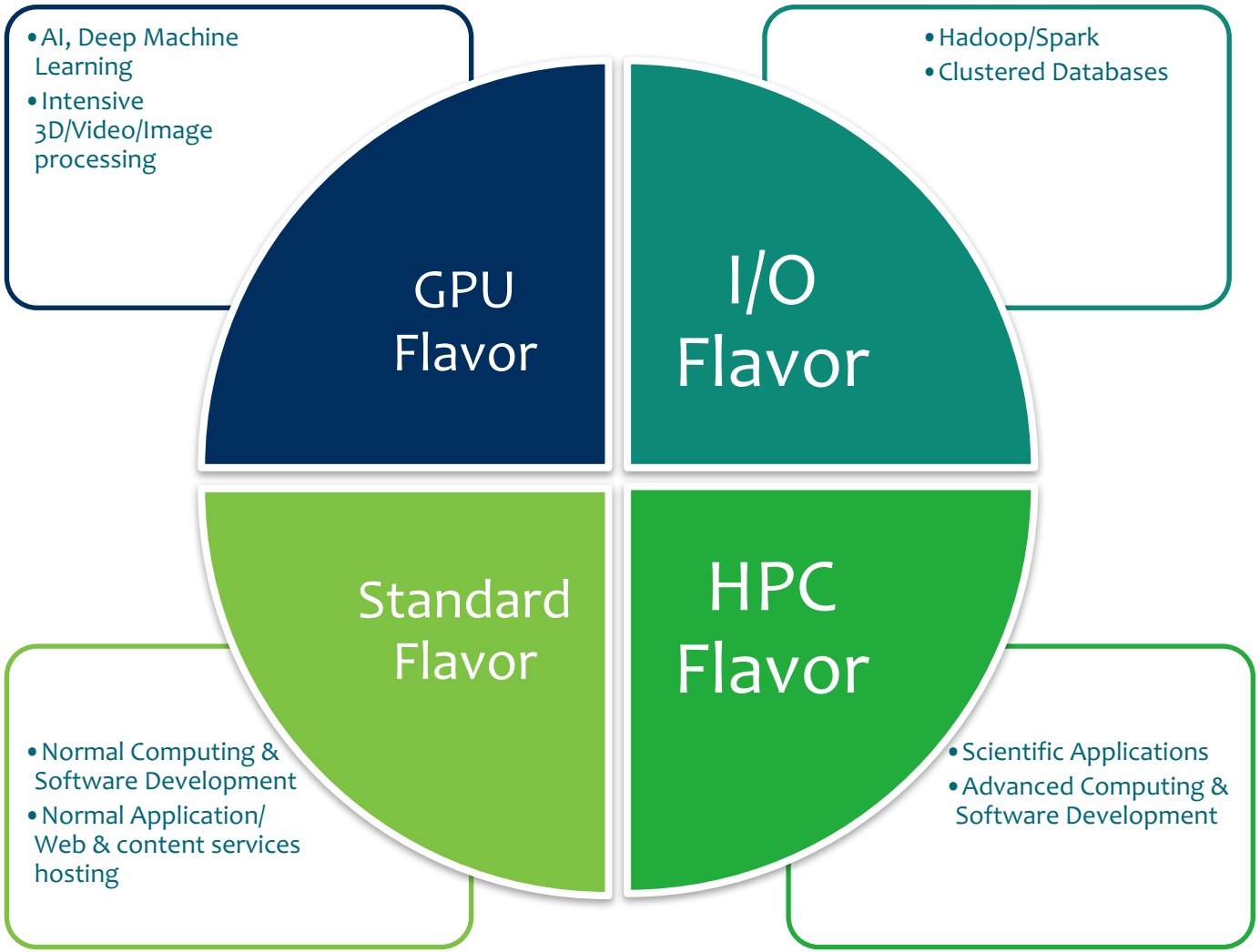
- With Latest Security patches

Images

- Full programmability of your resources

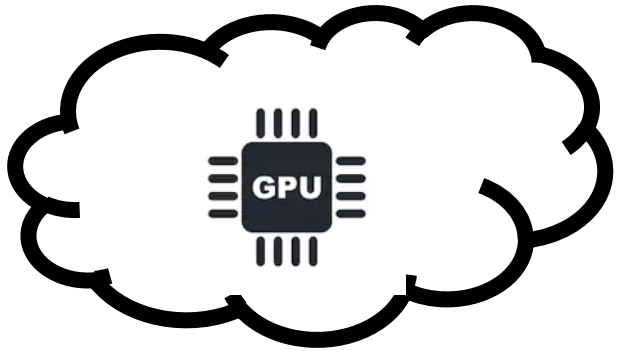
API

Pouta: Hardware Options



Diverse set of hardware options to support your computing needs

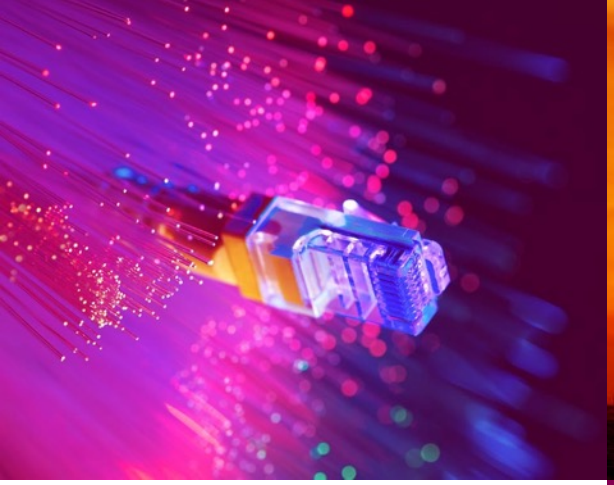
GPU Flavors in Pouta



- GPU Flavors provide high performance computing leveraging GPGPUs.
- GPU 1.* family flavors in cPouta are powered by NVIDIA Tesla P100 GPGPUs
- GPU 2.* family flavors in ePouta are powered by NVIDIA Tesla V100 GPGPUs
  - CPU: Intel(R) Xeon(R) Gold 6148, with hyper-threading
- GPU flavors are backed by local SSDs (RAID-0) on the server.
  - Advisable to use Volumes for storing important data.
- You can use CSC's Code Optimization Service in case you are coding your own application.
  - https://research.csc.fi/optimization-service
  - PCI passthrough is used to get GPGPUs in GPU flavor machines
  - OS images pre installed with latest CUDA version are available.
  - You may also use your own OS images by installing required libraries yourself.
- GPGPUs are also available in the batch system onPuhti: https://docs.csc.fi/#computing/system/

cPouta/ePouta

	cPouta public Cloud	ePouta private Cloud
ISO27001, VAHTI Raised Level	Certified	Certified
Usage	General purpose Computing	Computing on Sensitive Data
VM access	Internet	OPN/MPLS
Firewall, LB, VM installation, VM auto-recovery, Backups	Self-service	Self-service
Supported Operating Systems	No particular limits, but commercial OSs require a license	No particular limits, but commercial OSs require a license
OpenStack Version	Newton	Newton
GPU family	NVIDIA Tesla P100 GPGPUs	NVIDIA Tesla V100 GPGPUs
Object Storage	Yes (in build)	No (Could be indirectly used from your organization network)
Service availability target	99%	99%



OpenStack

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

OpenStack

- CSC's cPouta/ePouta cloud services are powered by OpenStack.
    - Current OpenStack version used by Pouta services is Newton

- OpenStack is a cloud software that allows end user to create and use their VM instances, networks and storage.

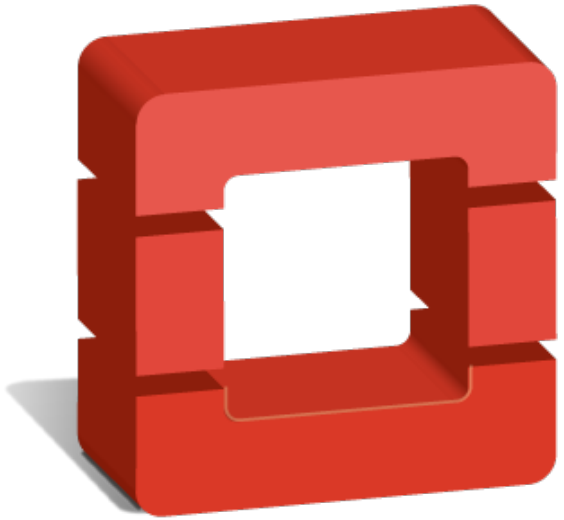
- Fast moving open source project with backing from industrial giants like AT&T, Red Hat, IBM, Intel, HP etc.

- Flexible architecture which may support different types of scalabilities.

- Used by many organizations from research institutes to service/content providers.

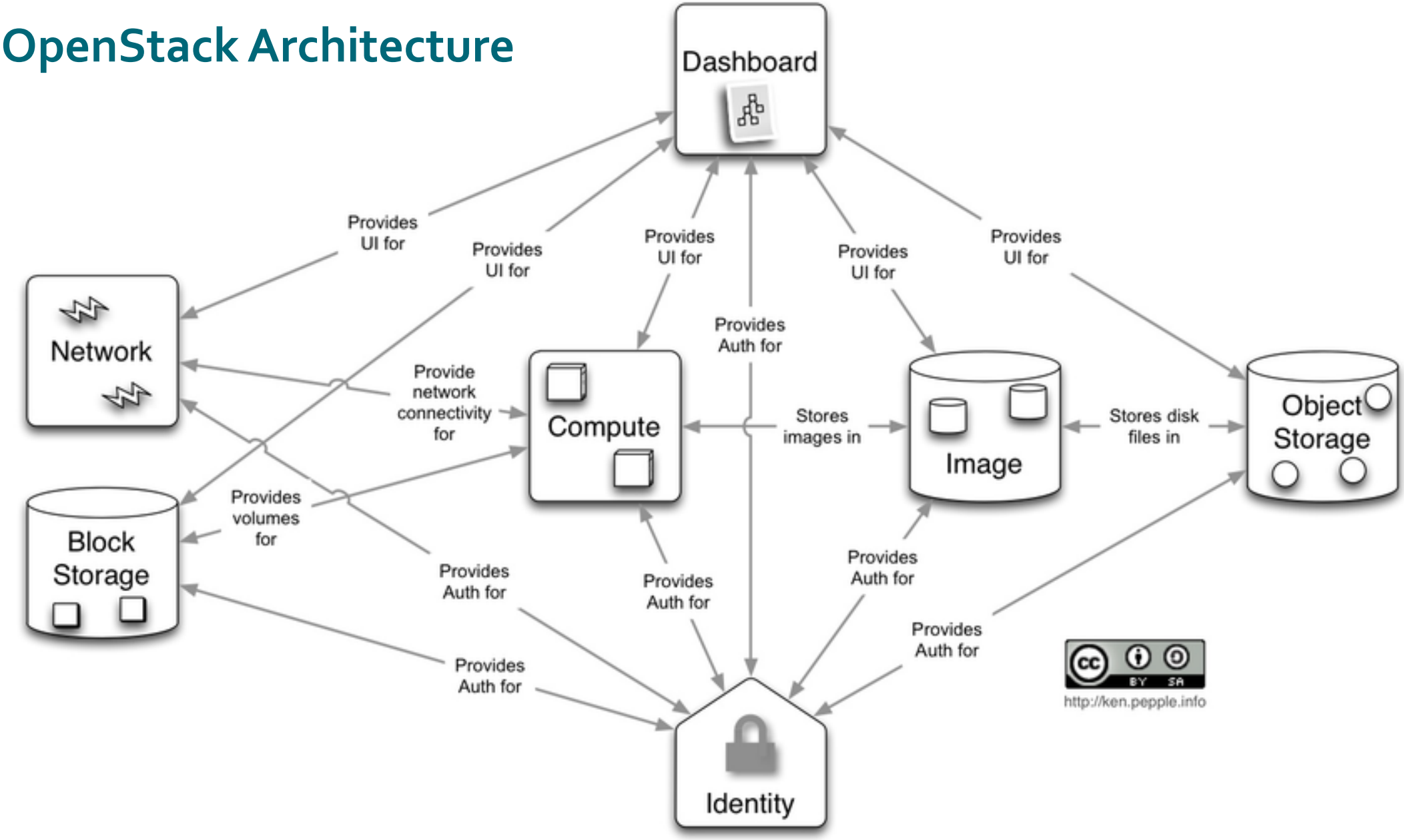
- Large customer base augments better availability of expertise, support and chances of continuity.

- Supports Web UI, CLI and REST Interfaces

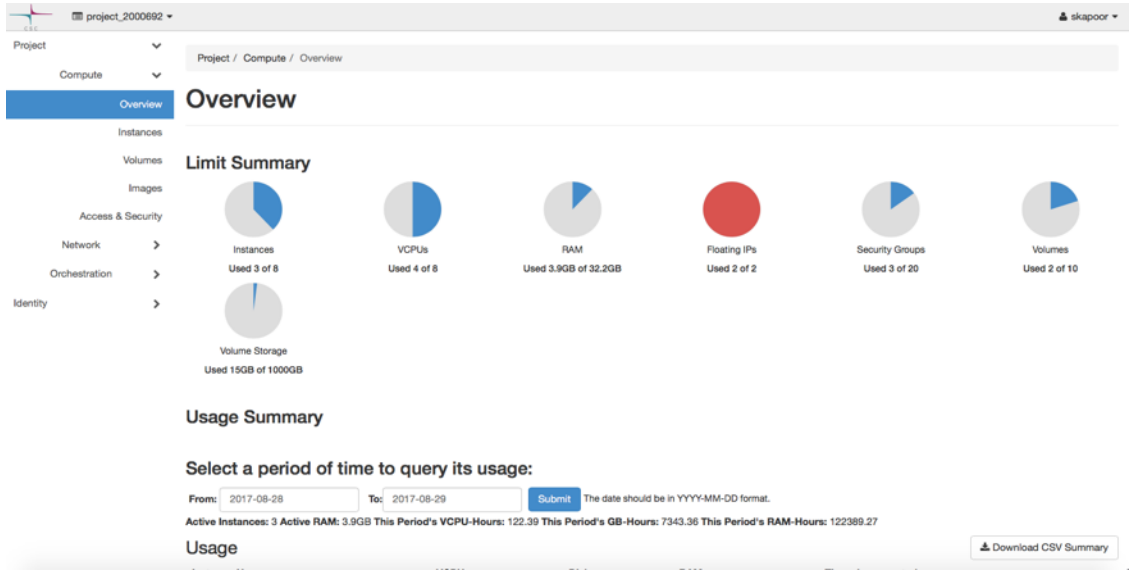


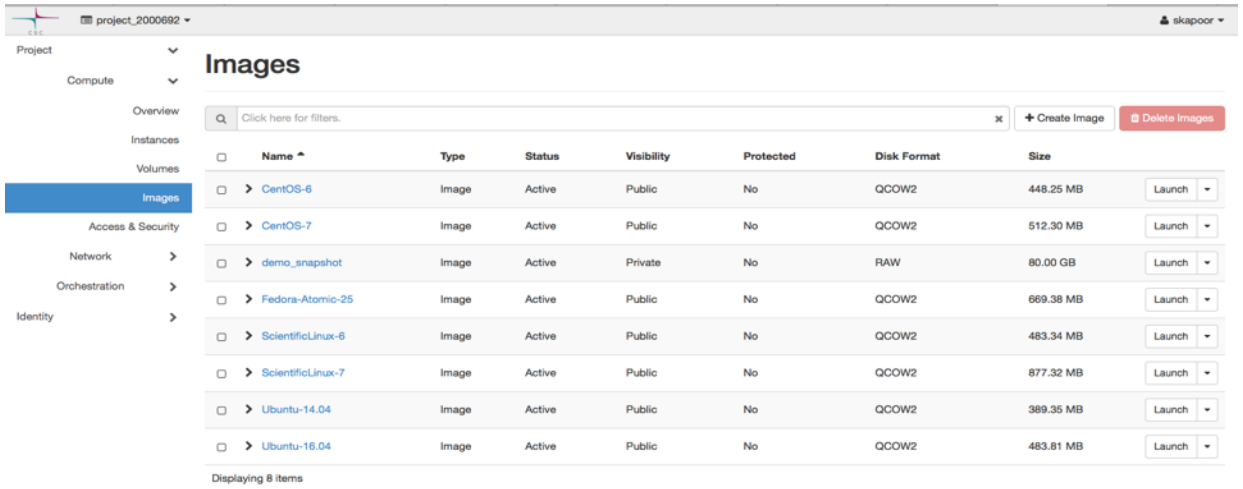
openstack™

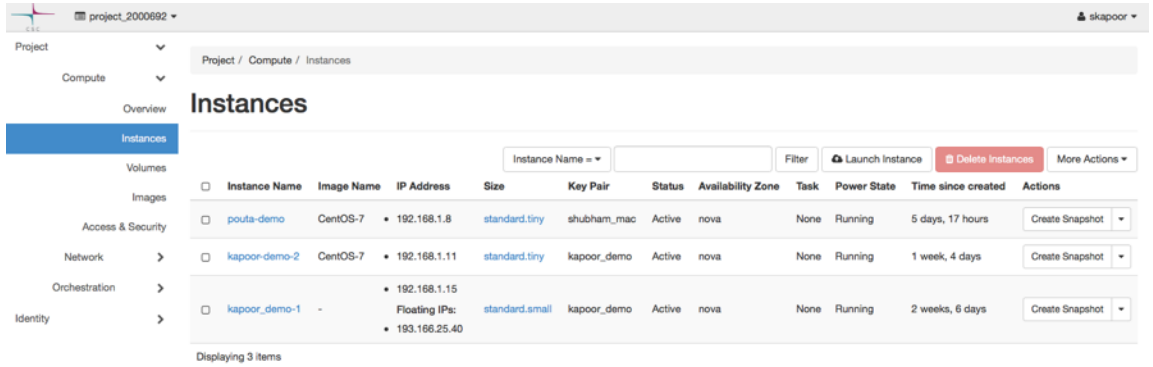
OpenStack Architecture

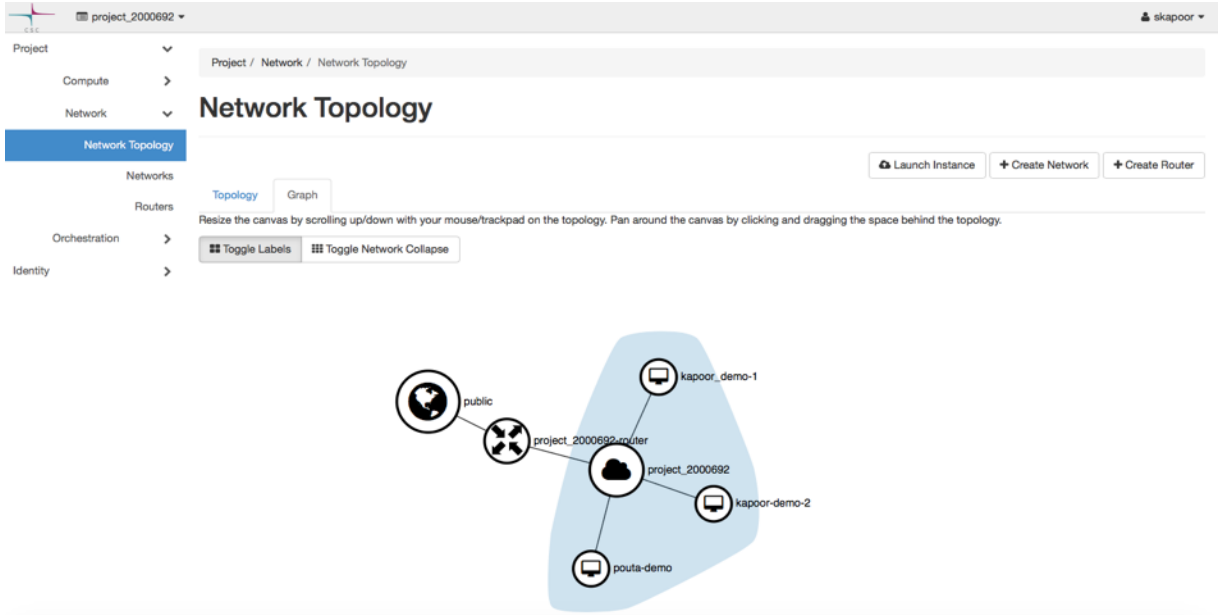


OpenStack WebUI









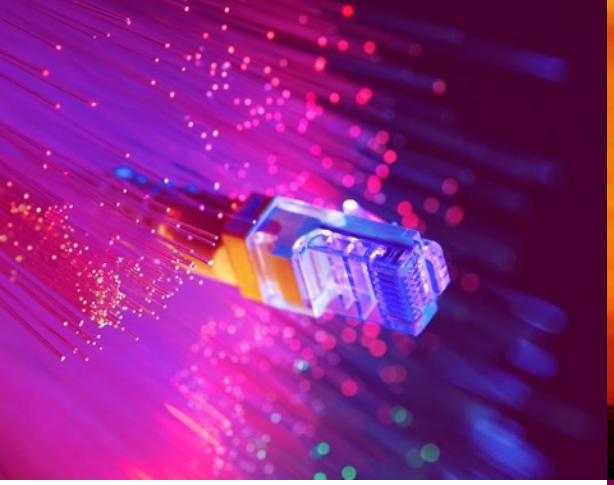
OpenStack CLI

```
(osclient) skapoor-air13:python_virtualenvs skapoor$ openstack image list
+--------------------------------------+--------------------+--------+
| ID                                   | Name               | Status |
+--------------------------------------+--------------------+--------+
| 4a36f474-4ffe-4f88-bc9f-dad674ef48d2 | CentOS-6           | active |
| 7add5463-20a9-4d2e-8bd8-b38d959aa83f | CentOS-7           | active |
| 5ad9d51b-b6eb-44e8-98b6-9d7f69cac5df | Fedora-Atomic-25   | active |
| c42266c9-7e05-45bd-a434-287539c0dc90 | ScientificLinux-6  | active |
| 1d9a34dc-2a79-41c2-b787-4193a9c5b726 | ScientificLinux-7  | active |
| 669bef35-f60a-4bea-93cc-a57348af2ff1 | Ubuntu-14.04       | active |
| 6cd4708e-fcb0-4dbc-92f5-faf4e9aa7424 | Ubuntu-16.04       | active |
| be8c32a5-e1c2-4584-b79c-1fb6caaf4501 | demo_snapshot      | active |
+--------------------------------------+--------------------+--------+
```

```
(osclient) skapoor-air13:python_virtualenvs skapoor$ openstack server list
+--------------------------------------+----------------------+--------+-----------------------------------------+--------------+
| ID                                   | Name                 | Status | Networks                                | Image        |
| Flavor       |
+--------------------------------------+----------------------+--------+-----------------------------------------+--------------+
| a8d5f4f8-5659-4599-93ef-c32a2c96ddf8 | kapoor_shubham_instance | ACTIVE | project_2000692=192.168.1.8             | Ubuntu-16.04
(osclient) skapoor-air13:python_virtualenvs skapoor$ openstack keypair show kapoor_shubham
+-------------+-------------------------------------------------+
| Field       | Value                                           |
+-------------+-------------------------------------------------+
| created_at  | 2017-09-15T09:24:15.000000                      |
| deleted     | False                                           |
| deleted_at  | None                                            |
| fingerprint | ad:3f:45:ff:de:09:65:be:84:f3:e7:ab:22:36:57:9e |
| id          | 183015                                          |
| name        | kapoor_shubham                                  |
| updated_at  | None                                            |
| user_id     | skapoor                                         |
+-------------+-------------------------------------------------+
```

```
(osclient) skapoor-air13:python_virtualenvs skapoor$ openstack flavor list
+--------------------------------------+-----------------+--------+------+-----------+-------+-----------+
| ID                                   | Name            |    RAM | Disk | Ephemeral | VCPUs | Is Public |
+--------------------------------------+-----------------+--------+------+-----------+-------+-----------+
| 0143b0d1-4788-4d1f-aa04-4473e4a7c2a6 | standard.tiny   |   1000 |   80 |         0 |     1 | True      |
| 053c4852-dd1e-42dc-947a-fe4263548fa9 | hpc-gen2.48core | 240000 |   80 |         0 |    48 | True      |
| 110eb004-f7cc-474b-8158-14bb244cb05e | hpc-gen2.24core | 120000 |   80 |         0 |    24 | True      |
| 1792db39-f38e-43ba-ae95-96b7549b4f84 | standard.xlarge |  16000 |   80 |         0 |     6 | True      |
| 27d232d6-d245-4cf4-8ab9-a0424005184b | hpc-gen2.8core  |  40000 |   80 |         0 |     8 | True      |
| 2f24b080-287f-49a9-8219-2295cde364c3 | hpc-gen2.16core |  80000 |   80 |         0 |    16 | True      |
| 41ec2177-604b-492c-8f19-f2d7c2bc8c07 | io.70GB         |  10000 |   20 |        70 |     2 | True      |
| 544e940c-4b9b-4f54-ab6f-f1ee1792fe48 | hpc-gen2.2core  |  10000 |   80 |         0 |     2 | True      |
| 58bbbf4c-e174-485f-b050-b0cc86c0f677 | hpc-gen1.16core |  60000 |   80 |         0 |    16 | True      |
| a82b2b5f-6788-41fd-80cb-ed7576ee1e7c | hpc-gen1.8core  |  30000 |   80 |         0 |     8 | True      |
| af9fa76e-818a-421e-9142-0341e7818d90 | io.340GB        |  40000 |   20 |       340 |     8 | True      |
| ba8f9270-93fe-47ee-b402-714a1352f190 | hpc-gen1.1core  |   3750 |   80 |         0 |     1 | True      |
| c0c7bb30-2679-4e0d-94ab-4395237f505e | hpc-gen1.4core  |  15000 |   80 |         0 |     4 | True      |
| c1da3536-f22d-426e-bc14-ef994f1bfaa7 | io.700GB        |  80000 |   20 |       700 |    16 | True      |
| c5ffaed0-6707-4a99-9498-9ef6d34c8add | io.160GB        |  20000 |   20 |       160 |     4 | True      |
| d4a2cb9c-99da-4e0f-82d7-3313cca2b2c2 | standard.small  |   2000 |   80 |         0 |     2 | True      |
| e7b3364e-f70c-4e3b-8e5a-fa249759d14c | standard.large  |   8000 |   80 |         0 |     4 | True      |
| f363d088-4967-48ff-bc80-86c0d05ff418 | standard.medium |   4000 |   80 |         0 |     3 | True      |
+--------------------------------------+-----------------+--------+------+-----------+-------+-----------+
```

```
(osclient) skapoor-air13:python_virtualenvs skapoor$ openstack server create --flavor standard.tiny --image 6cd4708e-fcb0-4dbc-92f5-faf4e9aa7424 --key-name kapoor_shubham kapoor_shubham_instance_2
+-----------------------------+-----------------------------------------------------+
| Field                       | Value                                               |
+-----------------------------+-----------------------------------------------------+
| OS-DCF:diskConfig           | MANUAL                                              |
| OS-EXT-AZ:availability_zone |                                                     |
| OS-EXT-STS:power_state      | NOSTATE                                             |
| OS-EXT-STS:task_state       | scheduling                                          |
| OS-EXT-STS:vm_state         | building                                            |
| OS-SRV-USG:launched_at      | None                                                |
| OS-SRV-USG:terminated_at    | None                                                |
| accessIPv4                  |                                                     |
| accessIPv6                  |                                                     |
| addresses                   |                                                     |
| adminPass                   | VAYj6QiSnN7t                                        |
| config_drive                |                                                     |
| created                     | 2017-09-15T12:07:17Z                                |
| flavor                      | standard.tiny (0143b0d1-4788-4d1f-aa04-4473e4a7c2a6) |
| hostId                      |                                                     |
| id                          | 61076662-6ca5-44af-93b4-7b1b832a644a                |
| image                       | Ubuntu-16.04 (6cd4708e-fcb0-4dbc-92f5-faf4e9aa7424) |
| key_name                    | kapoor_shubham                                      |
| name                        | kapoor_shubham_instance_2                           |
| progress                    | 0                                                   |
| project_id                  | 2d9e321be82f4066a3824284ce47b17d                    |
| properties                  |                                                     |
| security_groups             | name='default'                                      |
| status                      | BUILD                                               |
| updated                     | 2017-09-15T12:07:18Z                                |
| user_id                     | skapoor                                             |
| volumes_attached            |                                                     |
+-----------------------------+-----------------------------------------------------+
```

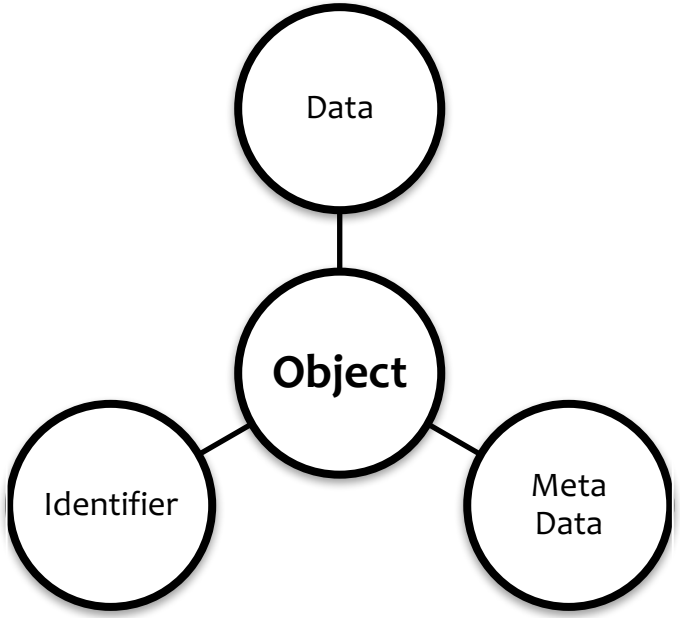


Object Storage

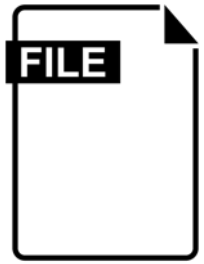
CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

What is Object Storage

- Object storage is a computer data storage architecture that manages data as objects.
- Each object has three things: Data, Metadata and Globally unique identifier.
- Different from other data storage architectures like File Storage: Data as a file hierarchy and Block Storage: Data as blocks within sectors & tracks.
- Accessed via APIs at application-level, rather than via OS at system level.
- Scalable and Self healing storage.

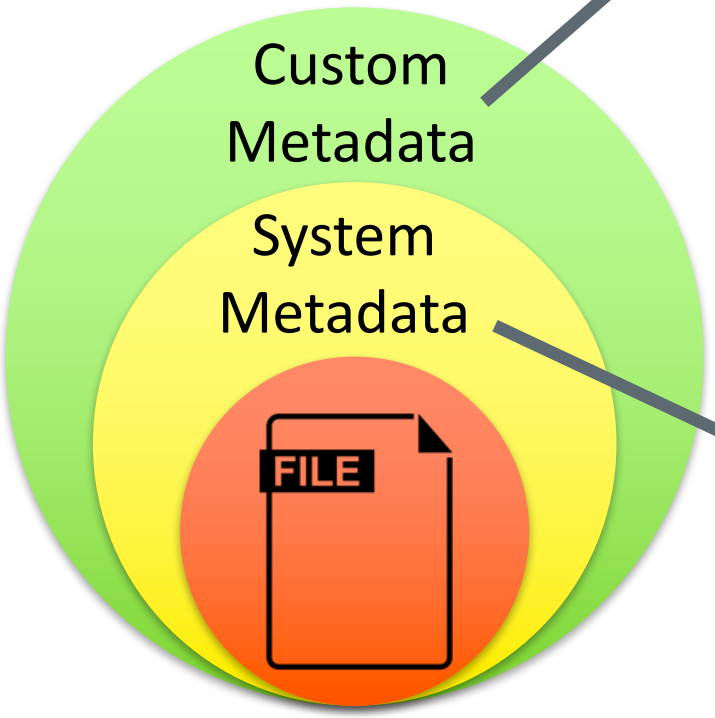


File Storage vs Object Storage



File Name: CTSCAN_Kapoor
Created by: User1
Created on: 19-09-2017
File Type: DICOM

File Storage



Object ID: 123456
Patient Name: Shubham
Patient ID: 23242
Physician Name: Dr. John
Prior1 : XYZ.DICOM
Self Destruct: 2 Year

File Name: CTSCAN_Kapoor
Created by: User1
Created on: 19-09-2017
File Type: DICOM

Object Storage

Where Object Storage Fits

On Basis of Data Type

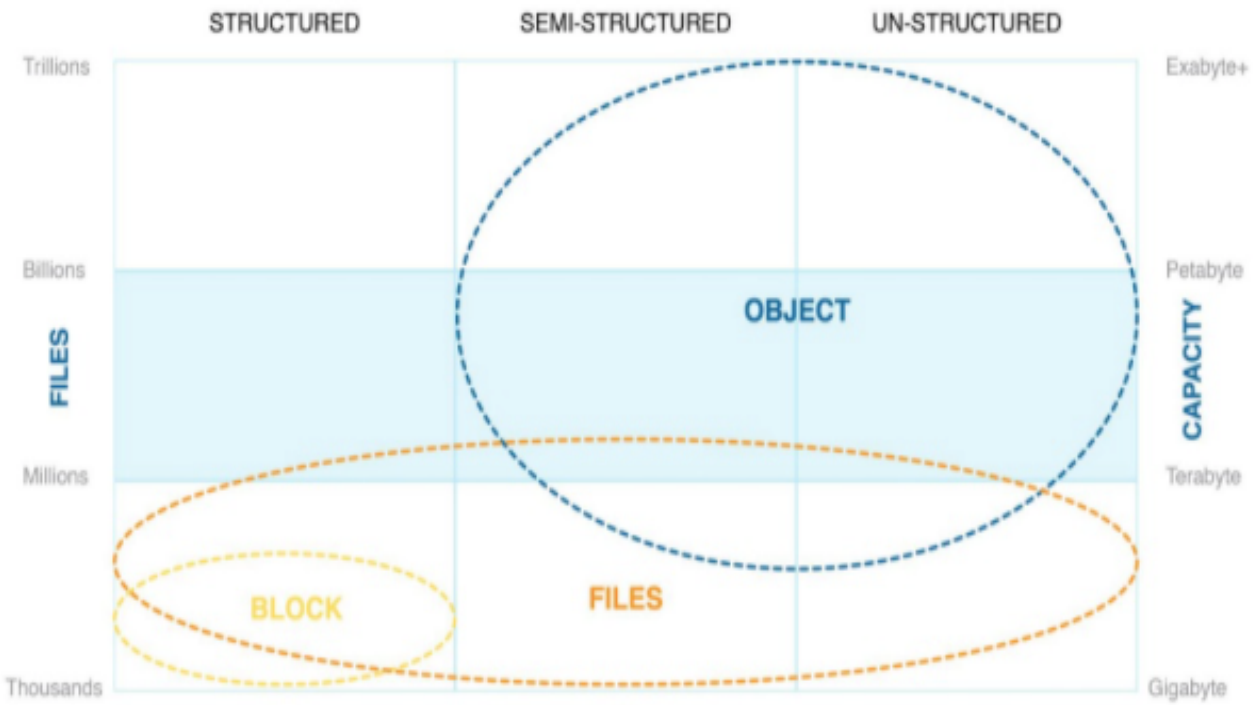
- Storage of Unstructured/ Semi structured Data like Media files, web contents, Backup Archives etc.

- Cold Storage of structured and semi structured data like Databases, Sensor Data, Log files etc.

- Archiving files in place of local tape drives.

- Big Data, large data sets

On Basis of Data Size



Where Object Storage Doesn't Fit

- Hot Data.
- Relational/OLTP Databases.
- Latency intolerant applications.
- Data with **Strict** consistency requirements.

Object Storage Around us

Amazon S3

Microsoft Azure
Blob Storage

IBM Cloud
Object Storage

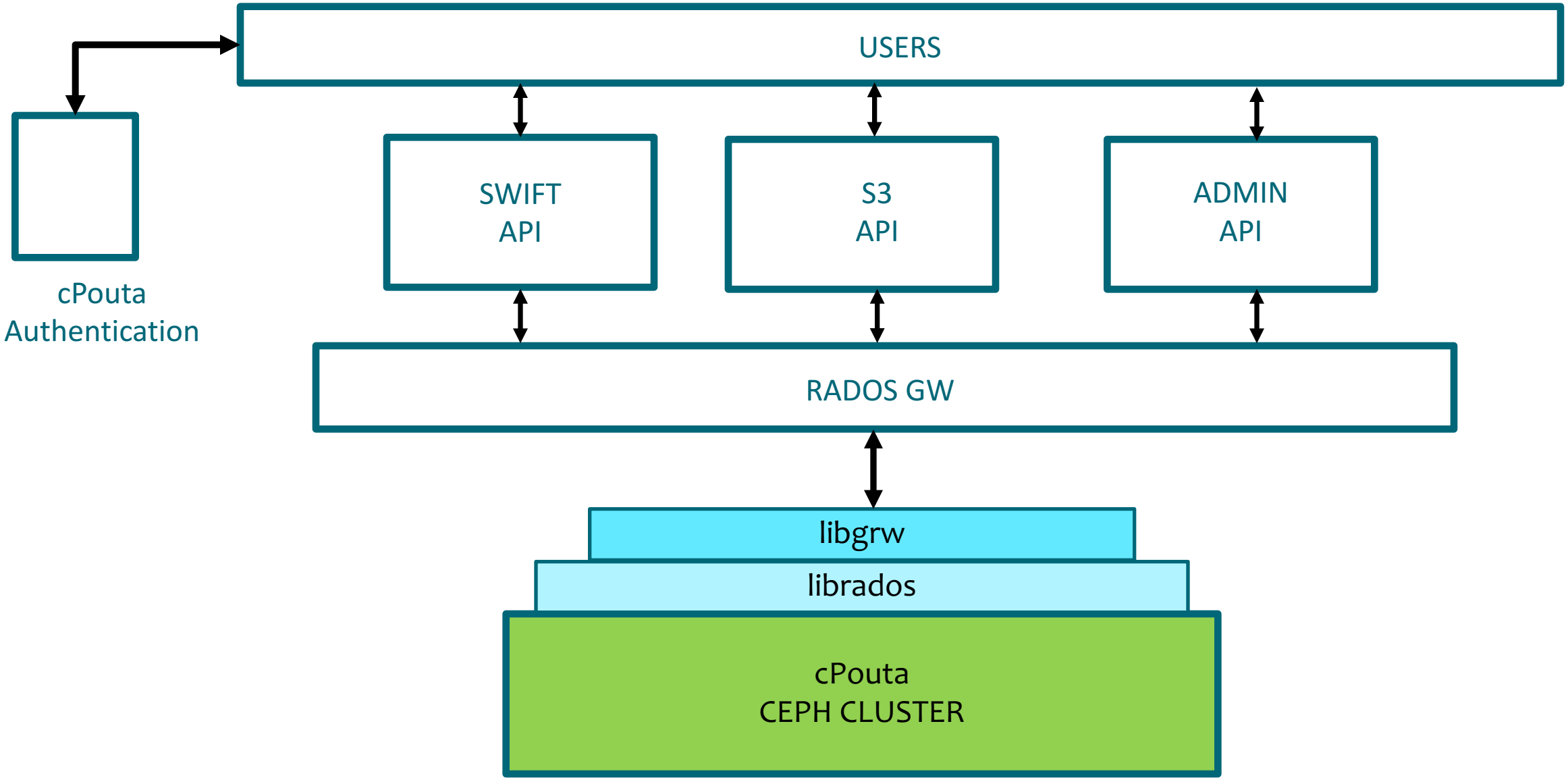
Google Cloud Storage

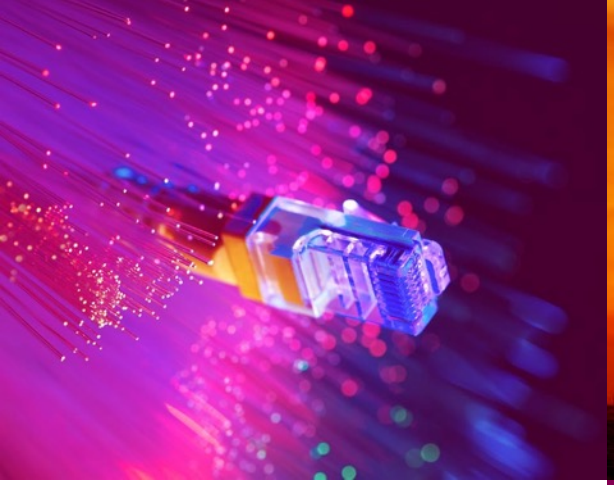
- Social Media Storage
- Big Data Analytics
- Offsite Backup/Archive
- Static Website Hosting
- Digital Archives

Object Storage in cPouta

- Launched for customers in 2018.
- BU consumption similar to existing block storage i.e. 3.5 BU/TiB/hr
- Charging on the basis of storage not transactions.
- Initial quota for object storage/project is 1 TiB.
  - Buckets per project: 1000
  - Objects per Bucket: 100000
  - Object Size: 5GB
  - For objects > 5GB, split the object into smaller segments.
- Content Agnostic, Distributed, Scalable and Highly available Data Storage.
- Access control possible for buckets/objects.
- REST API available, S3 and Swift API compatible, Supports WebUI and Swift/S3 CLI tools
- CSC's new Allas data lake to support similar APIs .

Object Storage in cPouta



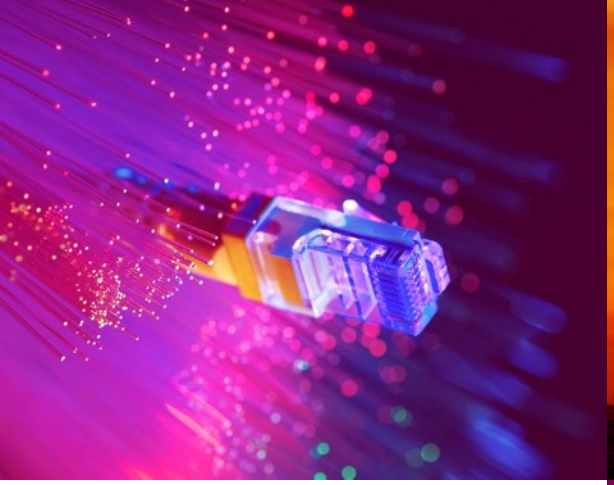


Persistent Volumes

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Persistent Data Volumes

- In Pouta, VM:s have only small local (virtual) disk (Typically 80GB)
- Virtual data volumes can be created.
- Volumes can be attached to one VM at a time.
- A project can have several volumes.
- Management with web interface or command line client
- Data stored in Persistent volumes is resilient to Disk Failures, Server failures, Accidental deletion of VMs, Crashing of VMs
  - But not to human errors → No Backups.
- Volumes are project specific, not user specific, but could be transferred to other projects in same cloud service.



CSC

Pouta: Managing Projects, Billing, VM Lifecycle…

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Pouta : Managing Project

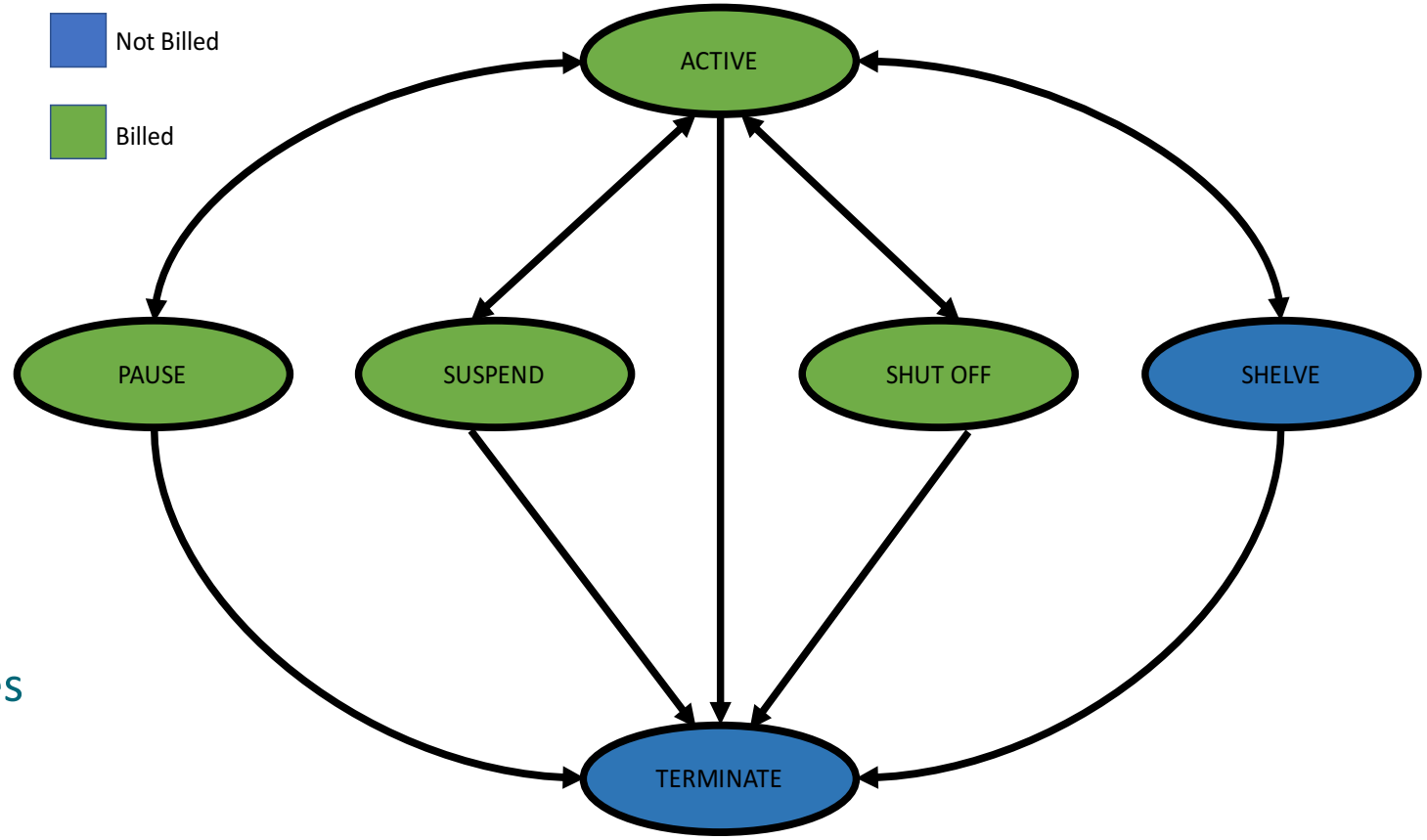
- A Pouta project contain a set of resources: cores, memory, storage, ip-addresses
- A default project contains:
  - For cPouta: 8 cores, 32 GB memory, 1 TB disk space, 2 floating IP addresses .
  - For ePouta: Negotiated between customer and CSC
- If needed you can ask for more resources for your project.
- Project members can build one or several VMs and volumes based on the granted resources.
- When VMs and Volumes are active they are consuming billing units (even if no one is using them).
- Project members can manage other members machines and volumes too.
- Your CSC account can be a member in many cPouta projects.

Pouta : Billing

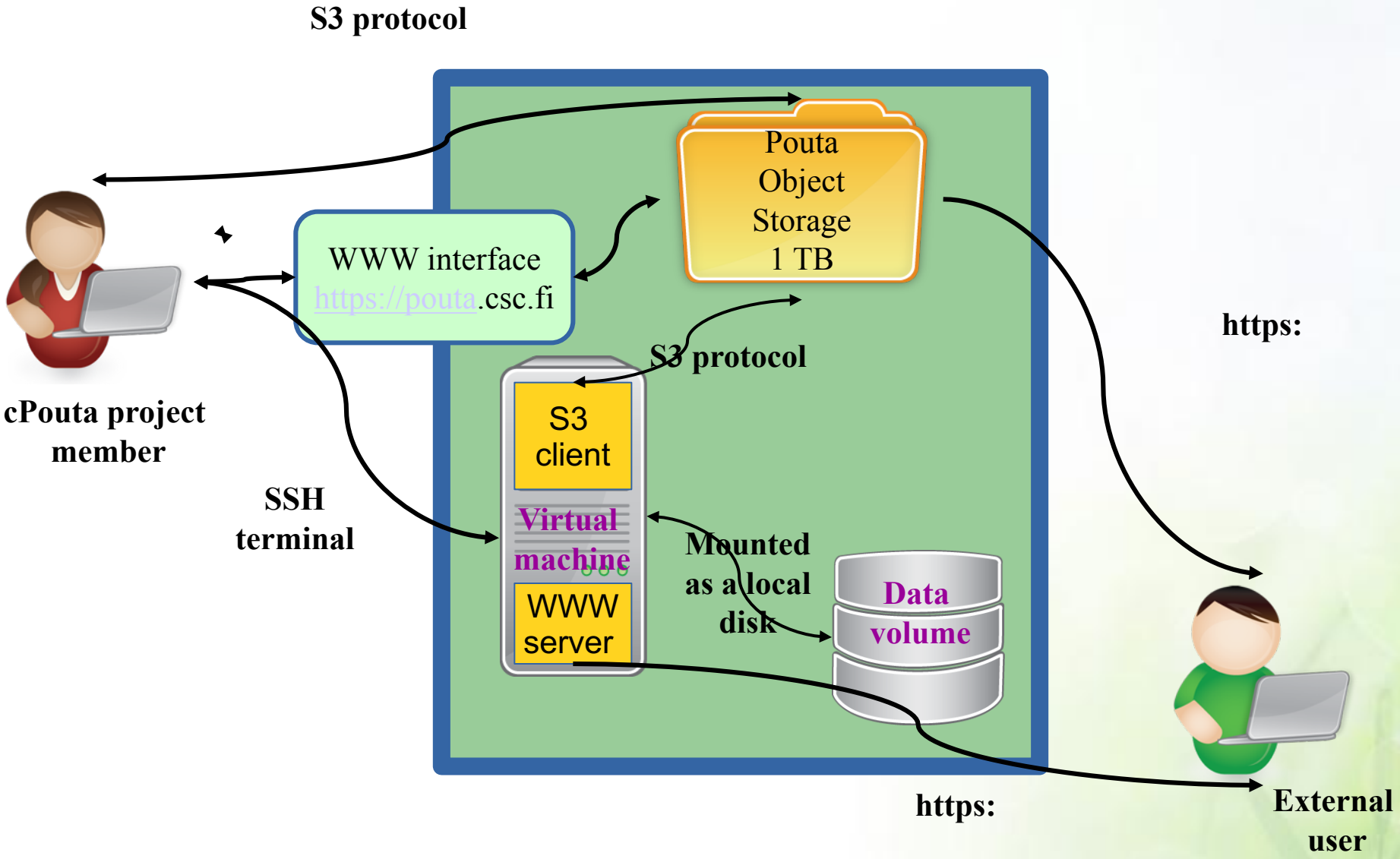
- In Pouta you are hourly billed for
  - VM usage (Based upon your flavor: https://research.csc.fi/pouta-flavours)
  - Volume Storage (3,5 BU/TiB/hr)
  - Object Storage (3,5 BU/TiB/hr)
  - Public IPs (0,2 BU/hr)
- VMs start consuming BUs once you create them
  - They consume BU regardless you use them or not.
  - Shutting down VM does not stop them consuming BUs.
  - You can Shelve/Terminate your VM for stopping BU consumption.
- Object/Volume Storage start consuming BUs once you create them
  - Storage volumes consume BUs even if they are not attached to virtual machines.
- Floating IPs are billed if they are allocated to the project or assigned to a virtual machine.

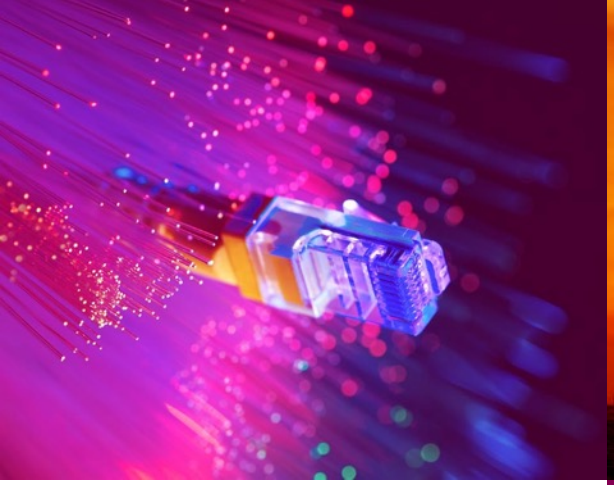
Pouta : VM States

- **Active** – Consumes billing units regardless of the real usage.
- **Shut off** – Not active, but still reserves the resources. Consumes still billing units.
- **Suspended** – Temporarily suspended. Current state saved. Can be revoked. Consumes billing units
- **Shelved** – VM is shut off, resources are freed & Sate is saved. Can be later on revoked if resources are free (un-shelved). Does not use billing units.
- **Terminate** – Removes the Virtual Machine.



cPouta in action





Building Application Stack on Pouta VM

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Installing software to your VM

- The VM images provided by CSC include only just the basic linux tools.
- You can/must add the tools you need with using tools like
- **System level repository installation**:
  - Centos and Scientific linux: **sudo yum**
  - Ubuntu: **sudo apt**
  - add missing libraries and linux commands and many applications too
- **Compile codes or download pre-compiled binaries.**
- Install Docker and use Docker images.
- Use Conda!

sudo command for system administration

The default user, cloud-user, does not have superuser rights, but can do admin operations with **sudo**.

sudo *linux-command-to-execute*

- Repository installations
- System libraries and directories
- User accounts

e.g.

```
sudo reboot
sudo yum install nano
sudo nano /etc/yum.conf
sudo useradd teppo
```

Repository installation in Centos and RedHat with yum

System wide installation of libraries and tools
Many application software are also available this way

yum help	List subcommands and options
sudo yum install *package*	Install a package from repository
sudo yum update	Update one or all packages in the system
yum provides *filename*	Check what packages include the defined file
yum search *term*	Search package names and descriptions
sudo yum localinstall *package*	Install locally available rpm file
sudo yum remove *package*	Remove a package

Repository installation in ubuntu with apt-get

System wide installation of libraries and tools
Many application software are also available this way

apt-get --help	List commands and options
sudo apt-get install *package*	Install a package from repository
sudo apt-get update	Update one or all packages in the system
apt-file *filename*	Check what packages include the defined file
apt-cache search *term*	Search package names and descriptions
sudo yum localinstall *package*	Install locally available rpm file
sudo apt-get remove *package*	Remove a package

Conda /Bioconda

- Easy way to install software tools together with their dependencies
- Bioconda- repository contains over 700 bioscience tools
- Does not need superuser privileges
- For installing conda and browsing bioconda packages, check bioconda home page:

  https://bioconda.github.io/

- Once you have conda installed, you can install application software with commands like:

conda create -n aligners bwa bowtie hisat star
source activate aligners
bwa

Getting Started, Doubts, Issues with Pouta Services

Contact servicedesk@csc.fi

Coffee break

Exercise Set 2: Atleast 3 of Following Exercises

- Exercise 4 - Install Docker CE & run a RStudio server in Docker Container
- Exercise 5 - Build your own RStudio Server
- Exercise 6 - Installing software with Conda
- Exercise 7 – Installing Stacks server in Ubuntu VM
- Exercise 8 – Install OpenStack CLI & use it
- Exercise 9 - Create Snapshot of VM
- Exercise 10 – Manage your Persistent Volumes
- Exercise 11 – Create your own Bucket & Object using WebUI
- Exercise 12 - Upload Object to your Bucket using s3cmd client