Leibniz Supercomputing Centre

of the Bavarian Academy of Sciences and Humanities

LRZ Compute Cloud for AI support

What includes in the course

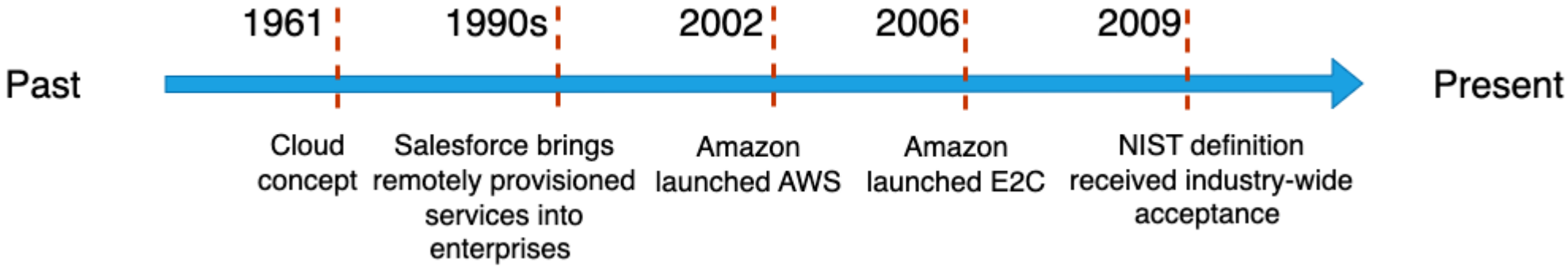
First steps	10:00 - 10:45	• A brief introduction to cloud computing • Meet the LRZ Compute Cloud
Hands-on sessions	11:00 - 12:00	• Extending the LRZ AI capabilities with the Compute Cloud
		Lunch break
	13:00 - 14:15	• Databases and the LRZ Compute Cloud • Managing the LRZ Compute Cloud from the command line
	14:30 - 15:45	• Secure AI workflow with LRZ middleware
Final touch	15:45 - 16:00	• Wrap-up

A Brief History of Cloud

Birth of Cloud

Tech giants such as Amazon, Google decides to rent out their machines in an off-season to avoid money draining by the idle machines.

Main timeline of Cloud history



Six Fun Facts of Cloud

1. Banking has the most activity in the history of cloud computing
2. China has the largest data center in the world:
   China Telecom Data Center located at Hohhot, size 100 hectares (~140 football fields)
3. A data center from world top 10 largest centers consumes as much power as a mid-sized town
4. Cooling systems consume ~70% of the total energy cost of a data center
5. Amazon dominates the Cloud Market: 32%, Q1 2023
6. ~34 percent of workers prefer working in the cloud and will look for a new job if required to return to the office

Hohhot

What is the Cloud Computing for you?

- Stay quietly in the background until something goes wrong...
- On-demand delivery of resources, e.g. compute resources (CPU, GPU), storage, databases.
- Quickly provisioned, access remotely,
- Pay as you go: idea for individuals, small/middle-sized business
- Flexible environments
- ….

It is not about the servers, but the services

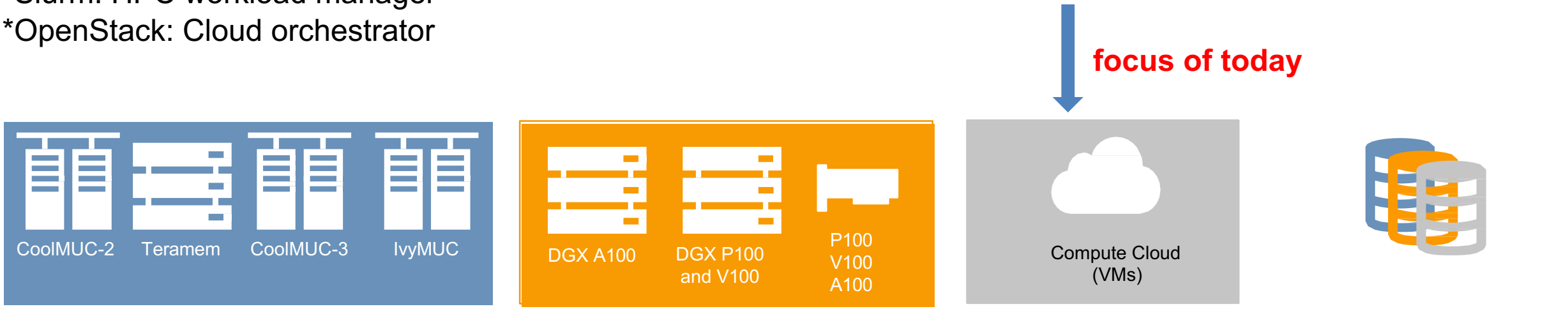
Cloud Computing definition:

Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources*

* Thomas, E., Zaigham, M. and Ricardo, P., 2013. *Cloud Computing Concepts, Technology & Architecture*. Prentice Hall.

High Level Overview of Main LRZ Resources

*Slurm: HPC workload manager
*OpenStack: Cloud orchestrator



HPC systems (Slurm)

AI Systems (Slurm)

Compute Cloud (OpenStack)

Data Science Storage

AI Ready

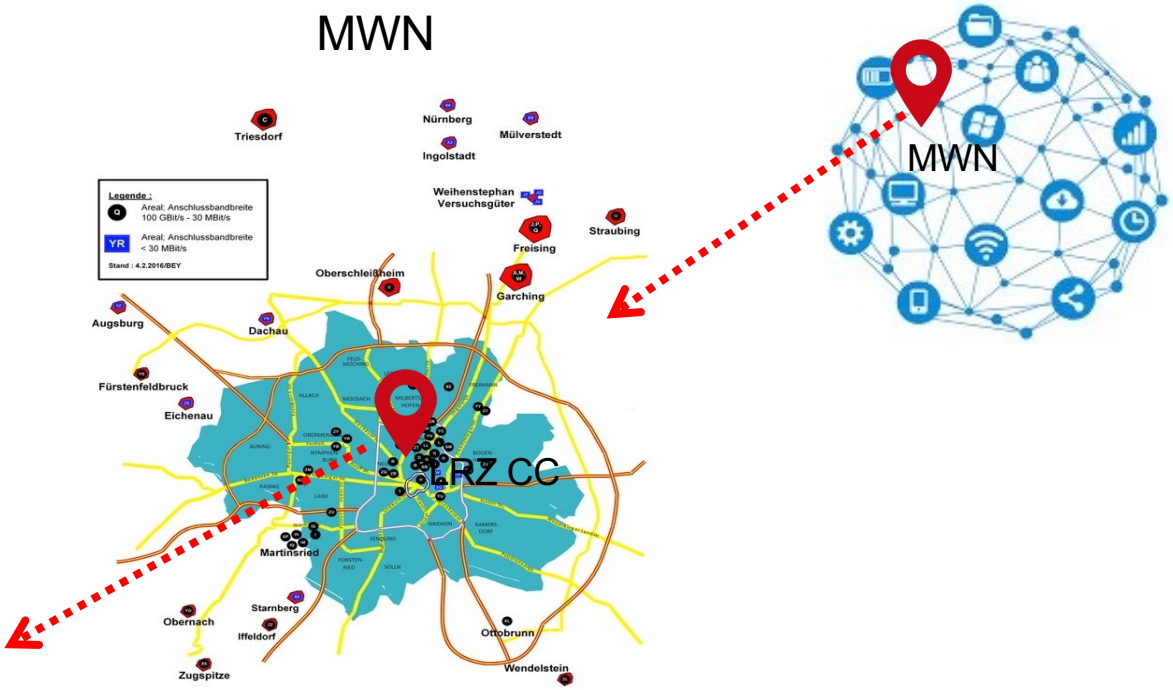
Interfaces to use Slurm and OpenStack are very different!

*Zhou, N., Zhou, H. and Hoppe, D.,Containerization for High Performance Computing Systems: Survey and Prospects. *in IEEE Transactions on Software Engineering*, 2023

The LRZ Compute Cloud at a Glance

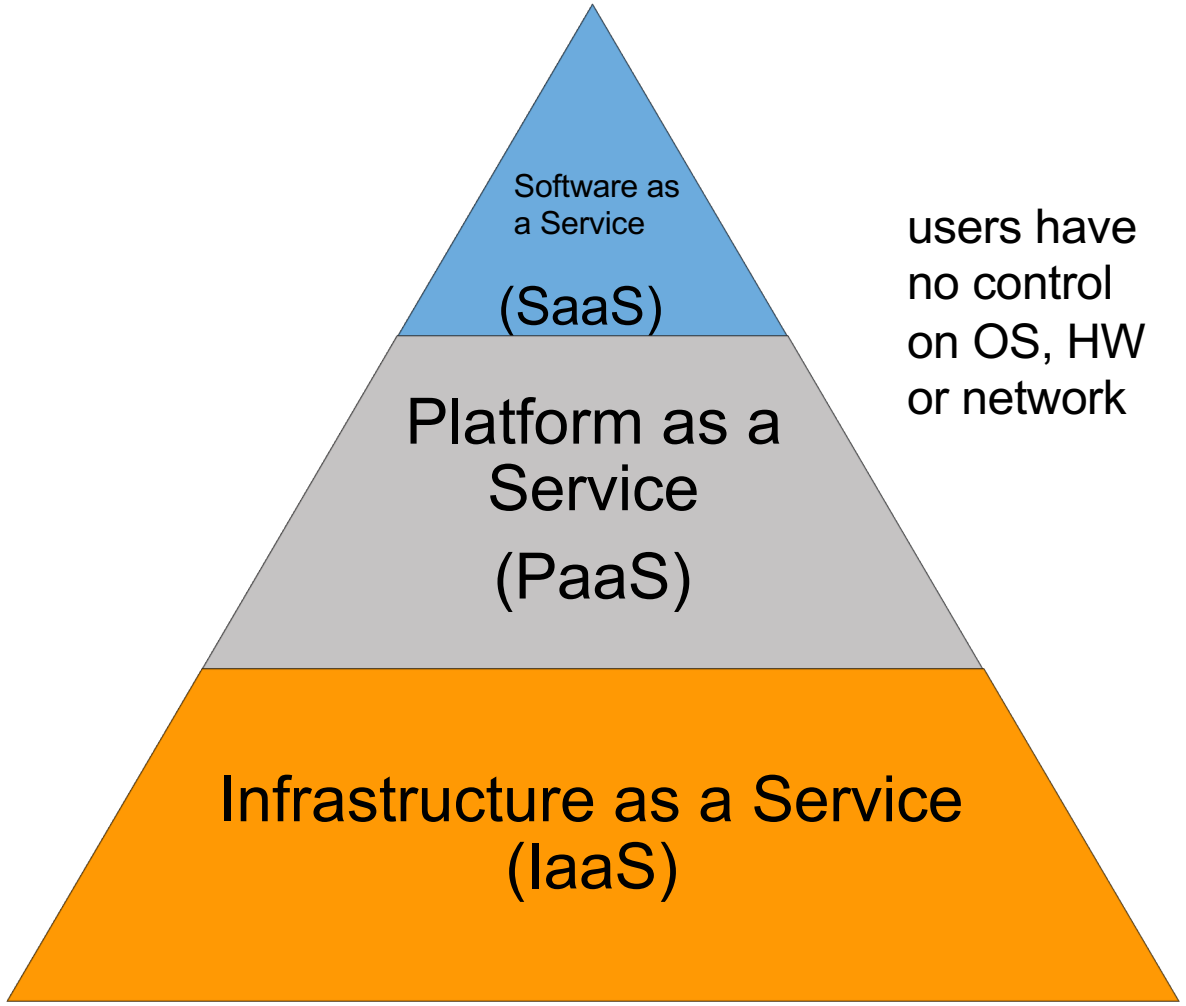
Internet

MWN



The LRZ compute cloud located within MWN

Cloud Services



users have no control on OS, HW or network

- SaaS – Fully developed software solution to be used,e.g. Google drive, Tiktok (to take online orders).
- PaaS – Provides an environment for an application: provides a framework on top of which to build, deploy, and manage software products, e.g. Google APP engine
- IaaS - Provides a completely virtualized computing infrastructure provisioned and managed over the internet, e.g. Amazon Elastic Compute Cloud (Amazon EC2), LRZ Compute Cloud

Further reading: Cloud Computing Characteristics*

A consumer can request and receive access to a service offering without an administrator or supporting staff having to fulfil the request manually.	**On-demand**
Cloud services should be easy to access. Services are not locked to devices, but can be accessed whenever and wherever users want.	**Broad network access**
Ability to grow with user demand.It should be relatively easy for the provider to add more resources. Users have "infinite" resources!	**Resource elasticity**
Cloud services must have the ability to measured usage. Usage can be quantified using various metrics, such as time, bandwidth used, and data used. This ability to measure allow what is known as pay as you go model.	**Measured service**
When resources (e.g. memory, CPU, network bandwidth) are not used, they should be released, which can be used by others.	**Resource pooling**

*Xiao, Z. and Xiao, Y., 2012. Security and privacy in cloud computing. *IEEE communications surveys & tutorials*, *15*(2), pp.843-859.

A List of Cloud Orchestrators

- OpenStack
- Kubernetes (k8s)
- Apache Mesos, Apache YARN
- Ansible
- Docker Swarm
- ...

Your homework:

What are the differences? Which one to choose? Which one is better maybe?

OpenStack: The Engine of the LRZ Compute Cloud

- What do we need for transforming a set of resources (data center) into a cloud?
  - to manage/admin the hardware
  - to provision machines to users
  - to allow users to authenticate
  - to manage the network across resources
  - …

> OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanism.

- OpenStack bundles together a bunch of different technologies, addressing the different needs transforming resources into a Cloud Service

OpenStack - Terminology

- Image

A single file which contains a virtual disk with a bootable operating system installed on it. Images are like a template of a computer's root drive. They contain the operating system and can also include software and layers of your application, such as database servers, web servers, and so on.

OpenStack - Terminology

- Instance or server

A copy of an image running as a virtual server the cloud. We will also call it server.

servers always in a status out of ..

ACTIVE: The server is active.
BUILD: The server has not yet finished the original build process.
DELETED: The server is deleted.
ERROR: The server is in error.
HARD_REBOOT: The server is hard rebooting. This is equivalent to pulling the power plug on a physical server, plugging it back in, and rebooting it.
MIGRATING:
PASSWORD:
PAUSED:
REBOOT: The server is in a soft reboot state. A reboot command was passed to the operating system.
REBUILD:
RESCUE:
RESIZE:
REVERT_RESIZE:
SHELVED: The server is in shelved state. Depends on the shelve offload time, the server will be automatically shelved off loaded.
SHELVED_OFFLOADED: The shelved server is offloaded (removed from the compute host) and it needs unshelved action to be used again.
SHUTOFF: The server was powered down by the user, either through the OpenStack Compute API or from within the server.
SOFT_DELETED:
SUSPENDED:
UNKNOWN:
VERIFY_RESIZE:

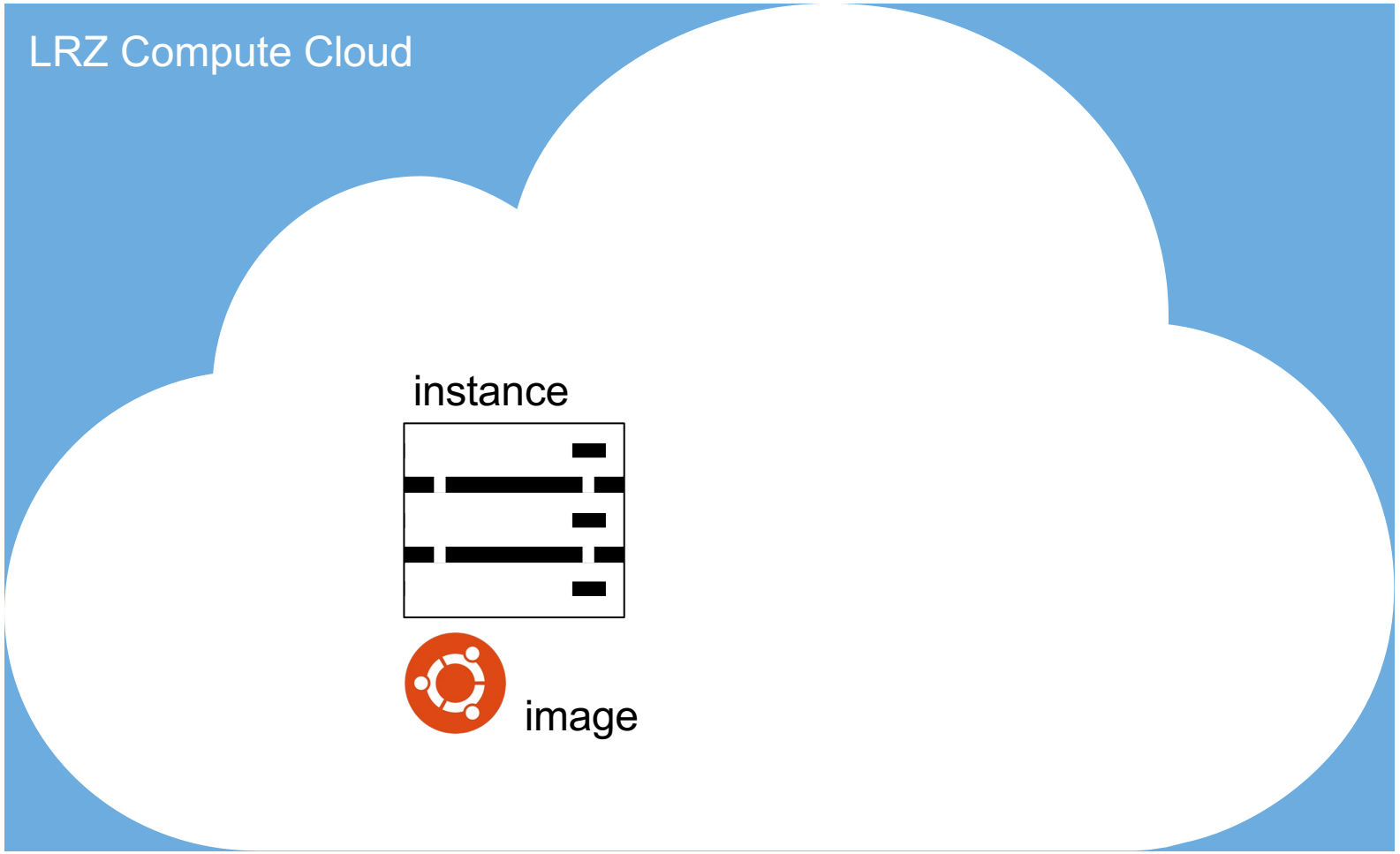
OpenStack - Terminology

- Flavor

Flavors define the compute, memory, and storage capacity of instances. To put it simply, a flavor is an available hardware configuration for a server.

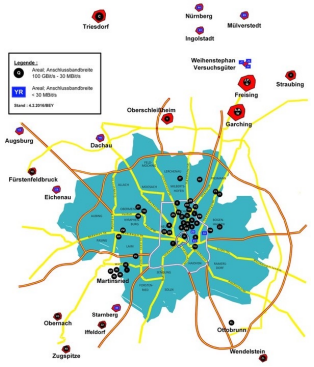
Name	vCPUs	RAM	Remarks	Access
tiny	1	512 MB	for testing purposes only, most Operating Systems will not boot due to restricted resources	public
nvidia-v100.2	40	700 GiB	use 2 GPUs on a GPU node (use entire GPU node)	restricted, contact us
nvidia-v100.1	20	350 GiB	use 1 GPU on a GPU node	restricted, contact us
lrz.xlarge	10	47.5 GiB	use 1/4 compute node	public
lrz.xhuge	48	1488 GiB	use 1/4 of the hugemem node	restricted, contact us
lrz.small	1	4.75 GiB	use 1/40 compute node	public
lrz.medium	2	9.5 GiB	use 1/20 compute node	public
lrz.large	4	19 GiB	use 1/10 compute node	public
lrz.huge	24	744 GiB	use 1/8 of the hugemem node	restricted, contact us
lrz.4xlarge	40	190 GiB	use entire compute node	restricted, contact us
lrz.4xhuge	192	5952 GiB	use entire hugemem node	restricted, contact us
lrz.2xlarge	20	95 GiB	use 1/2 compute node	restricted, contact us
lrz.2xhuge	96	2976 GiB	use 1/2 of the hugemem node	restricted, contact us

The LRZ Compute Cloud at a Glance



Internet

MWN

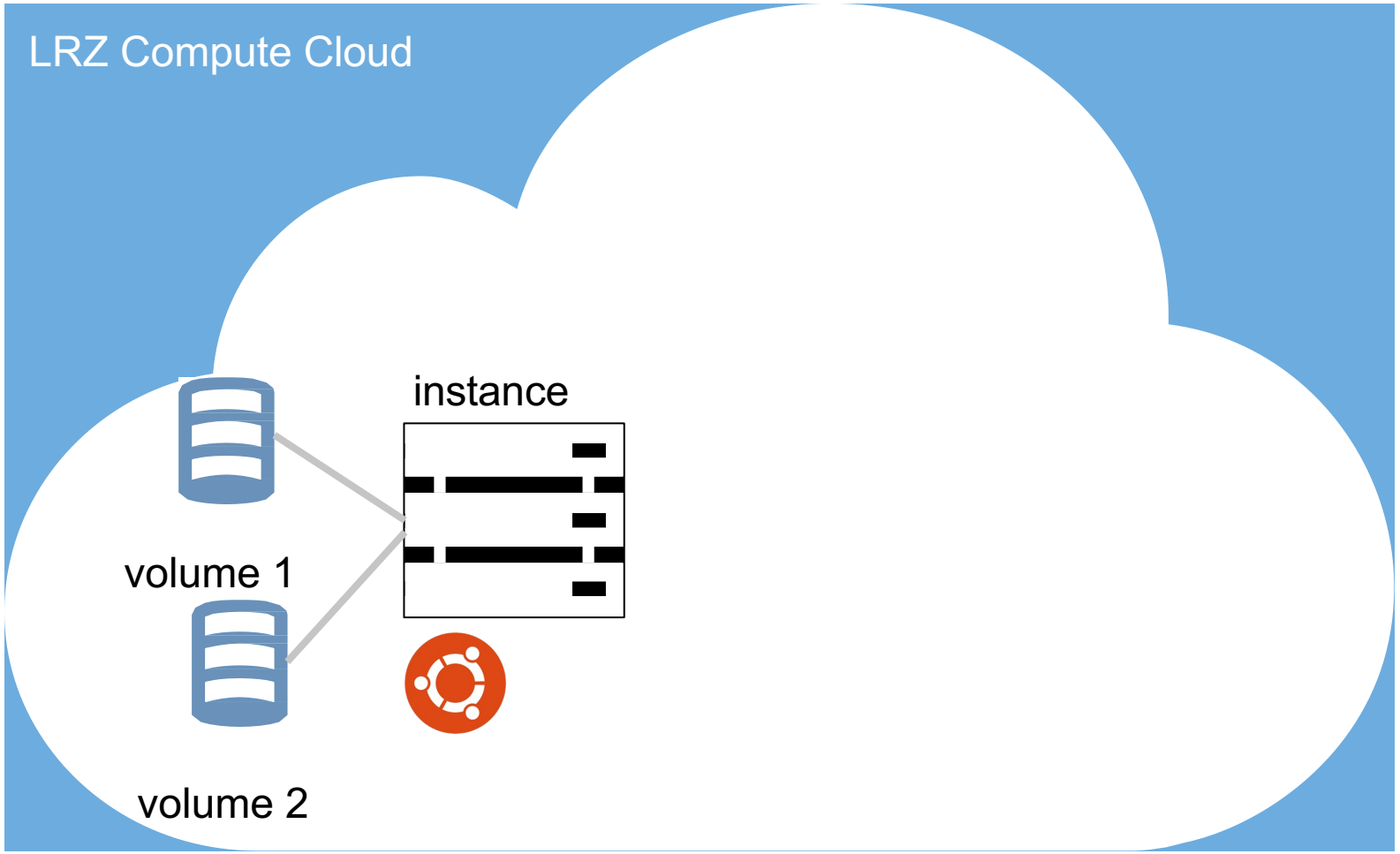


OpenStack - Terminology

- Volume

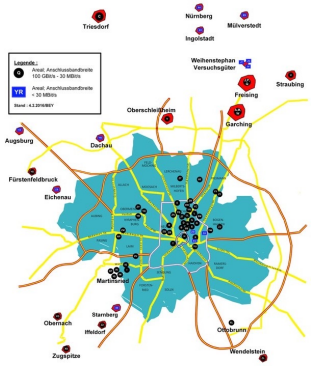
  A volume is a detachable block storage device, similar to a USB hard drive. You can attach a volume to only one instance. But an instance can attach several volumes

The LRZ Compute Cloud at a Glance



Internet

MWN



OpenStack - Terminology

- Networking

  OpenStack provides networks, subnets, and routers as object abstractions. Each abstraction has functionality that mimics its physical counterpart: networks contain subnets, and routers route traffic between different subnets and networks. Instances are created within internal private networks. These networks can be routed to external networks (e.g., Internet or or MWN) via a virtual router.

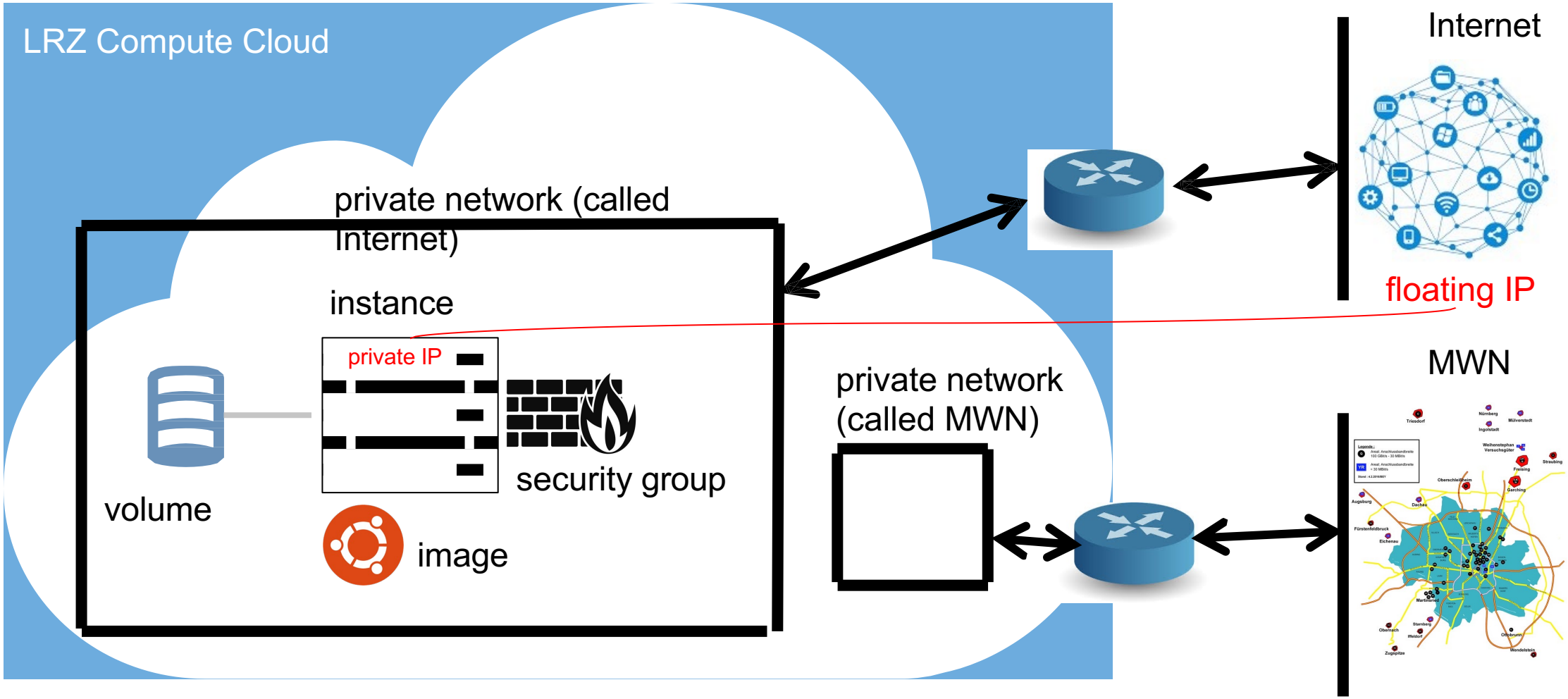
- Private and Floating IP

  Each instance has a fixed IP within its private Network. That IP can be associated to an IP of the external network that network is connected by means of what it is called *floating IP address*. The floating IP address will allow addressing the instance from the outside.

- Security group

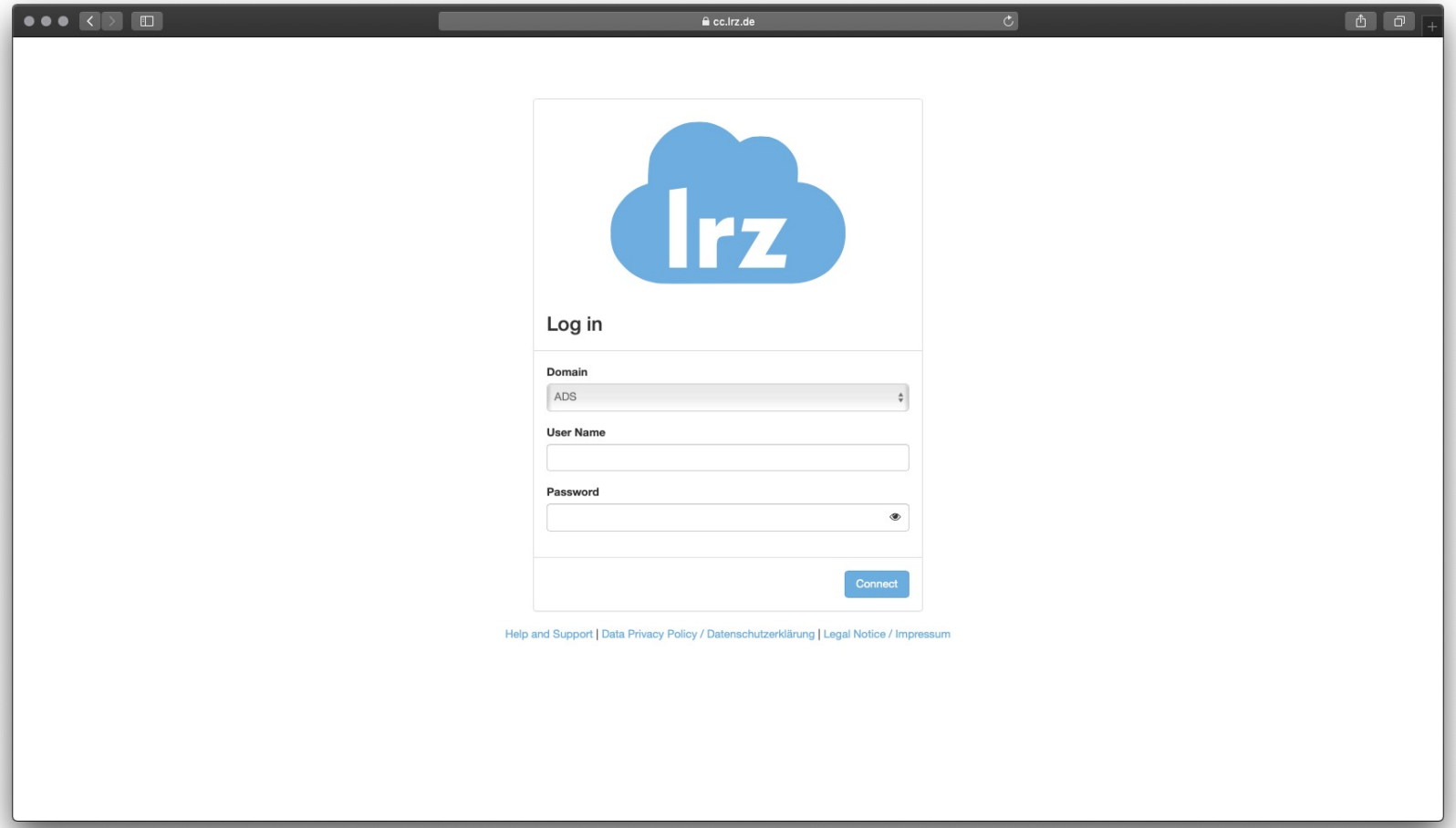
  A security group acts as a virtual firewall for servers and other resources on a network. It is a container for rules for allowing different types of network traffic to and from an instance.

The LRZ Compute Cloud at a Glance



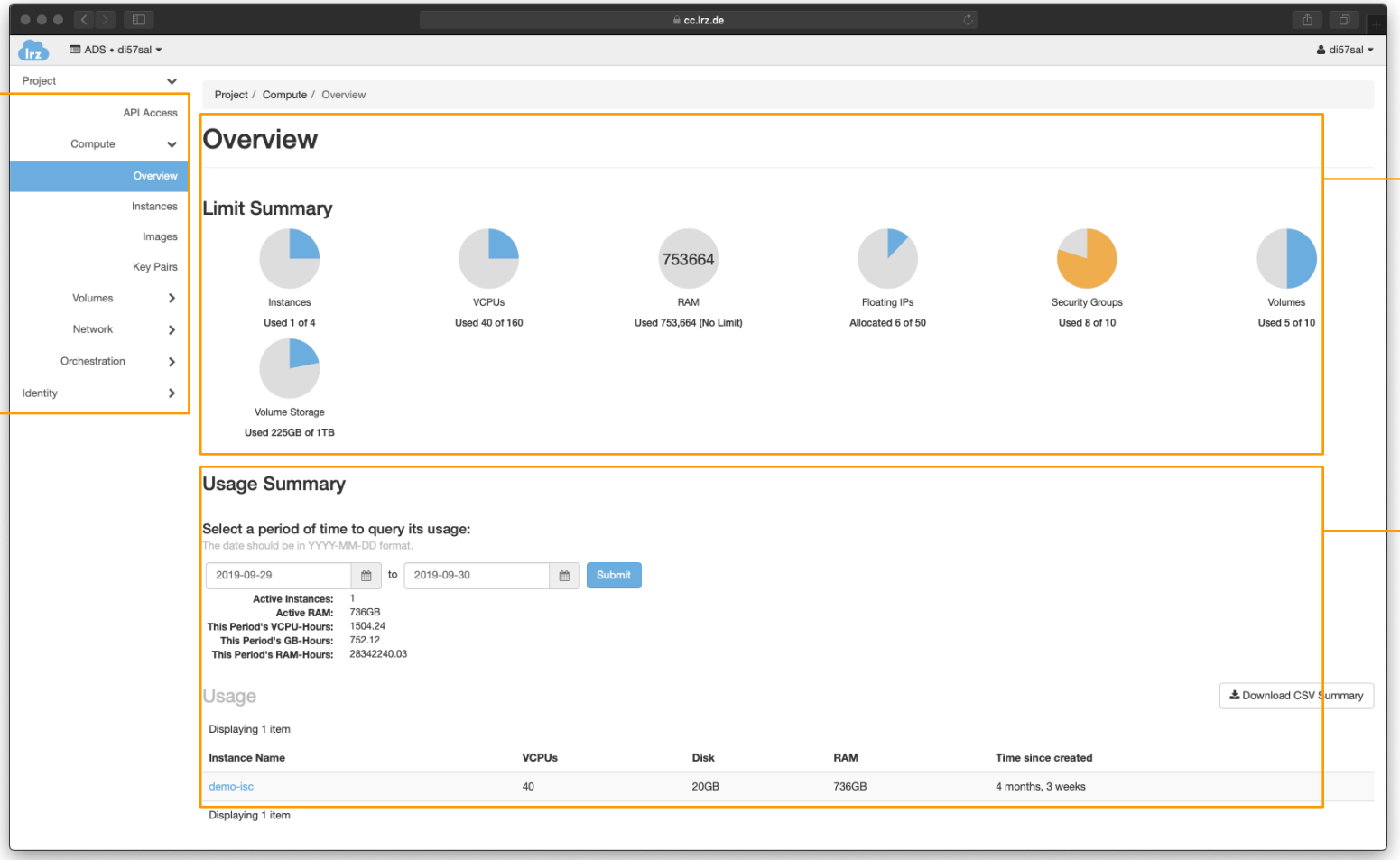
The Compute Cloud Web Interface

https://cc.lrz.de



The Compute Cloud Web Interface

Left panel.
Allows you to operate with the CC.

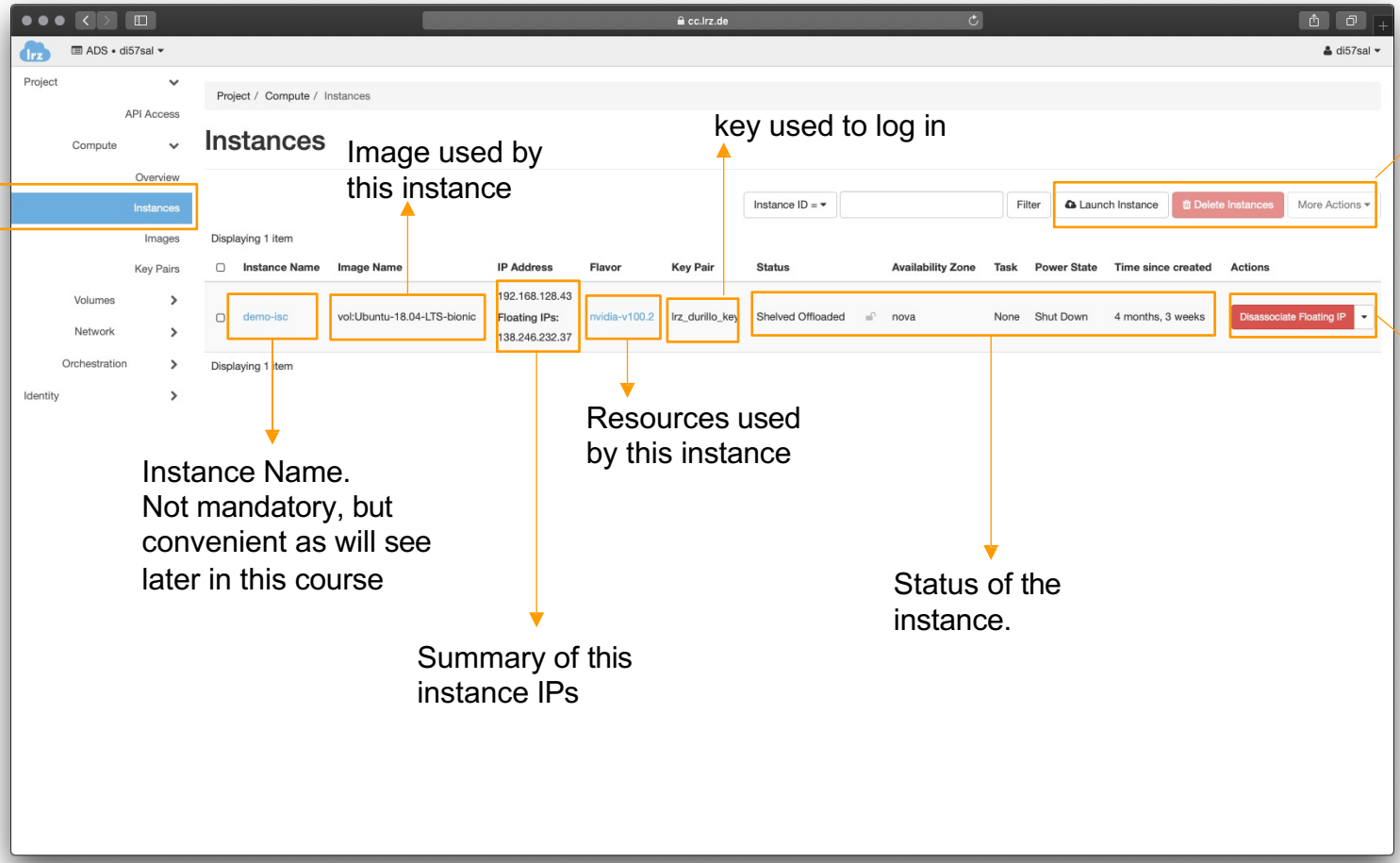


Resources Limits.
Shows you what you are allow and what you have in use.

Summary.
Show the instances you have. They might be in different status.

The Compute Cloud Web Interface

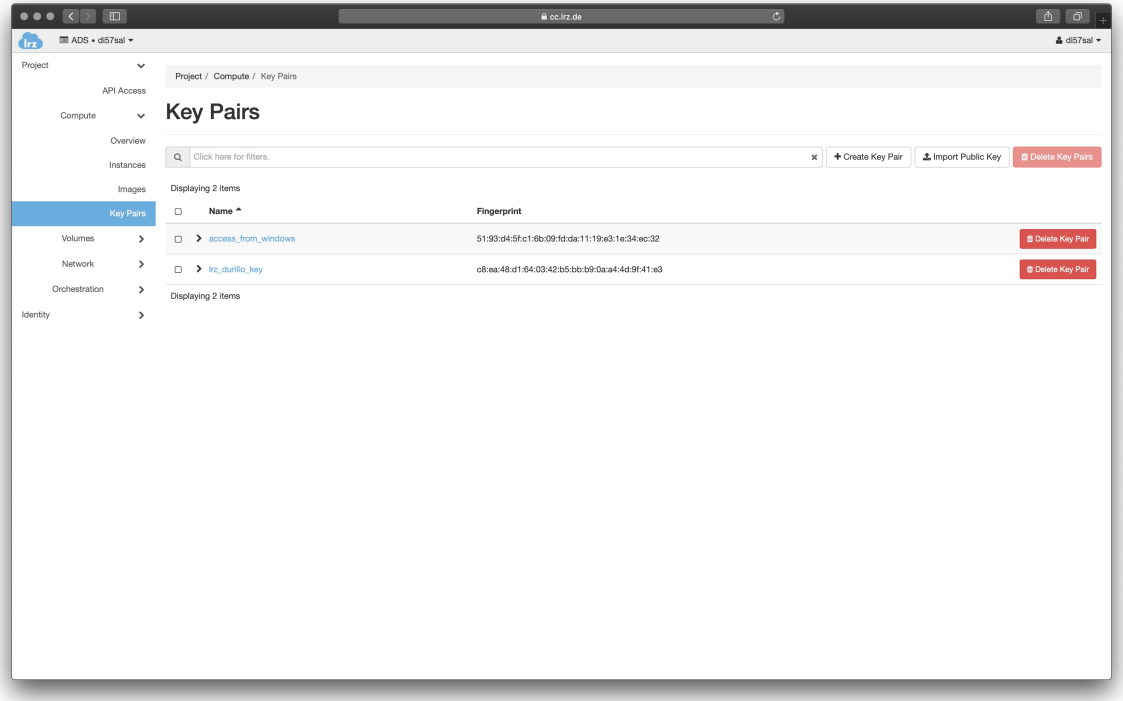
Allows a more informative view of our instances

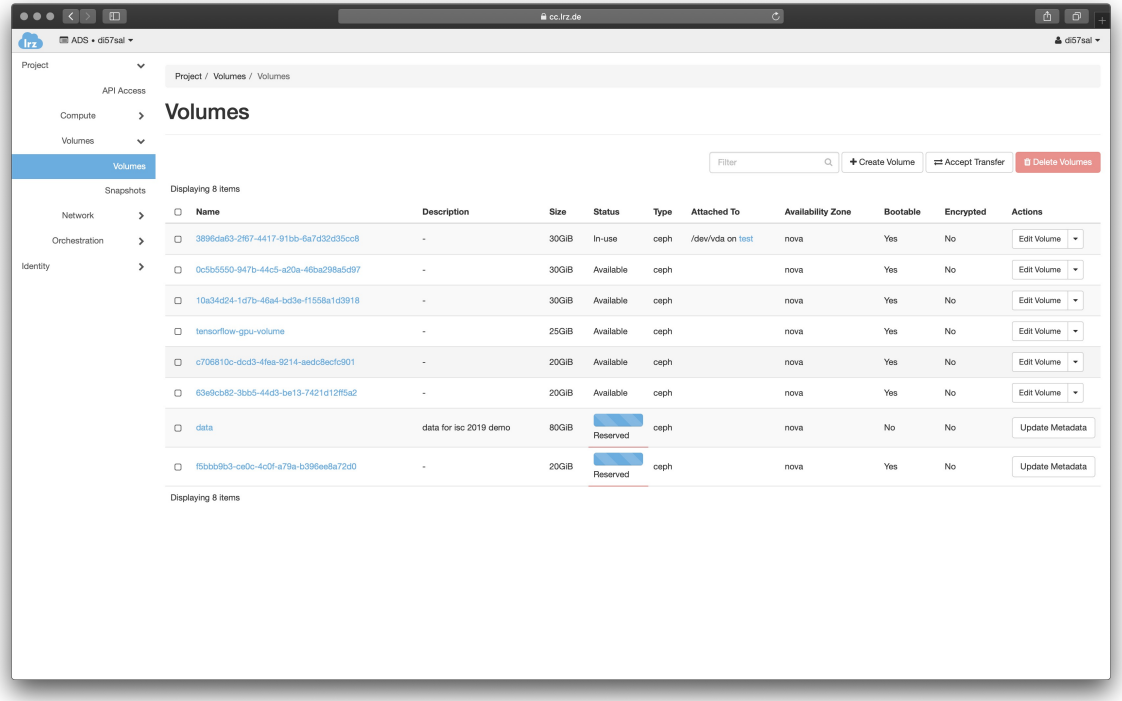


Allows creating new instances / Deleting existing ones

Allows operating this instance (e.g., attach a volume, associate new floating IP)

The Compute Cloud Web Interface





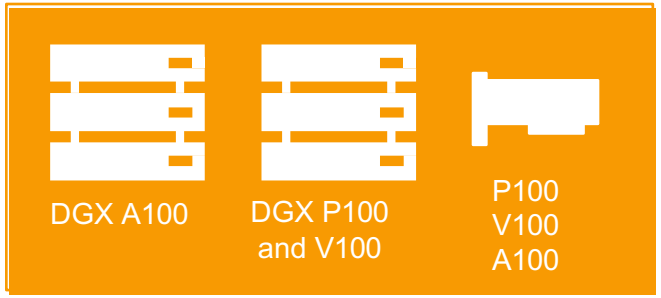
Use-case: Generate Container Images for the LRZ AI Systems

HAPPY USER

Dockerfile

```
FROM ubuntu:22.04
RUN apt install -y
python
RUN ...
ADD ...
```

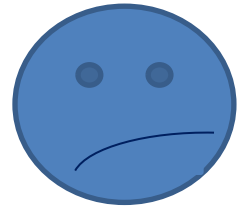
+



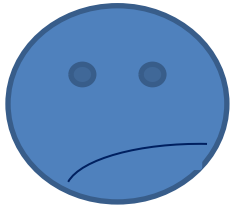
AI Systems (Slurm)

=

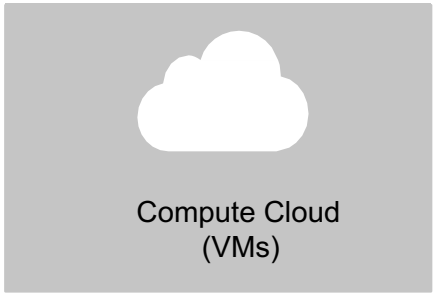
UNHAPPY USER



UNHAPPY USER



+

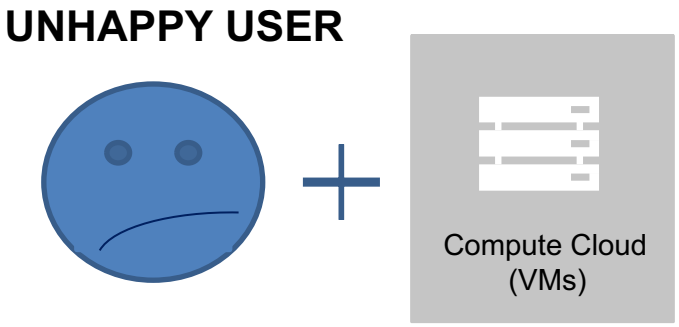


=

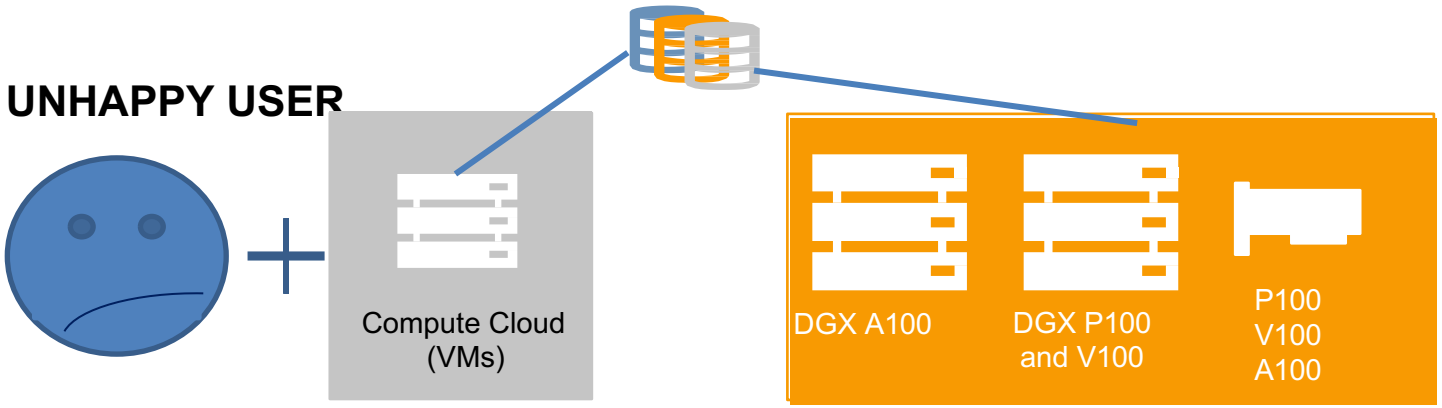
HAPPY USER

Use-case: Generate Container Images for the LRZ AI Systems

Step 1: Server on the CC

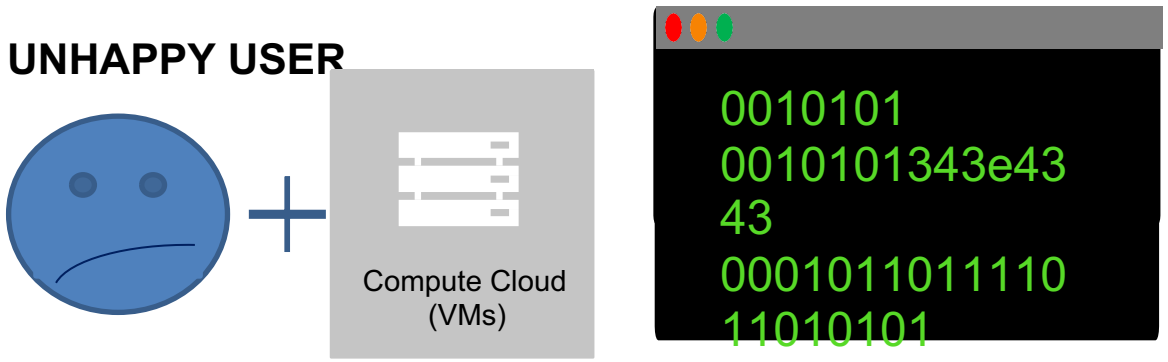


Step 2: Shared Storage CC and LAI

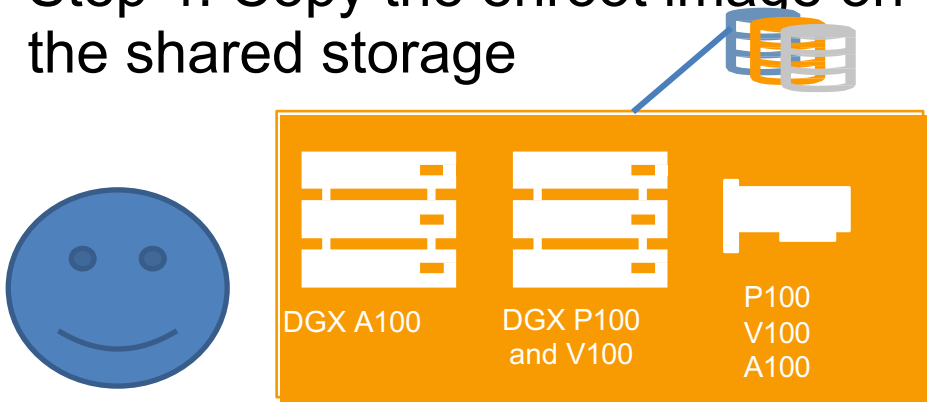


AI Systems (Slurm)

Step 3: Create the docker and enroot images in the CC using the dockerfile



Step 4: Copy the enroot image on the shared storage



Step 1 - Server on the CC - work on the terminal

- De-facto operation is no Graphical Interface on the provided images
- Accessing instances via ssh
  - No login based on password by default (public and private keys!)
- OpenStack must be aware of your public key(s) to add it(them) to newly created instances (otherwise you will not be able to login)
  - You can import a public key of a keypair generated using your method of preference
  - You can generate a keypair using OpenStack
    - the private key will be downloaded to your computer
    - the public will be recorded by OpenStack

Step 1 - Server on the CC - creating a KeyPair

- Default login onto servers is based on public-private key infrastructure
- Two options
  - To create a new pair of public and private keys ⟵ Used in this course
  - To upload the public key part of existing pair

Live demo

image: Freepik.com

Step 1 - Server on the CC - creating a KeyPair

Too fast? Check a video example here

https://tinyurl.com/kxhxpdaf

*For this hands-on only the creation of the key is relevant (you can ignore connecting to the server for now.)

Step 1 - Server on the CC - creating the server

- Create a server running Ubuntu 22.04

Live demo

image: Freepik.com

Step 1 - Server on the CC - creating the Server

Too fast? Check a video example here

https://tinyurl.com/yckp8ym4

Step 1 - Server on the CC - accessing the server

- Default IPs are on the private OpenStack network
- Floating IPs allow bridging external networks to the private OpenStack network

Live demo

image: Freepik.com

Step 1 - Server on the CC - creating and assigning a floating ip

Too fast? Check a video example here

https://tinyurl.com/bdfha7sz

Step 1 - Server on the CC - creating and assigning security groups

- By default all ports are closed by the firewall of OpenStack
- Security groups is the mechanism provided for opening these firewalls
- This demo shows how to open the port 22, required by ssh

Live demo

image: Freepik.com

Step 1 - Server on the CC - creating and assigning security groups

Too fast? Check a video example here

https://tinyurl.com/2s3f6vny

Step 1 - Server on the CC - connecting to the created server

- This demo shows how to connect via ssh to the server created before

Live demo

image: Freepik.com

Step 1 - Server on the CC - connecting to the created server

Too fast? Check a video example here

Linux and MacOS

https://tinyurl.com/kxhxpdaf

Windows using PuTTy

https://tinyurl.com/mt9cxzpn

*For windows users connecting via the Windows Subsystem for Linux (WSL) (e.g., Ubuntu app in windows):
You need to copy the downloaded file from Windows into the WSL space. If your file has been downloaded into C:/Downloads/pair.pem from the Ubuntu terminal copy the file with:
$ cp /mnt/c/Downloads/pair.pem ./key.pem

After that follow the Linux and MacOS video for connecting, assuming key.pem is your key (i.e., change the permissions and use this file in the ssh command).

Step 1 - Server on the CC - summary of steps

- On the Compute Cloud Web Interface
  - Generate a new keypair (see videos before)
- On your computer (see video links the slide before)
  - A file with the extension .pem will be downloaded to your machine (the private key) from previous step
    - **In Linux/UNIX:** change the permission of that file to 600 ($ chmod 600 …)
    - **In Windows with WLS:** copy the downloaded file to inside the WSL (/mnt/c/ allows you accessing C:\ in windows from WSL,)
      - **Once copied, change the permissions as in the Linux/UNIX case**
    - **In Windows with Putty:**
      - import it using PuttyGen and configure Putty for using the imported key

Step 1 - Server on the CC - summary of steps

- On the Compute Cloud Web Interface
  - Create an instance (next steps are documented with screen shots in successive slides)
    - Choose Ubuntu as image
    - CPU only flavor (preferably a small one)
    - Should be accessible from Internet
      - Place the instance on the private network called internet
      - Once the instance is created assign it a floating IP from the Internet pool

Step 1 - Server on the CC - summary of steps

Subtask: Access the created instance via SSH

- On the Computer Cloud Web Interface
  - **Remember you should have created a security group that allow ingress connections to port 22! and added it to the instance**
- On your computer
  - Open a terminal application

```
ssh -i <path_to_the_pem_file> ubuntu@<floating-ip>
```

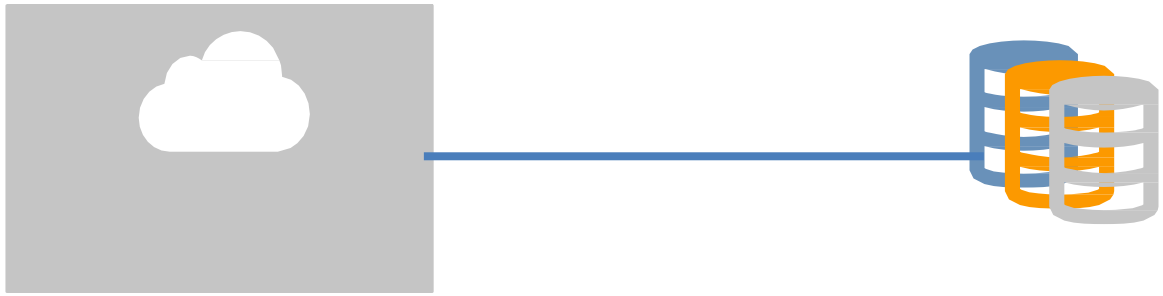
After this step, the rest of slides assume everyone is connected via ssh to the created instance

Step 2: Shared Storage CC and LAI

This use case shows the integration among different LRZ Services

Examples where such integration would be needed

1. Pre- or Postprocessing of data to use/used in SMUC-NG, Linux Cluster or LRZ AI Systems
2. Downloading a large dataset to be consumed on another LRZ service
3. Others (like our example: use that storage to shared an enroot image with the LRZ AI System)



compute cloud instance

DSS Storage

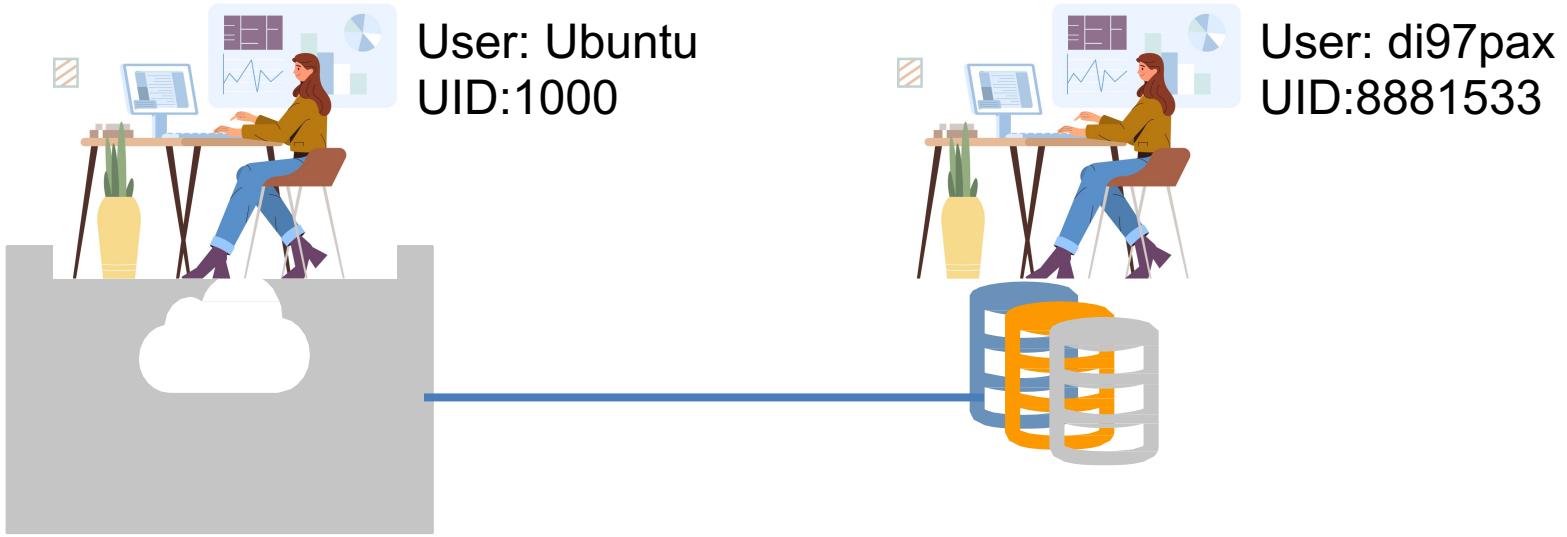
Step 2: Shared Storage CC and LAI

Different users

On a VM: a user within the OS running in that VM

On DSS: LRZ user

Access to DSS containers and data defined on UID and GID basis



Step 2: Shared Storage CC and LAI

1st. create in the VM a Unix group with the group of the container (usually the container name)

```
$ sudo groupadd --gid <GID> <container-name>
```

2nd. create in the VM a user with the same UID that your LRZ account and assign it to the created group in 1st

```
$ sudo adduser --home /home/myuser --uid <UID> --ingroup <container-name> myuser
```

3rd. install the nfs-common package (to mount nfs shares)

```
$ sudo apt install nfs-common
```

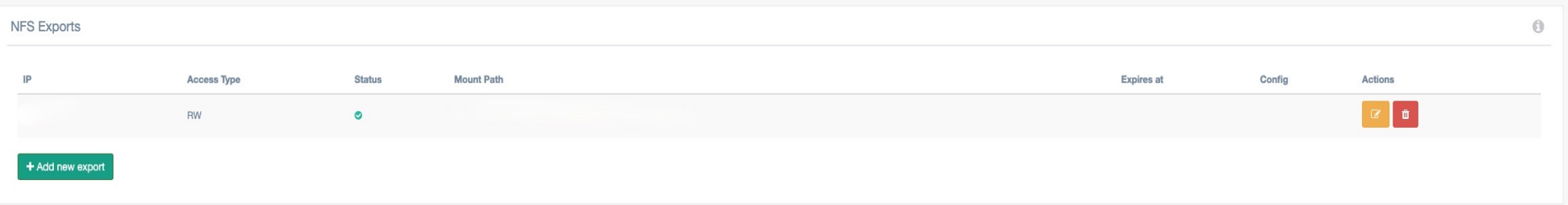
4th. create a folder where the DSS container will be mounted

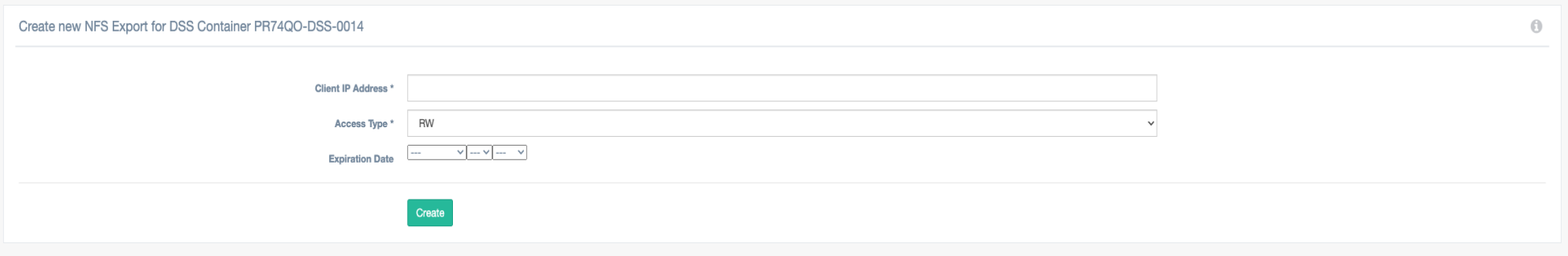
```
$ sudo mkdir -p <container-name>
```

Steps to do in your

Step 2: Shared Storage CC and LAI

5th. create a new NFS export of your DSS container
(only *data curators* can perform this step)





Steps to do via DS
or dsscli

Step 2: Shared Storage CC and LAI

6th. mount the container

```
$ sudo mount -t nfs -o rsize=1048576,wsize=1048576,hard,tcp,bg,timeo=600,vers=3 <mount-path>
/dss/<container-name>
```

To be obtained via the DSSWeb or dsscli

7th. access the data on your DSS container

```
$ sudo su - myuser
$ cd /dss/<contianer-name>
```

Steps to do in your

Step 2: Shared Storage CC and LAI

- This demo shows how to mount a DSS container

Live demo

image: Freepik.com

Step 3: Create the docker and enroot images in the CC using the dockerfile: installing the tools

1st. Install Docker on the Server

```
$ sudo apt install docker.io
$ sudo usermod -aG docker ubuntu
```

Exit and login again for this step to be effe

Alternative way of installing docker:

curl -fsSL https://get.docker.com/
 | bash

2nd. Install enroot

```
$ arch=$(dpkg --print-architecture)
$ curl -fSsL -O https://github.com/NVIDIA/enroot/releases/download/v3.4.1/enroot_3.4.1-1_${arch}.deb
$ curl -fSsL -O https://github.com/NVIDIA/enroot/releases/download/v3.4.1/enroot+caps_3.4.1-1_${arch}.deb
$ sudo apt install -y ./*.deb
```

Step 3: Create the docker and enroot images in the CC using the dockerfile: exporting the container

3th. Create the Docker image

```
$ docker build . -t test
```

```
from ubuntu:23.04
RUN apt update
RUN apt install -y python3
```

4th. Check that the image is there

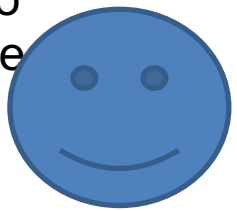
```
$ docker image list
REPOSITORY   TAG      IMAGE ID      CREATED        SIZE
test         latest   0bce02617813  6 minutes ago  158MB
ubuntu       23.04    639282825872  7 weeks ago    70.3MB
```

You use **dockerd** as want to import the image from the local docker daemon

5th. Export the image to an enroot one

```
$ enroot import dockerd://test
$ ls
test.sqsh Dockerfile
```

You can copy now test.sqsh to the mounted DSS and it will be available to use in the LRZ AI System



Step 3: Create the docker and enroot images in the CC using the dockerfile

Live demo

image: Freepik.com

Other ways of interacting with the LRZ Compute Cloud.

The OpenStack Client Tools

- Command-line client for OpenStack that brings the command set for Compute, Identity, Image, Object Storage and Block Storage APIs together in a single shell with a uniform command structure
- https://docs.openstack.org/python-openstackclient
- There are different ways of installing the tools. Some examples are:
  - In Ubuntu using apt as front end package manager

```
$ sudo apt install python3-openstackclient
```

  - Using python pip

```
$ pip3 install openstackclient
```

Hands-on: configuring the openstack client tools

- A configuration file specific to your account can be downloaded for configuring your system to access the compute cloud via the command line

Live demo

image: Freepik.com

Hands-on: configuring the openstack client tools

Too fast? Check a video example here

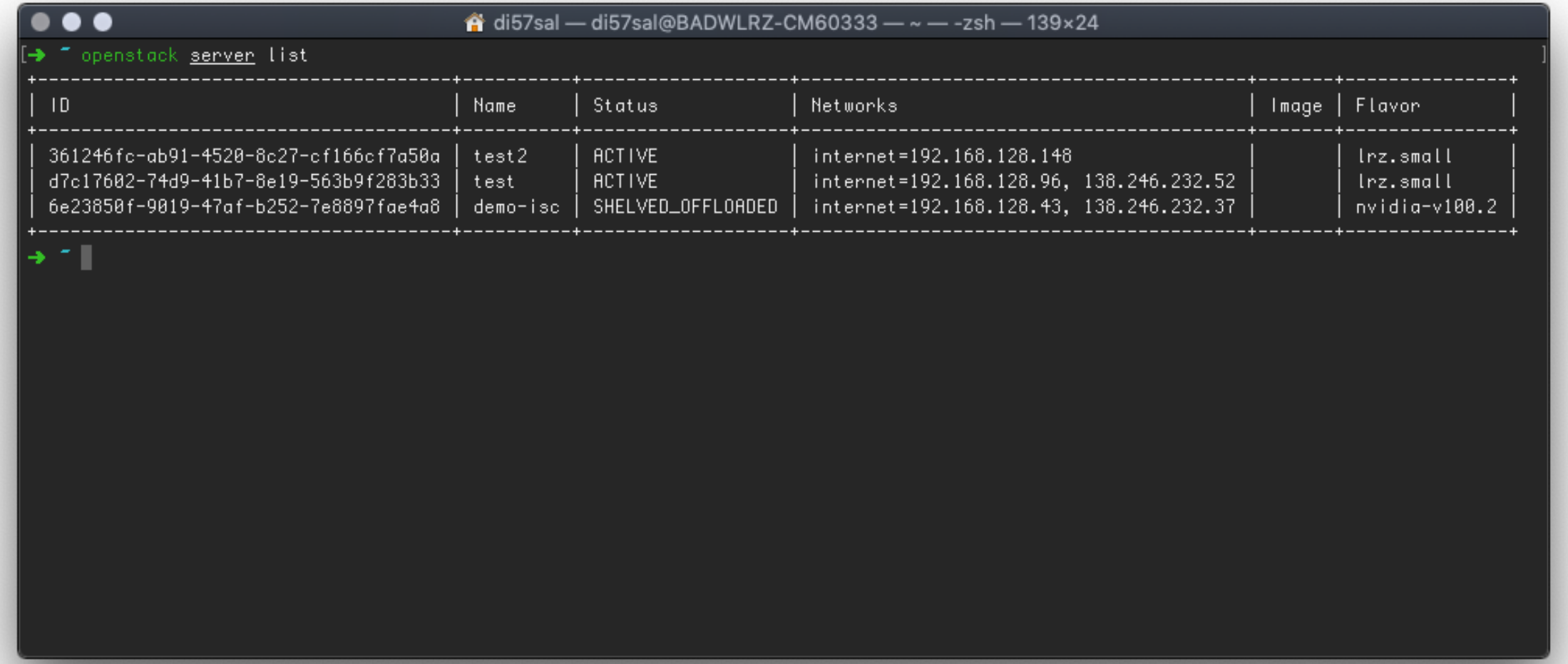
https://tinyurl.com/bdz42kaa

The OpenStack Client Tools

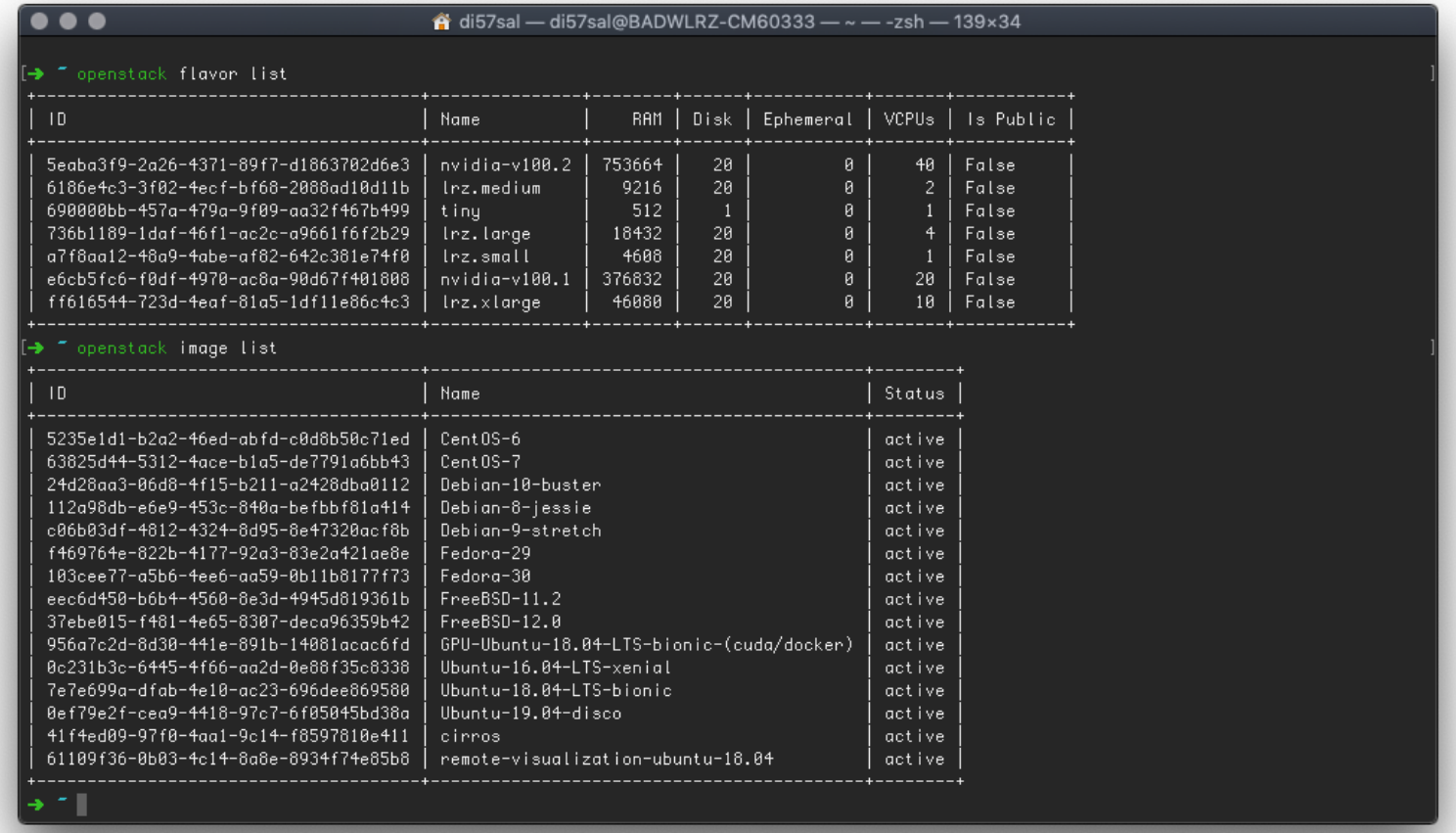
- Execute the downloaded file
  - e.g. in my case

```
$  source /Download/di57sal-openrc.sh
```

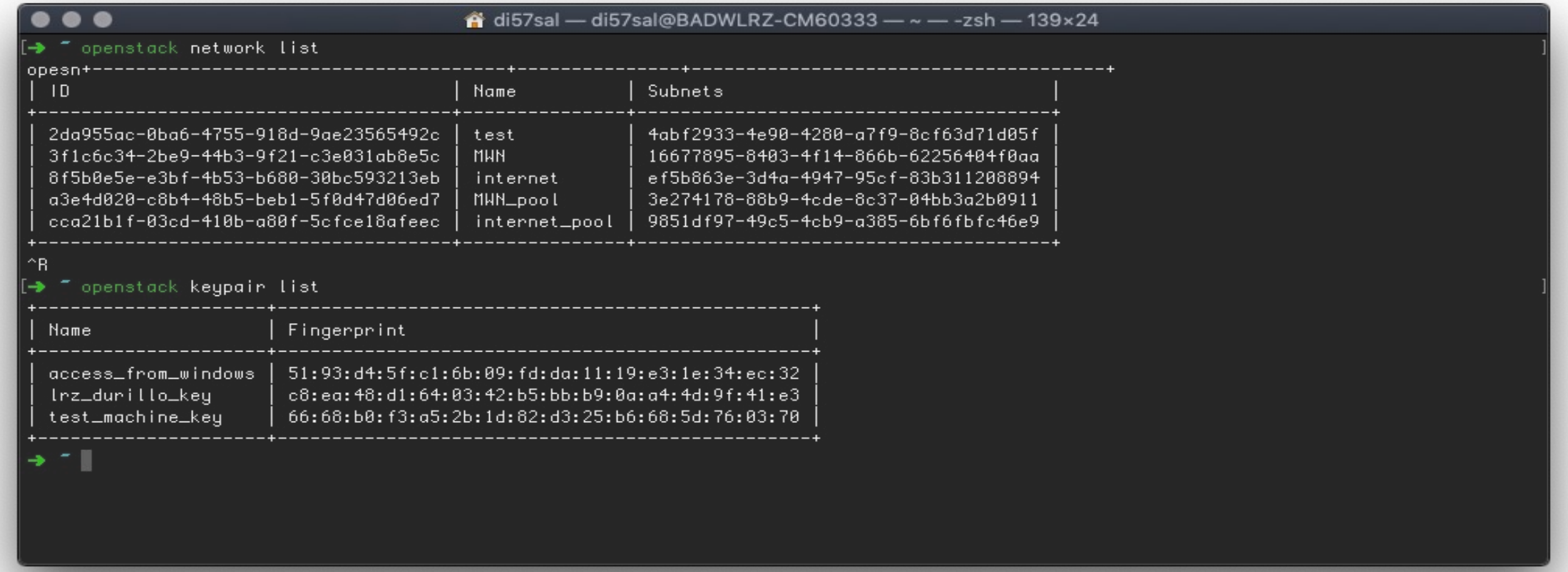
The OpenStack Client Tools - Listing



The OpenStack Client Tools - Listing



The OpenStack Client Tools - Listing

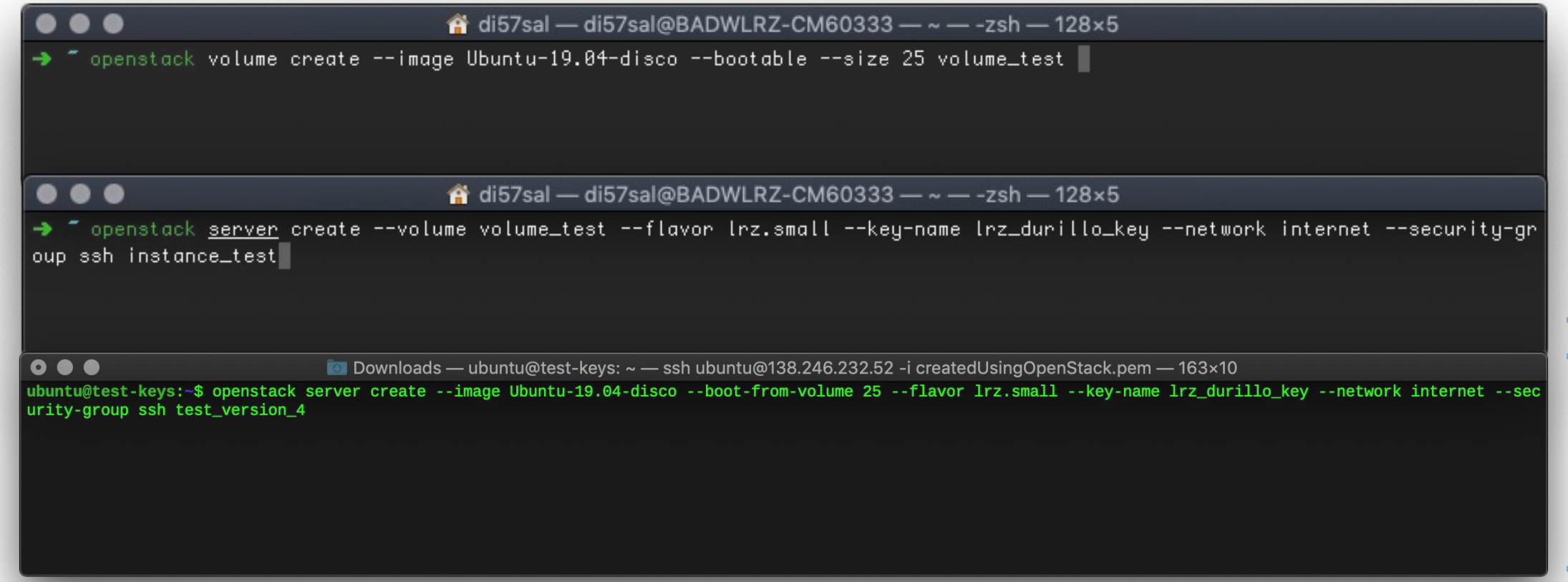


The OpenStack Client Tools - Creating

- Simple unified API for all OpenStack Entities
- Access to the help of each command
  - Simply executing it with no additional arguments
  - Executing it with –h | -- help option

The OpenStack Client Tools – Creating a new server

- Depending on the version of the tools



Version 3.19.0

Version > 4.0.0

openstack server create --image Ubuntu-22.04-jammy --boot-from-volume 25 --flavor lrz.small --key-name \
yourkeyname --network MWN --security-group your_security_group_name instance_name

Version 6.0.0

Hands on Session 2

- Recreate the instance of Hands-on session before using the Openstack Command Line tools

Live demo

image: Freepik.com

Hands-on 2

Too fast? Check a video example here

https://tinyurl.com/6bcepbc5

Hands-on session—-Secure your AI Workflow with LRZ middleware

Purpose: your workflow includes sensitive info

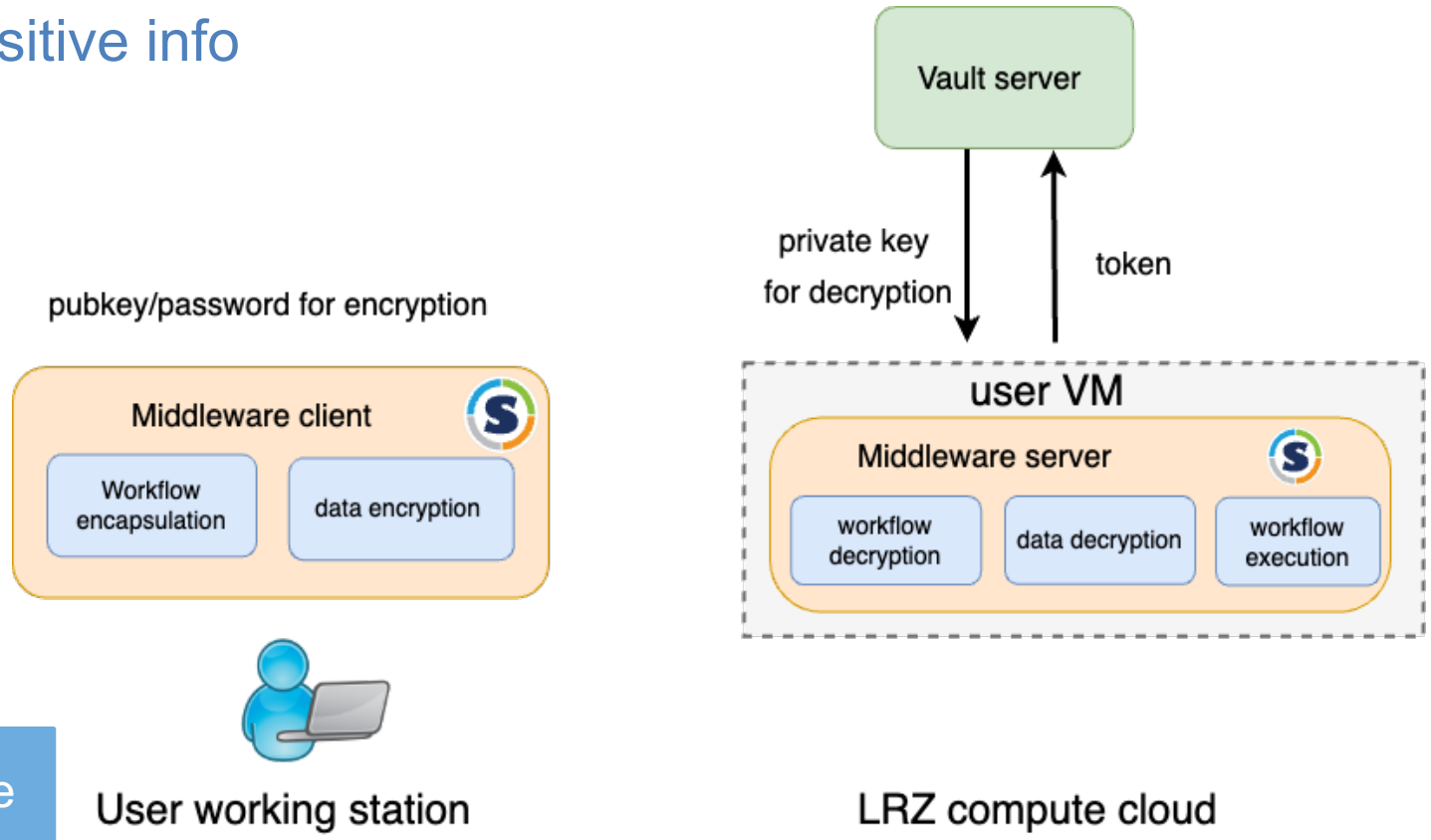
What does this workflow do?

1. A KNN model
2. Read patient heath data from a csv file
3. Display data property and plot a correlation map

Hands-on session—-Secure your AI Workflow with LRZ middleware*

Purpose: your workflow includes sensitive info

- Pack your code inside encrypted **workflow container**
- Pack your data (if large size, e.g. 100GB) into an encrypted **data container**



the main components of LRZ middleware

This hands-on session uses both Middleware client and Middleware server on VM

* N. Zhou, F. Dufour, V. Bode, P. Zinterhof, N. J. Hammer, and D. Kranzlmller, “Towards Confidential Computing: A Secure Cloud Architecture for Big Data Analytics and AI,” in *IEEE International Conference on Cloud Computing (IEEE CLOUD)*, (Chicago, Illinois, USAI), 2023.

Hands-on session—-Secure your Workflow with LRZ middleware

Steps:

1. Prepare the customized image for uploading to Openstack: e.g. *ibguestfs, packer*
2. Upload an customized VM image to Openstack (an image has been uploaded for you)
3. Create an VM with the uploaded image
4. Create an encrypted data container with password
5. Create an encrypted workflow container with a given pubkey
6. Execute the encrypted workflow with a given token

A token is used to fetch a private key stored safely on the Vault server. A private key is used to decrypt our workflow container. The token is NOT the decryption key for the workflow container

Hands-on session—-Secure your Workflow with LRZ middleware

Steps:

1. Prepare the customized image for uploading to Openstack:

$ qemu-img convert -f qcow2 -O raw jammy-server-cloudimg-amd64.img aicourse.raw (optional, on your computer)

2. Upload an customized VM image to Openstack (on your computer)

$ openstack image create --container-format bare --disk-format raw --file aicourse.raw AI-course (optional)

$ openstack image set --accept [image_id] (cml for today, replace image_id with the id in the course)

3. Create an VM with the uploaded image (on your computer)

$openstack server create --image AI-course --flavor lrz.xlarge --network MWN --security-group ai_course --key-name aicourse --boot-from-volume 20 aicourse

4. create an encrypted data container with a password

$cd ~/middleware ; $ lrzclient encrypt -data mycontainer.data

5. create an encrypted workflow container with a given pubkey (located at lrz-middleware/rsa_pub.pem)

$ lrzclient encrypt -workflow myworkflow.sif

6. execute the encrypted workflow with a given token (located at lrz-middleware/token)

$ lrzserver run -workflow myworkflow.sif -data mycontainer.data

Hands-on session—-Secure your Workflow with LRZ middleware

Other cmls to play with

1. Decrypt your data container:

$ lrzserver unencrypt -data mycontainer.data (optional)

2. Seal your decrypted data container

$ lrzserver seal (optional)

Wrap-up

- Motivation of Cloud Computing?
- Introduction to Cloud Computing and OpenStack
- Hands-on session using the web interface
- Hands-on session using the OpenStack Client Tools
- Hands-on using custermised VM image
- Hands-on to secure AI workflow with LRZ middleware

Further reading:

1. Thomas, E., Zaigham, M. and Ricardo, P., 2013. *Cloud Computing Concepts, Technology & Architecture*. Prentice Hall.
2. Sefraoui, O., Aissaoui, M. and Eleuldj, M., 2012. OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications*, *55*(3), pp.38-42.
3. Netto, M.A., Calheiros, R.N., Rodrigues, E.R., Cunha, R.L. and Buyya, R., 2018. HPC cloud for scientific and business applications: taxonomy, vision, and research challenges. *ACM Computing Surveys (CSUR)*, *51*(1), pp.1-29.

Course Evaluation – Module 6: Introduction to the LRZ Compute Cloud

Please visit
url-for-dec23
and rate this course.

Your feedback is highly appreciated!
Thank you!