




Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

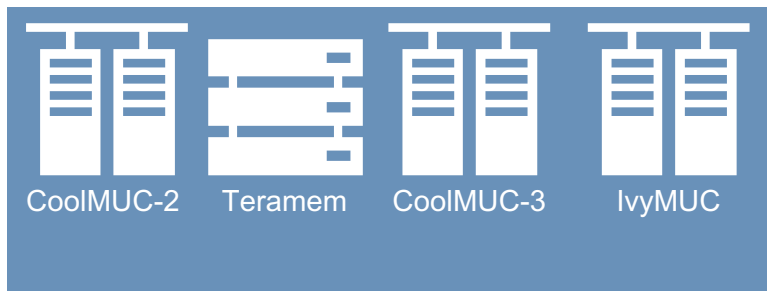
The background of the slide is a photograph of a modern, multi-story building with a glass and metal facade, likely the LRZ building. The image is overlaid with a semi-transparent blue filter. In the foreground, there are some trees and a street with a few people walking.

Deep Learning on the LRZ AI Systems

An High Level Overview of Some LRZ Resources



DSS
(Data Science Storage)

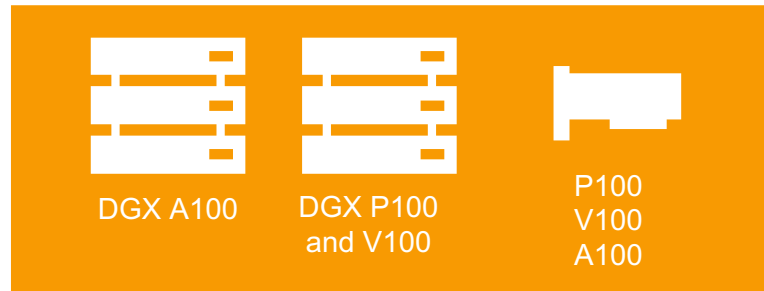


lxlogin8.lrz.de

lxlogin[1-4].lrz.de

lxlogin10.lrz.de

<https://www.rstudio.lrz.de>



LRZ AI Systems



Compute Cloud

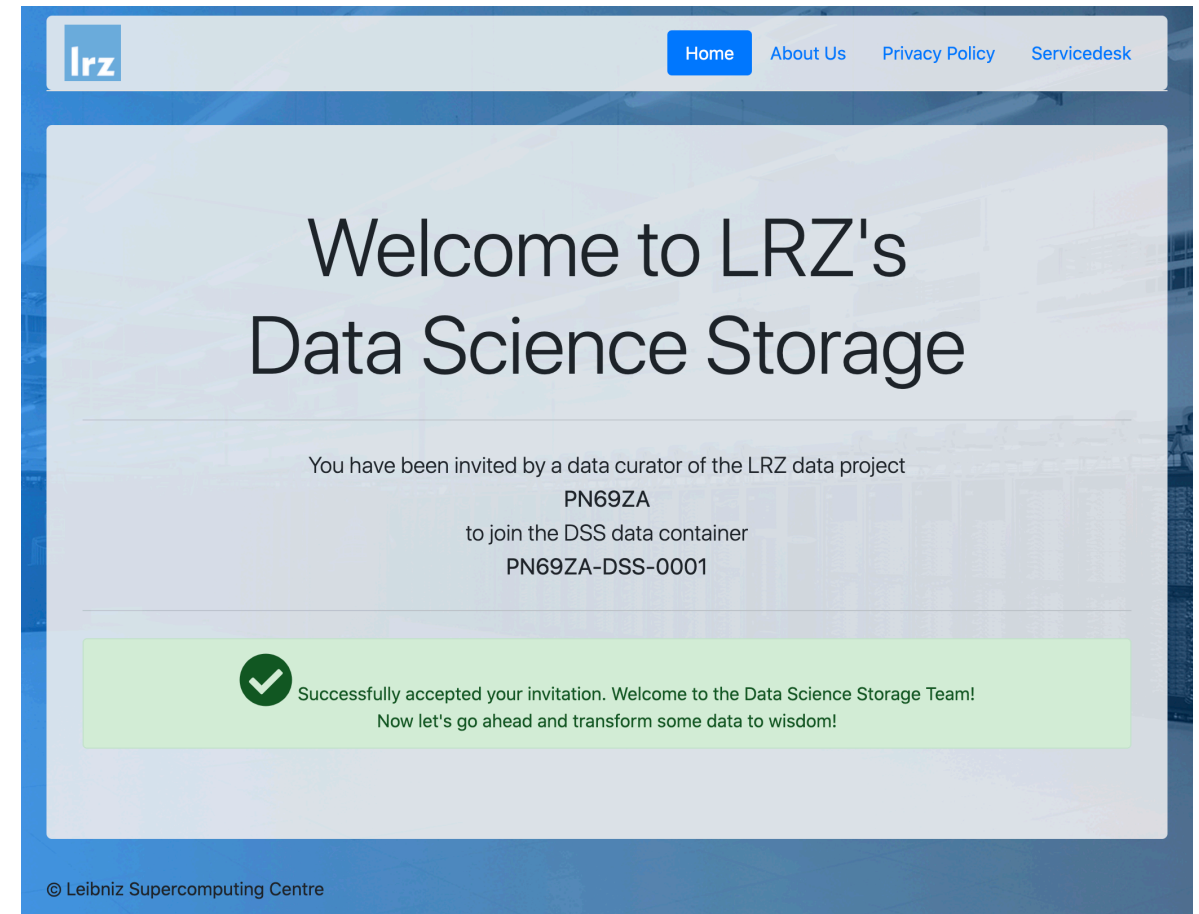


AI Ready

The LRZ AI System

Who can access?

- User requirements to get the access:
 1. Own a Linux Cluster account,
 2. Send a request through a service request ticket explaining the intended use.
- Upon approval, you will be invited to a DSS container - You need to accept this invitation before being able to access the AI systems!
- This DSS container will be used as your \$HOME (although this is going to change in the future.)



Resources Overview

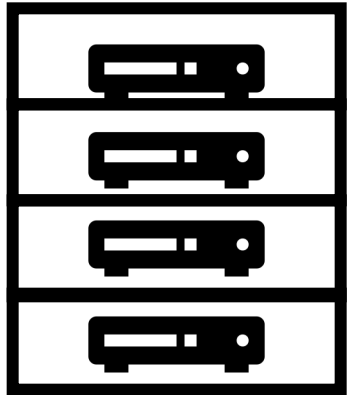
	Slurm Partition	Number of nodes	CPUs per node	Memory per node	GPUs per node	Memory per GPU
DGX A100 Architecture	lrz-dgx-a100-80x8	4	256	2 TB	8 NVIDIA A100	80 GB
	lrz-dgx-a100-40x8	1	256	1 TB	8 NVIDIA A100	40 GB
DGX-1 V100 Architecture	lrz-dgx-1-v100x8	1	80	512 GB	8 NVIDIA Tesla V100	16 GB
DGX-1 P100 Architecture	lrz-dgx-1-p100x8	1	80	512 GB	8 NVIDIA Tesla P100	16 GB
HPE Intel Skylake + NVIDIA Node	lrz-hpe-p100x4	1	64	256 GB	4 NVIDIA Tesla P100	16 GB
V100 GPU Nodes	lrz-v100x2 (default)	4	20	368 GB	2 NVIDIA Tesla V100	16 GB

Using the Cluster

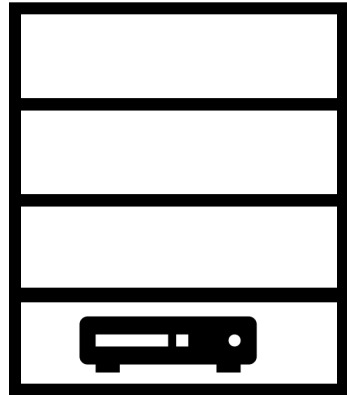
- SLURM: Simple Linux Utility for Resource Management
 - open source
 - fault-tolerant
 - highly scalable
- Cluster management and job scheduling
- (Three) main tasks
 - allocates exclusive and/or non-exclusive access to resources (compute nodes)
 - provides a framework for starting, executing, and monitoring work on the allocated nodes
 - arbitrates contention for resources by managing a queue of pending work.



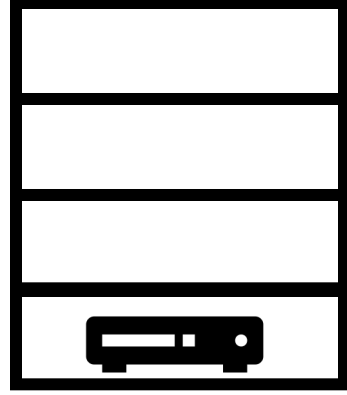
LRZ AI System Configuration



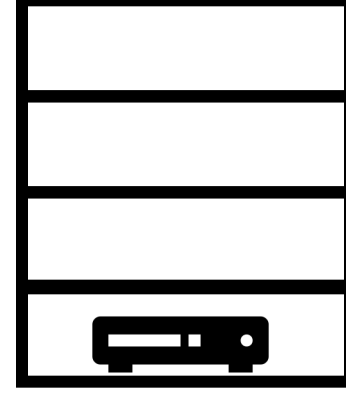
lrz-dgx-a100-80x8



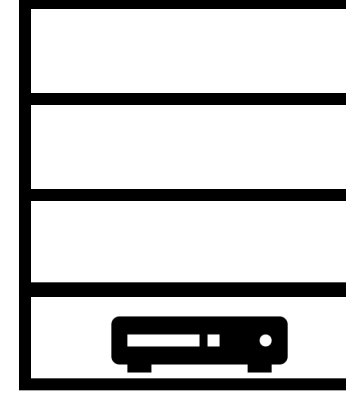
lrz-dgx-a100-40x8



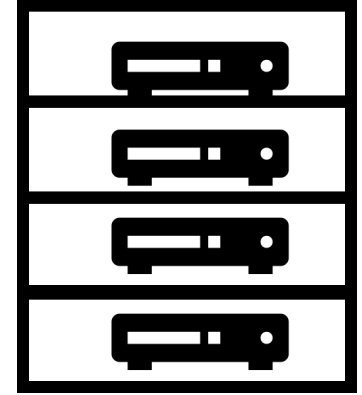
lrz-dgx-1-v100x8



lrz-dgx-1-v100x8



lrz-hpe-p100x4



lrz-v100x2



datalab2.srv.lrz.de



datalab3.srv.lrz.de



Private Network



MWNet

Accessing the LRZ System

- Login node datalab2.srv.lrz.de accessible via ssh

```
ssh -Y datalab2.srv.lrz.de -l xxyyyzz
```

- From the login node, jobs are submitted to the hardware described at the beginning of this course using SLURM
- A couple of handy SLURM commands

```
$ squeue
```

```
$ sinfo
```

```
$ salloc /  
scancel
```

```
$ srun
```

Allocating and Starting Jobs Interactively

- Get resources allocated

```
$ salloc -p dgx --ntasks=8 --gres=gpu:8
```

resource queue

start n tasks in the machine per node (one per GPU required by some software framework)

Indicate that access to 8 GPU resources is required

- Submit start job in the allocated resources

```
$ srun --pty bash
```

```
$ srun hostname
```

Allocating and Starting Jobs in Batch Mode

- Batch jobs are the preferred way of using the LRZ AI Systems.
- The **sbatch** command submits jobs described in a *sbatch script* file.
- Two additional arguments required in sbatch scripts: *output* and *error messages* file.
- After the preamble, the job to be executed is described.

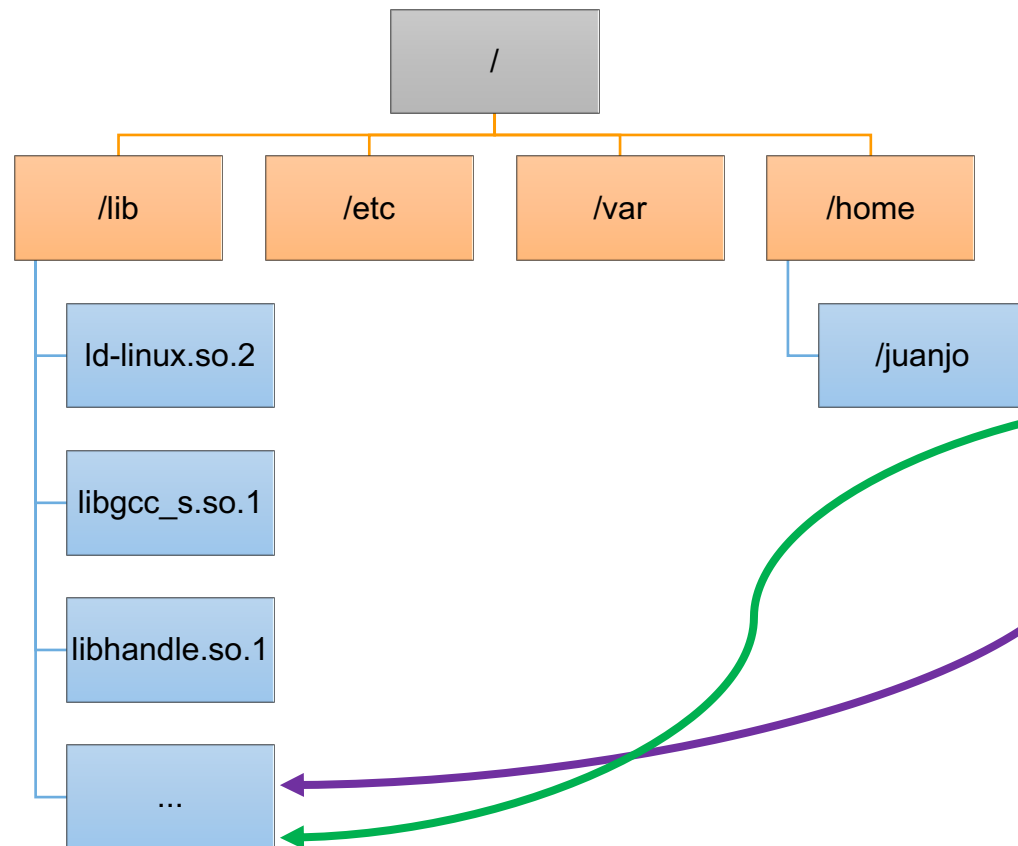
```
#!/bin/bash
#SBATCH -N 1
#SBATCH -p dgx-1-p100
#SBATCH --gres=gpu:2
#SBATCH --ntasks=2

srun nvidia-smi
```

```
$ sbatch test.sbatch
```

User Defined Software Stack: Container Technologies

Typical Linux File System

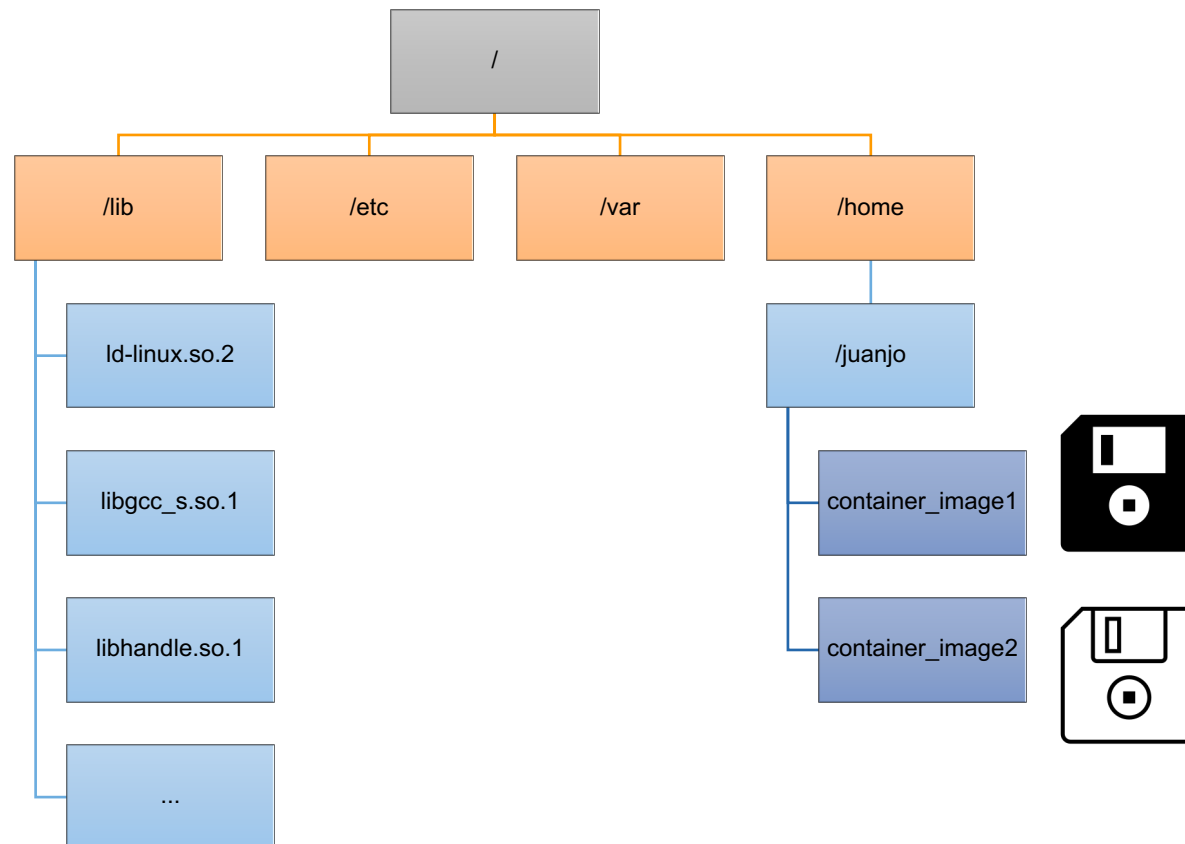


Program In Execution in Linux

```
$ ldd /bin/program
linux-vdso.so.1 (0x00007fff204e8000)
libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007f411cc6a000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f411ca78000)
libpcre2-8.so.0 => /usr/lib/x86_64-linux-gnu/libpcre2-8.so.0 (0x00007f411c9e8000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f411c9e2000)
/lib64/ld-linux-x86-64.so.2 (0x00007f411ccd0000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f411c9bf000)
```

User Defined Software Stack: Container Images

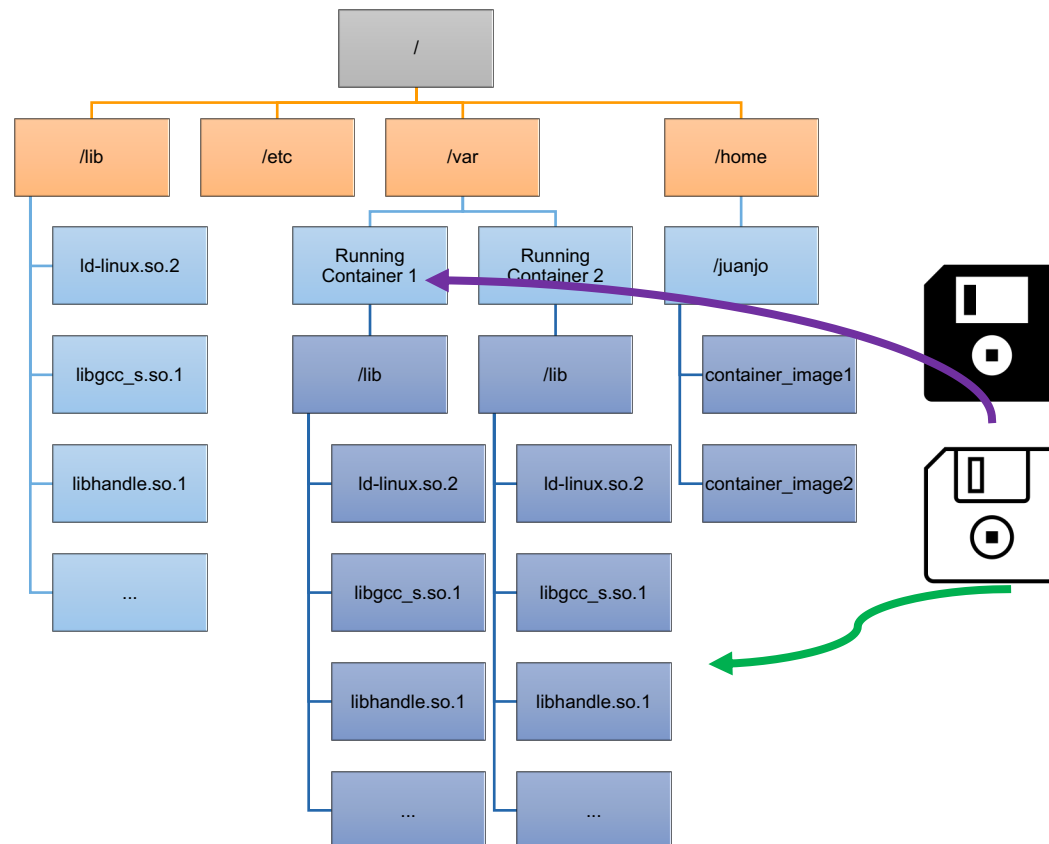
Typical Linux File System



- Typically, a single compressed file
 - It contains a complete Linux File System + Metadata
- Different container technologies might:
 - use different formats
 - e.g., OCI format is a **specification for container images based on the Docker Image Manifest Version 2, Schema 2 format**
 - hide images to users
- Are meant to be static
- Not to be confused with a docker file

User Defined Software Stack: Container

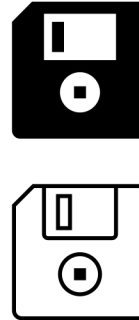
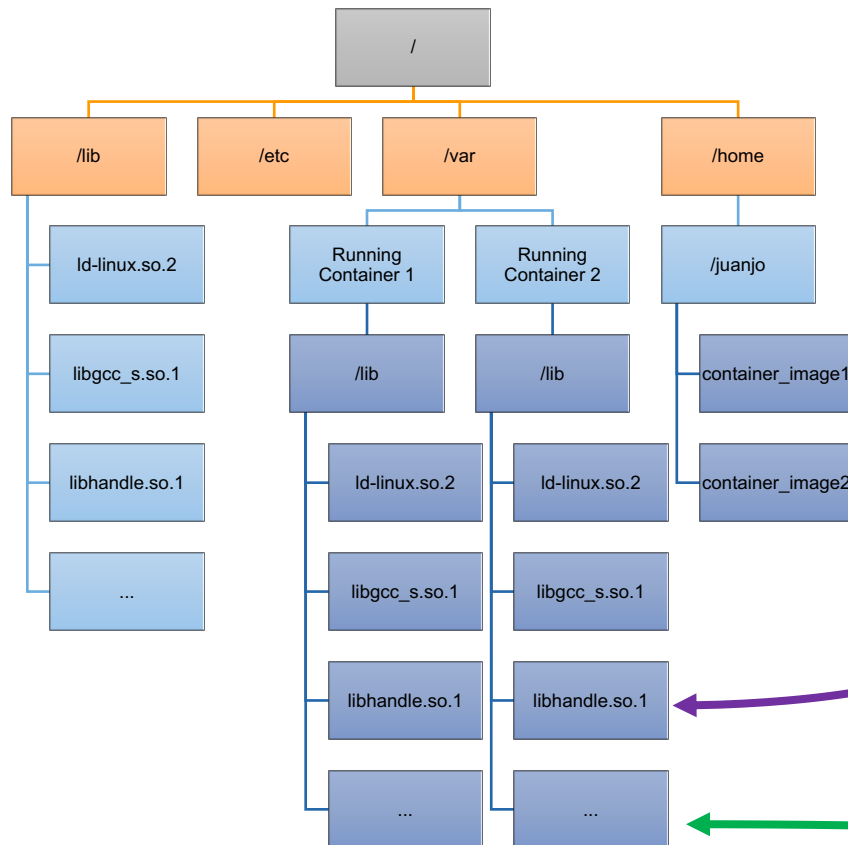
Typical Linux File System



- A running instance of a container image
 - A complete Linux File System within a Linux File System
 - Libraries might be different (versions)
 - Provided programs might be different
- Specific program in charge of unpacking the image and storing it within the proper folder
 - Docker, Podman, Enroot, etc.
- More than one container can
 - Exist at any point in time
 - Be generated from a single image

Understanding Containers: Container

Typical Linux File System



- It is possible to “run a process within a container”
 - Confine the process to the content of the container File System
 - Specific program in charge for confining and running the process within the container
 - docker run/start, enroot start

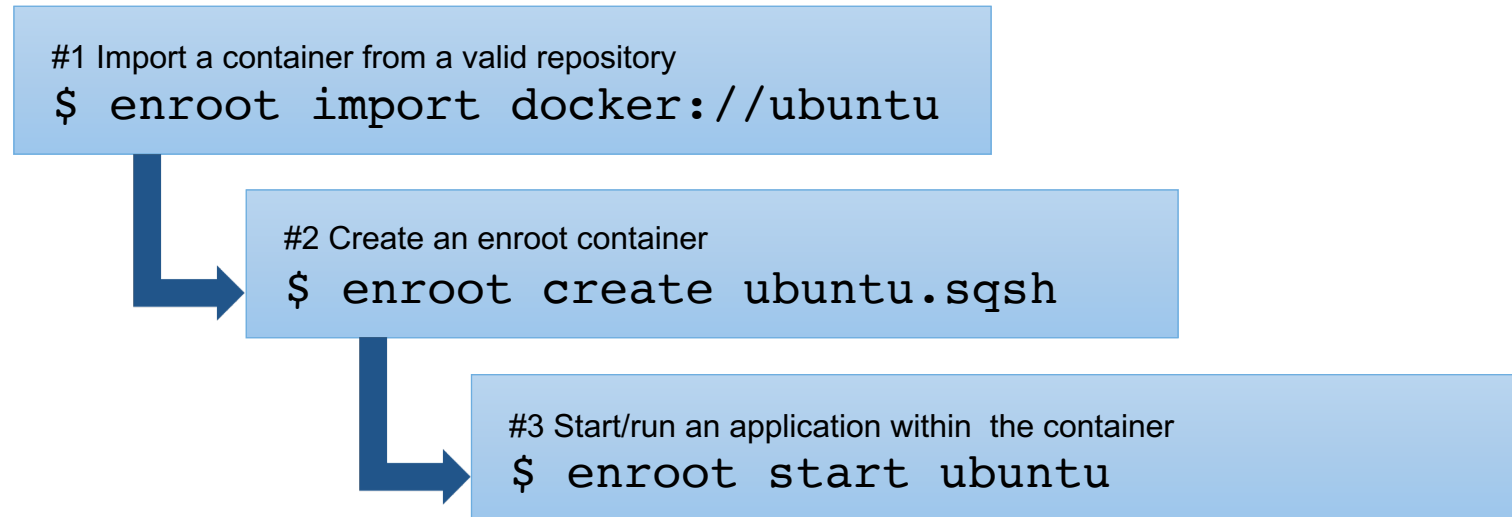
```

$ ldd /bin/program
linux-vdso.so.1 (0x00007fff204e8000)
libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1
(0x00007f411cc6a000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6
(0x00007f411ca78000)
libpcre2-8.so.0 => /usr/lib/x86_64-linux-gnu/libpcre2-8.so.0
(0x00007f411c9e8000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2
(0x00007f411c9e2000)
/lib64/ld-linux-x86-64.so.2 (0x00007f411ccd0000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0
(0x00007f411c9bf000)
    
```

Process run within Running Container 2

The Enroot Container Technology

- Containerized applications with `enroot`, a rootless container runtime by Nvidia
- Slightly different workflow than with `Docker`



- It should be noticed that the workflow in the AI System consists in submitting jobs that run containerized within an `enroot` defined container

Nvidia NGC: Container Images Repository



Beyond Existing NGC: Creating Custom Enroot Image

- Get resources / get an allocation of resources
 - `salloc -p test-v100x2 -q testing --gres=gpu:1`
- Open a terminal on the allocated resources
 - `srun --pty bash`
- (Optional) Create a base container image (e.g., pulling from NGC)
 - `enroot import -o pytorch_base.sqsh docker://nvcr.io#nvidia/pytorch:22.06-py3`
- Create a container out of an image
 - `enroot create --name pytorch_container pytorch_base.sqsh`

Beyond Existing NGC: Creating Custom Enroot Image

- Start a terminal your recently created container
 - `enroot start pytorch_container bash`
- Modify your container accordingly
- Leave your container
- Export the created container as an image
 - `enroot export --output hugging_face.sqsh pytorch_container`

Using the Cluster with Enroot Containers

- Get resources allocated

```
$ salloc -p dgx --ntasks=8 --gres=gpu:8
```

← Indicate that access to the GPU resources is required

- Submit containerized job

```
$ srun --pty enroot start --mount ./data-test:/mnt/data-test ubuntu bash
```

mounting outside folders inside
the container

the container previously created

- Meet the pyxis plugin: container creating and job submission in a single step

```
$ srun --container-mounts=./data-test:/mnt/data-test --container-name=horovod  
--container-image='horovod/horovod+0.16.4-tf1.12.0-torch1.1.0-mxnet1.4.1-  
py3.5' bash
```

Open On Demand: Web Frontend for the LRZ AI System

- Interactive web service for AI systems where Jupyter Notebook, JupyterLab and RStudio Server environments are available, at <https://datalab3.srv.lrz.de>.
- Given that requested resources are available, the status of the session will change from "Queued" to "Starting" and finally "Running".
- <https://doku.lrz.de/display/PUBLIC/LRZ+AI+Systems>

The screenshot displays the 'LRZ AI Systems Web UI (TEST INSTANCE)' interface. The top navigation bar includes 'Files', 'Jobs', 'Clusters', and 'Interactive Apps'. The main content area is titled 'Jupyter Notebook version: 0d9944a' and 'Jupyter Notebook Access'. It features a sidebar with 'Interactive Apps' and 'Servers' sections, where 'Jupyter Notebook' is selected. The main form includes a dropdown for 'Choose the partition where the job will run' (set to 'gpu-v100'), a link to 'Check available partitions https://doku.lrz.de/x/sQCuAw', a dropdown for 'Choose an Nvidia NGC container image or "Custom" to provide the container info in the next field' (set to 'Pytorch'), a link to 'Check https://tinyurl.com/3uscc23c to configure your Nvidia NGC access', input fields for 'Number of hours' (1) and 'Desired number of GPUs for your job' (1), a text area for 'Comma separated list of mounts to perform from the host inside the container in the format <path-in-home><path-in-container>', a checkbox for 'Make it Jupyter Lab!', and a 'Launch' button.

The Compute Cloud

Cloud Computing Characteristics

A consumer can request and receive access to a service offering without an administrator or some sort of support staff having to fulfil the request manually.

on demand

Cloud services should be easy to access. Ideally only a basic network connection should be required.

broad
network
access

Ability to grow with user demand. If the system is well defined it should be relatively easy for the provider to add more

flexible

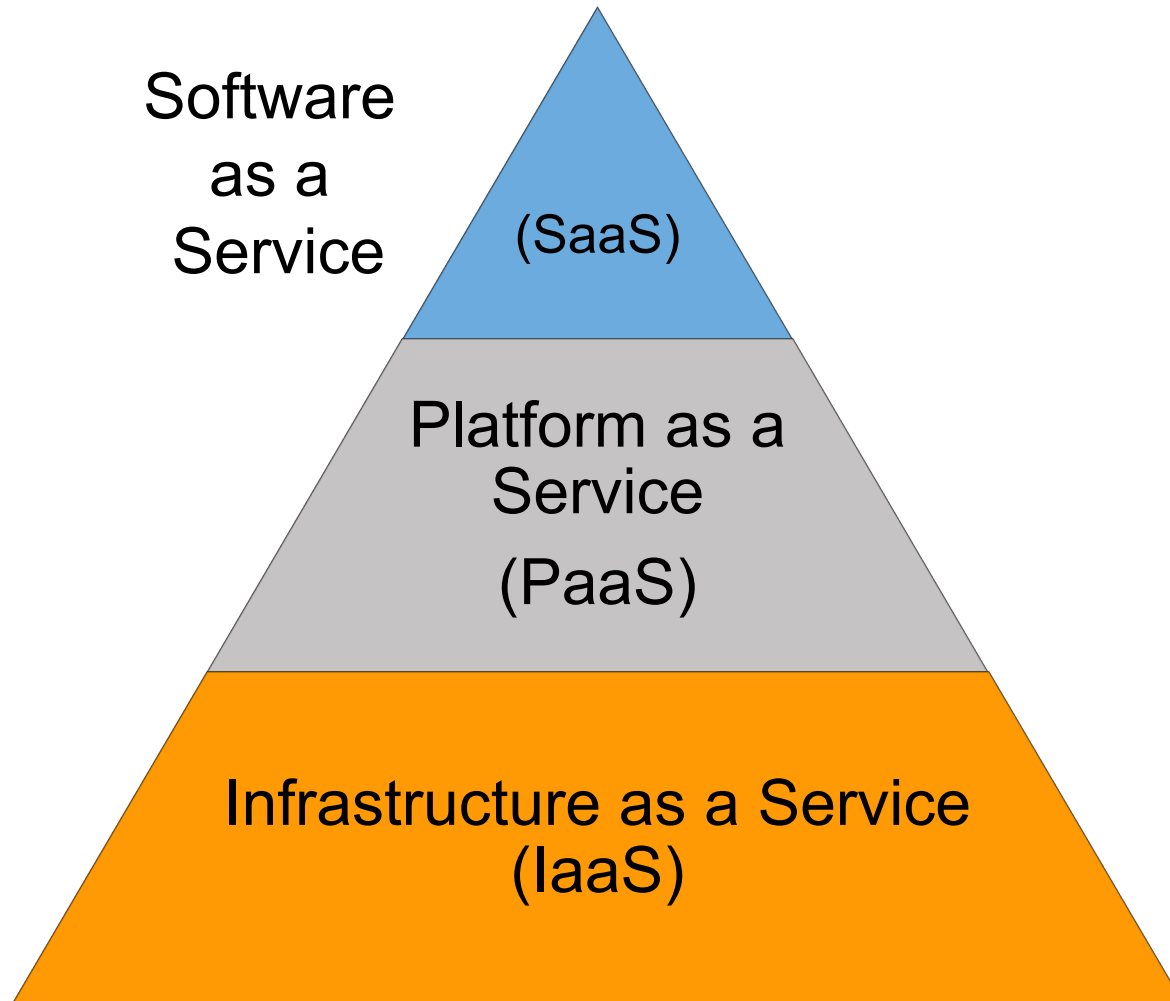
Cloud services must have the ability to measured usage. Usage can be quantified using various metrics, such as time, bandwidth used, and data used. This ability to measure allow what is known as pay as you go model.

measurable

A user will not need all the resources available to her. When resources are not used, they should be released, and other users can benefit of it or they can be simply not used (not consuming energy.)

resource
pooling

Cloud Services



- SaaS – Fully developed software solution to be used
 - e.g., Google Drive
- PaaS – Provides a framework on top of which is possible to build, deploy, and manage software products
 - e.g., Heroku
- IaaS - Provides a completely virtualized computing infrastructure provisioned and managed over the internet
 - e.g., LRZ Compute Cloud

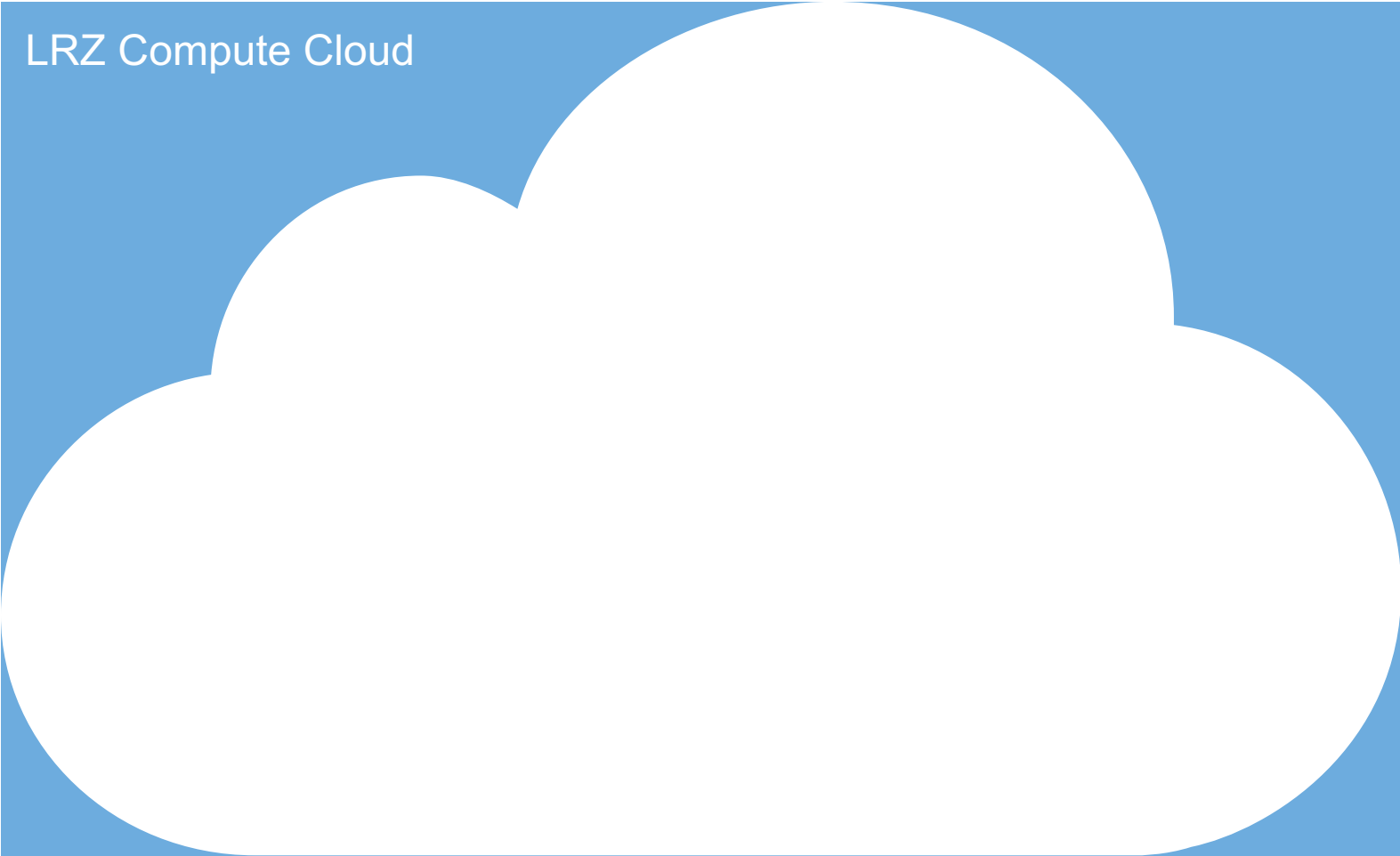
OpenStack: The Engine of the LRZ Compute Cloud

- 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

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

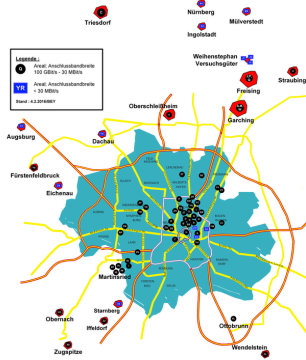
The LRZ Compute Cloud at a Glance



Internet



MWN



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.



FreeBSD



CentOS

OpenStack - Terminology

- Instance

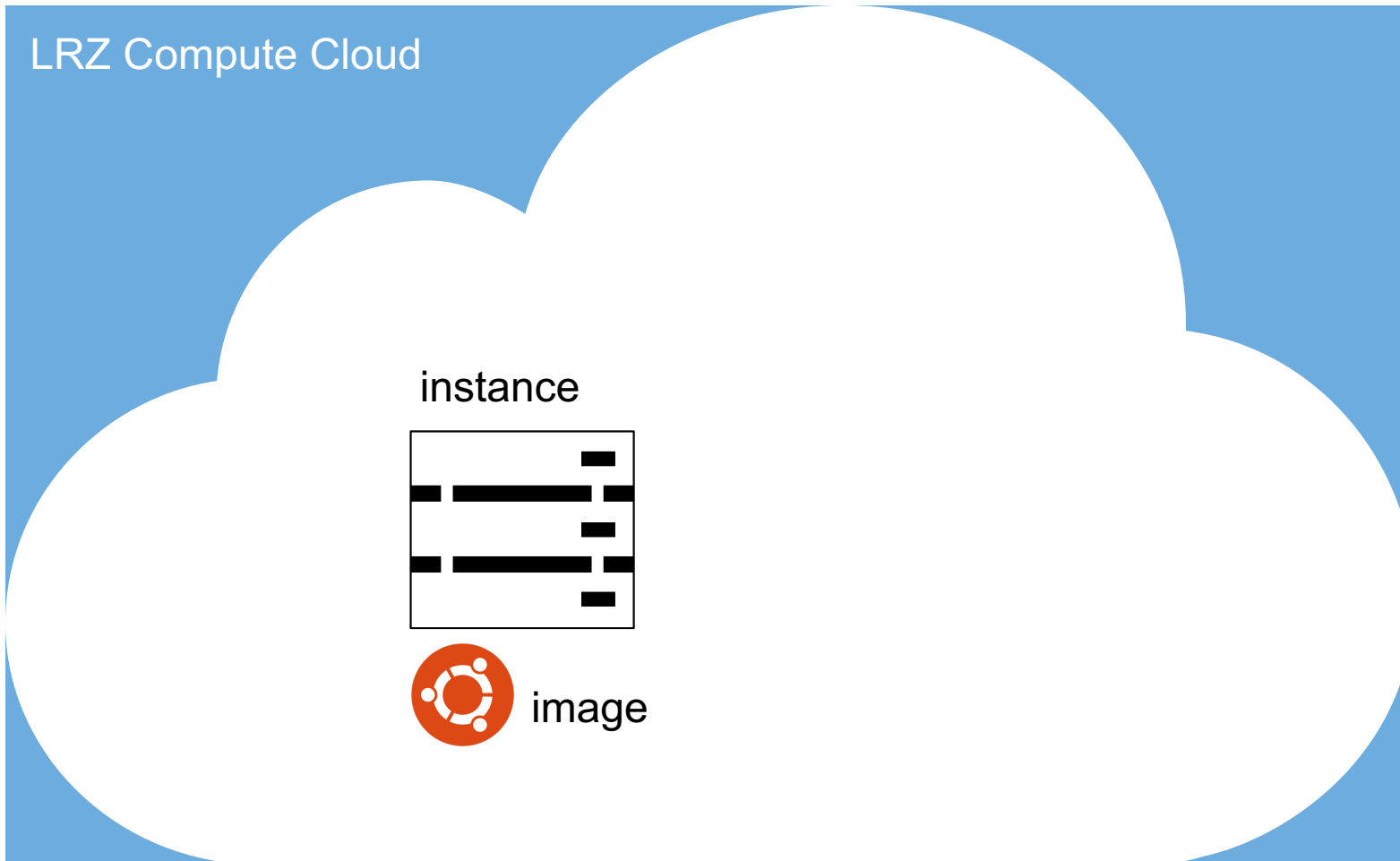
A copy of an image running as a virtual server the cloud. We will also call it server.

- 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

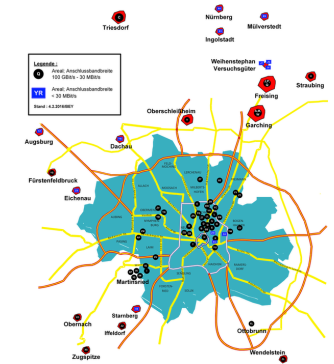
The LRZ Compute Cloud at a Glance



Internet



MWN

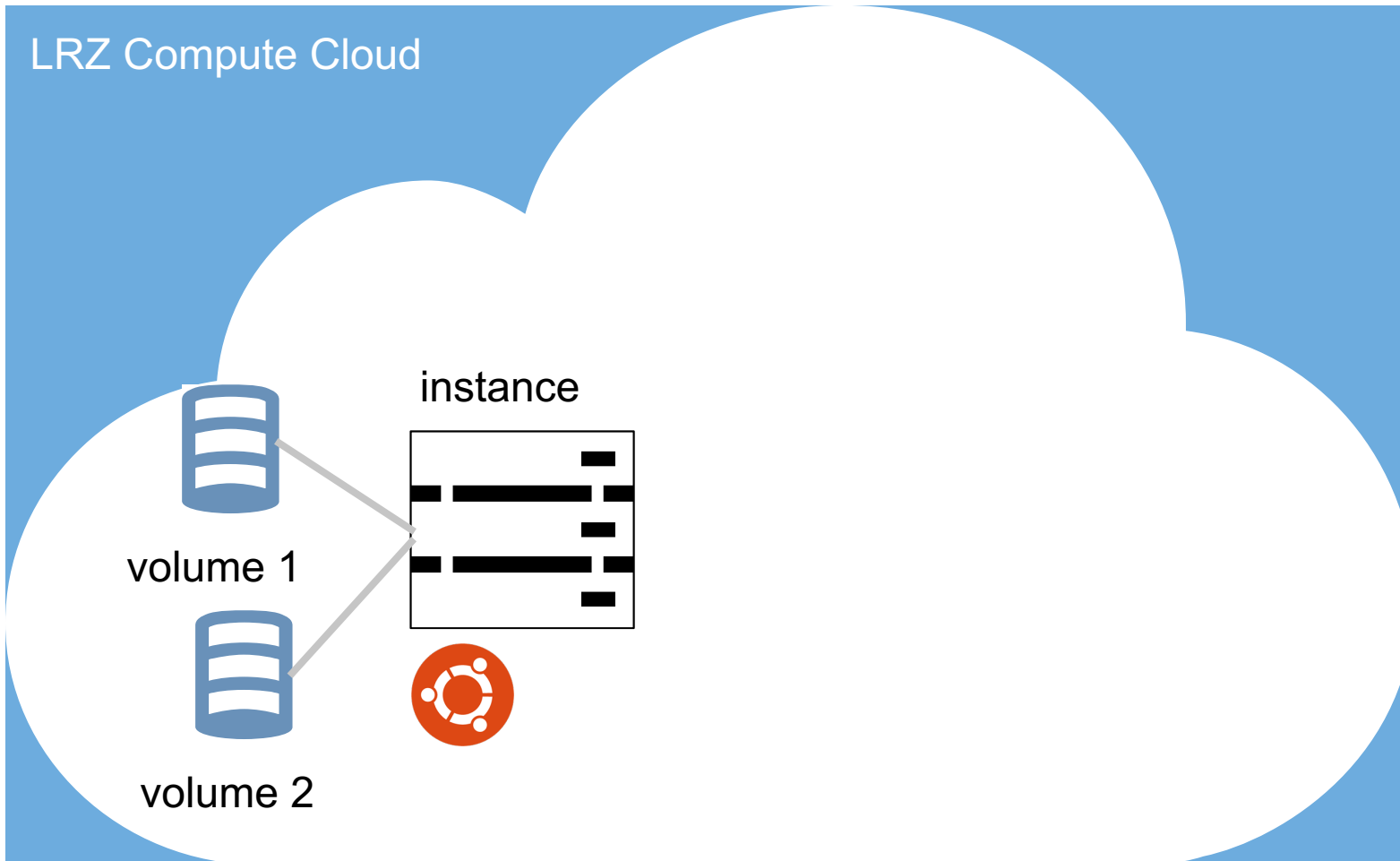


OpenStack - Terminology

- 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

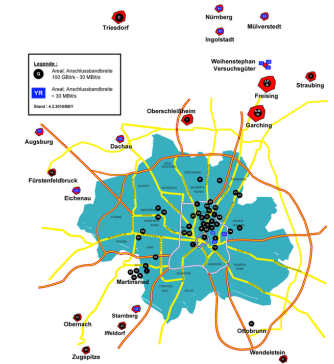
The LRZ Compute Cloud at a Glance



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.

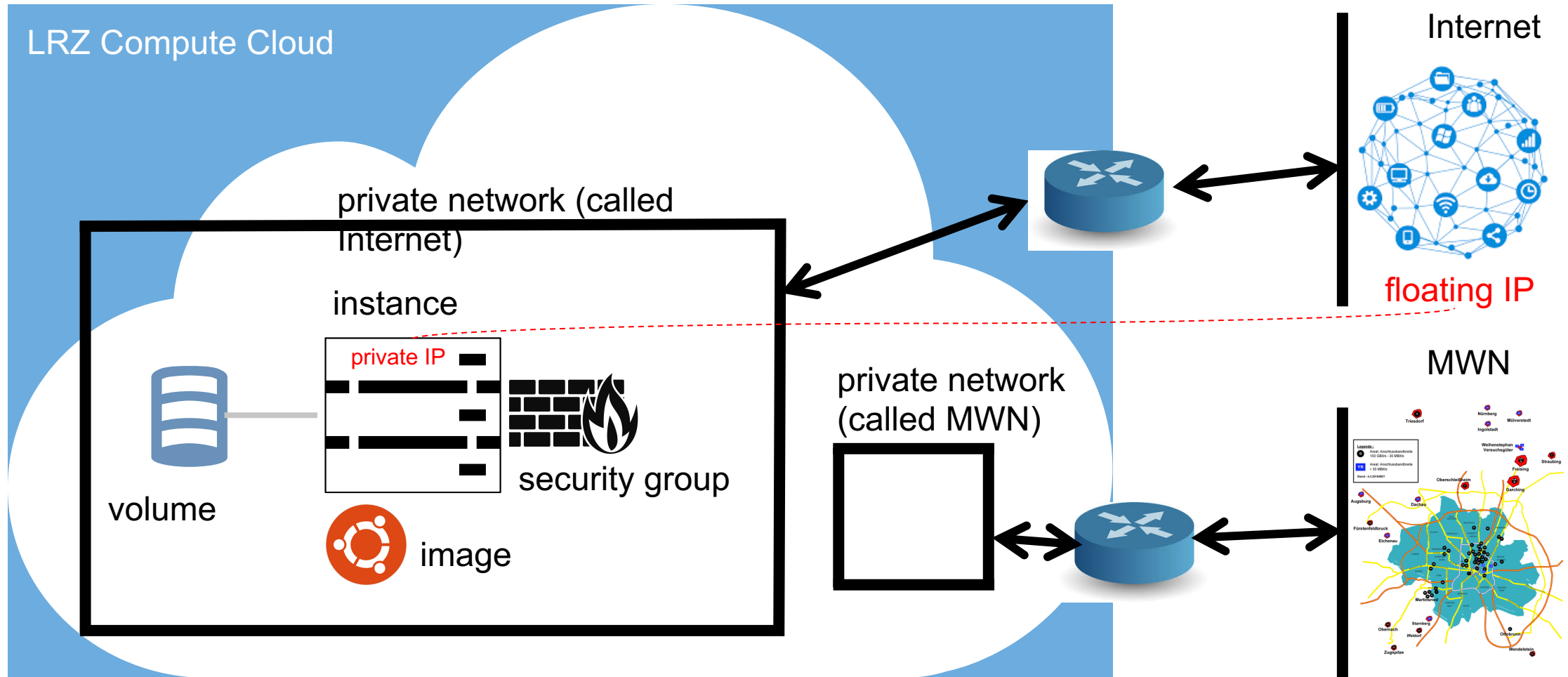
- Private and Floating IP

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.

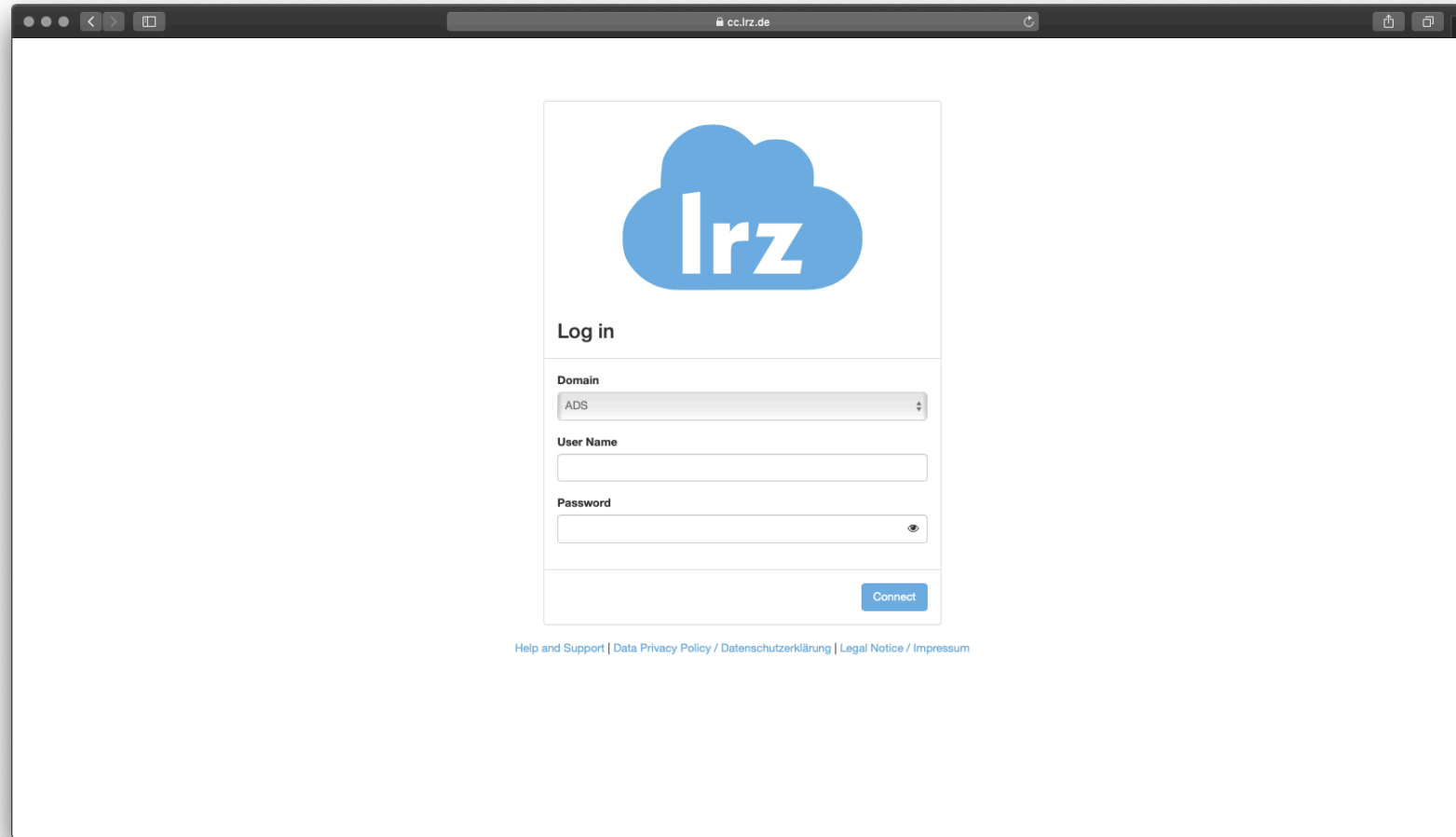
The LRZ Compute Cloud at a Glance



A server on the Compute Cloud

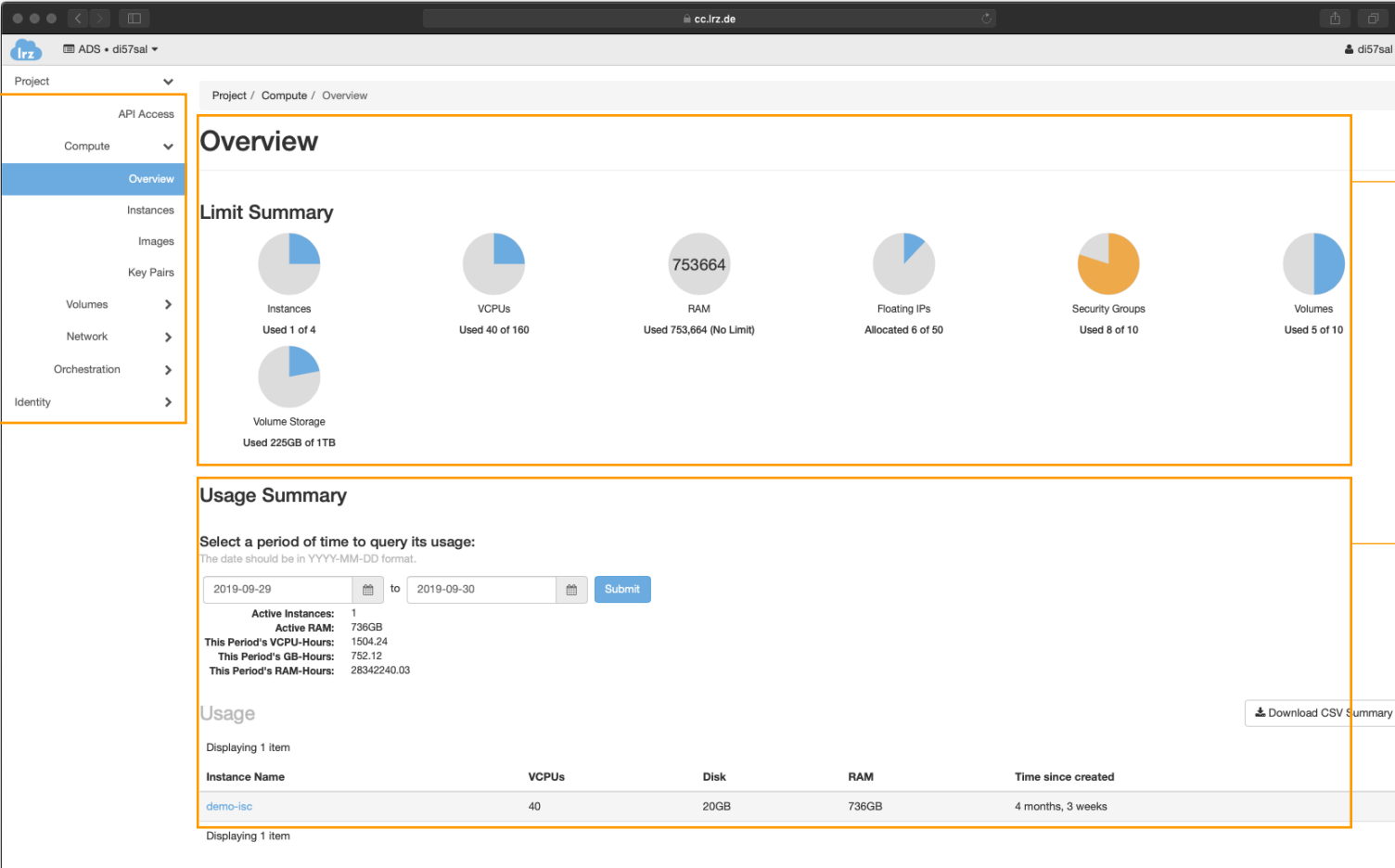
- De-facto operation is no Graphical Interface on the provided images
 - Although it is possible (e.g., <https://rv.lrz.de> ← only accessible from within MWN)
- Accessing instances via ssh
 - No login based on password by default (public and private keys!)
- OpenStack must be aware of your public key(s) 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

The Compute Cloud Web Interface



The Compute Cloud Web Interface

Left panel.
Allows you to
operate with the CC.



The screenshot shows the LRZ Compute Cloud Web Interface. The left sidebar contains a navigation menu with categories like API Access, Compute, Instances, Images, Key Pairs, Volumes, Network, Orchestration, and Identity. The main content area is titled 'Project / Compute / Overview' and is divided into two main sections: 'Limit Summary' and 'Usage Summary'.

Limit Summary: This section displays six resource usage metrics, each with a pie chart showing the percentage of resources used relative to the total limit. The metrics are:

- Instances: Used 1 of 4
- VCPUs: Used 40 of 160
- RAM: Used 753,664 (No Limit)
- Floating IPs: Allocated 6 of 50
- Security Groups: Used 8 of 10
- Volumes: Used 5 of 10

Usage Summary: This section allows users to query usage for a specific period. It includes a date range selector (from 2019-09-29 to 2019-09-30) and a 'Submit' button. Below the selector, it displays the following usage statistics:

- Active Instances: 1
- Active RAM: 736GB
- This Period's VCPU-Hours: 1504.24
- This Period's GB-Hours: 752.12
- This Period's RAM-Hours: 28342240.03

Below the usage summary is a table titled 'Usage' with a 'Download CSV Summary' button. The table displays the following data for one instance:

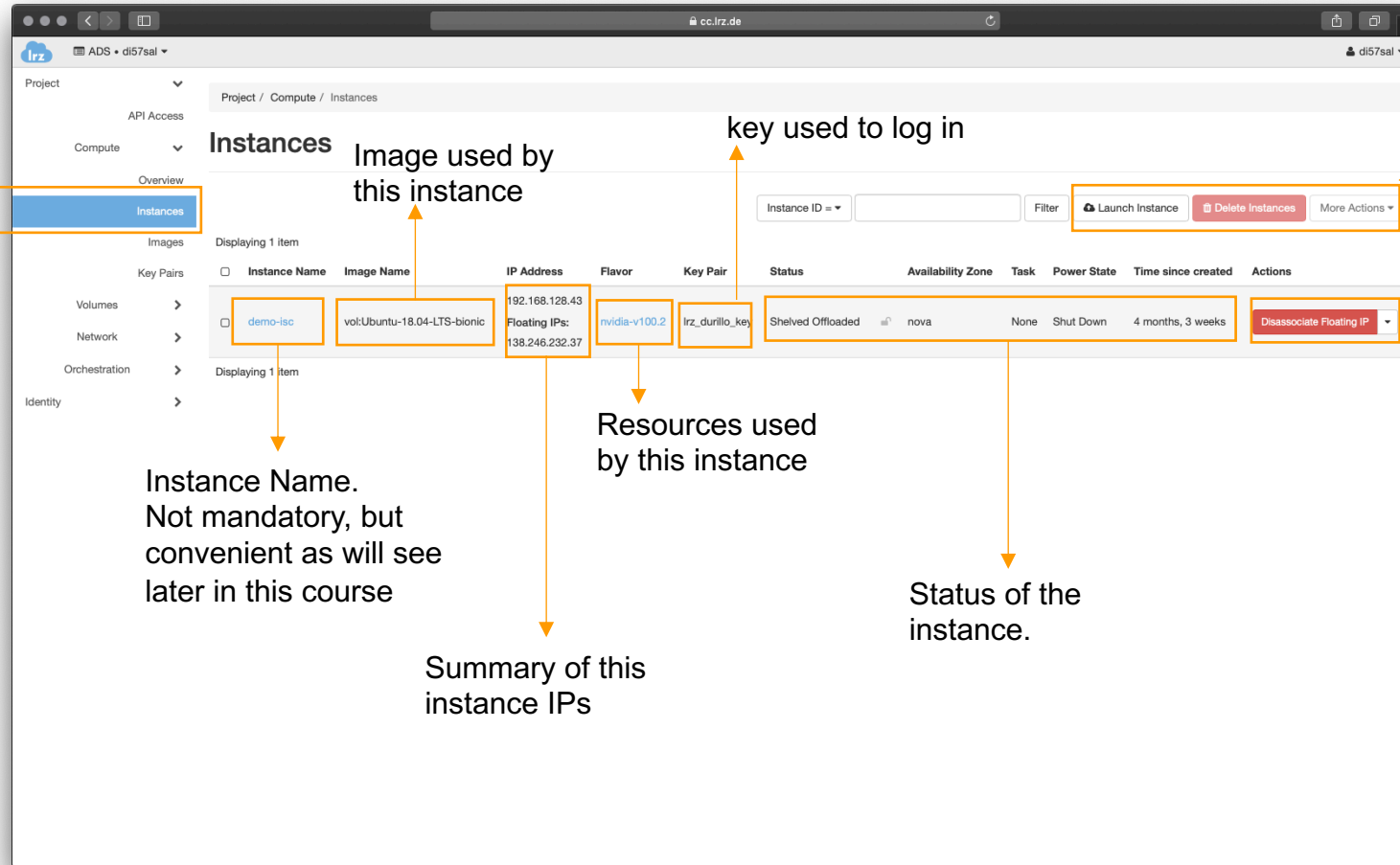
Instance Name	VCPUs	Disk	RAM	Time since created
demo-isc	40	20GB	736GB	4 months, 3 weeks

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



The screenshot shows the 'Instances' page in the Compute Cloud Web Interface. The interface includes a sidebar with navigation options like 'Overview', 'Images', 'Key Pairs', 'Volumes', 'Network', 'Orchestration', and 'Identity'. The main content area displays a table of instances with the following columns: Instance Name, Image Name, IP Address, Flavor, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. A single instance named 'demo-isc' is shown with the following details:

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
demo-isc	vol:Ubuntu-18.04-LTS-bionic	192.168.128.43 Floating IPs: 138.246.232.37	nvidia-v100.2	lrz_durillo_key	Shelved Offloaded	nova	None	Shut Down	4 months, 3 weeks	Disassociate Floating IP

Annotations on the screenshot include:

- An arrow pointing to the 'Instances' tab in the sidebar: 'Allows a more informative view of our instances'.
- An arrow pointing to the 'Image Name' column: 'Image used by this instance'.
- An arrow pointing to the 'Key Pair' column: 'key used to log in'.
- An arrow pointing to the 'IP Address' column: 'Summary of this instance IPs'.
- An arrow pointing to the 'Flavor' column: 'Resources used by this instance'.
- An arrow pointing to the 'Status' column: 'Status of the instance.'
- An arrow pointing to the 'Actions' column: 'Allows operating this instance (e.g., attach a volume, associate new floating IP)'.
- An arrow pointing to the 'Launch Instance' and 'Delete Instances' buttons: 'Allows creating new instances / Deleting existing ones'.

The Compute Cloud Web Interface



Project / Compute / Key Pairs

Displaying 2 Items

Name	Fingerprint	Actions
access_from_windows	51:93:d4:5f:c1:8b:09:fd:da:11:19:a3:1e:34:ec:32	Delete Key Pair
lrz_durillo_key	c8:aa:48:d1:64:03:42:b5:bb:b9:0a:a4:4d:9f:41:a3	Delete Key Pair

Project / Volumes / Volumes

Displaying 8 Items

Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
3896da63-2f67-4417-91bb-6a7d32d35cc8	-	30GiB	In-use	ceph	/dev/vda on test	nova	Yes	No	Edit Volume
0c5b5550-947b-44c5-a20a-46ba298a5d97	-	30GiB	Available	ceph		nova	Yes	No	Edit Volume
10a34d24-1d7b-46a4-bd3e-f1556a1d3918	-	30GiB	Available	ceph		nova	Yes	No	Edit Volume
tensorflow-gpu-volume	-	25GiB	Available	ceph		nova	Yes	No	Edit Volume
c706810c-dcd3-4f6a-9214-aadc8e6c901	-	20GiB	Available	ceph		nova	Yes	No	Edit Volume
63e9cb62-3bb5-44d3-be13-7421d12ff5a2	-	20GiB	Available	ceph		nova	Yes	No	Edit Volume
data	data for isc 2019 demo	80GiB	Reserved	ceph		nova	No	No	Update Metadata
f5bb9b3-c0c-4c0f-a79a-b396ee8a72d0	-	20GiB	Reserved	ceph		nova	Yes	No	Update Metadata

A Guided Example

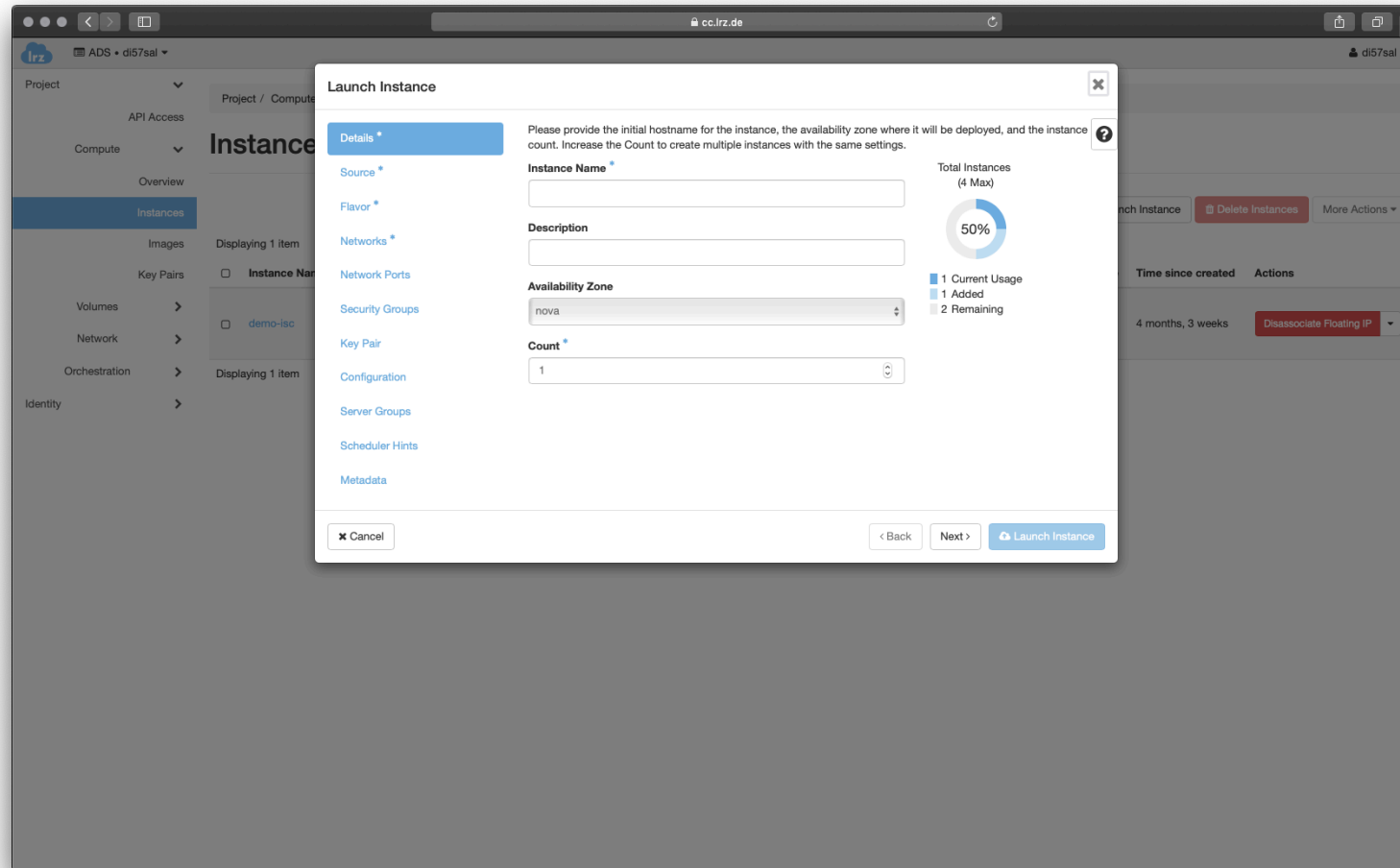
TASK : Create a Ubuntu-based server for running Jupyter Notebooks

- **On the Compute Cloud Web Interface**
 - Generate a new keypair (follow along the live demo)
- **On your computer**
 - 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
 - check <https://stackoverflow.com/questions/3190667/convert-pem-to-ppk-file-format> if you need help

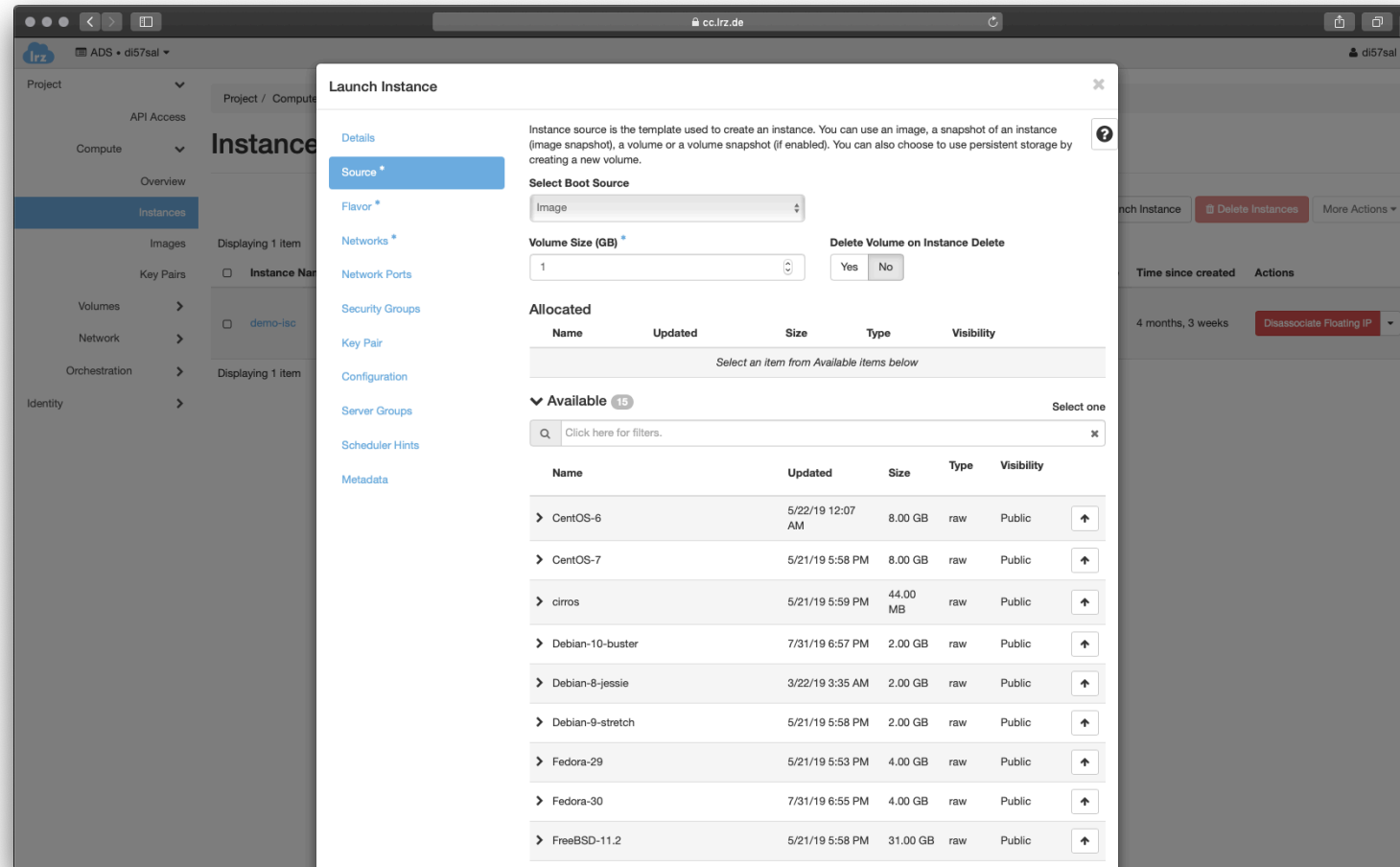
A Guided Example

- [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

Creating an Instance using the Web Interface



Creating an Instance using the Web Interface



Launch Instance

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

Select Boot Source

Image

Volume Size (GB) 1

Delete Volume on Instance Delete Yes No

Allocated

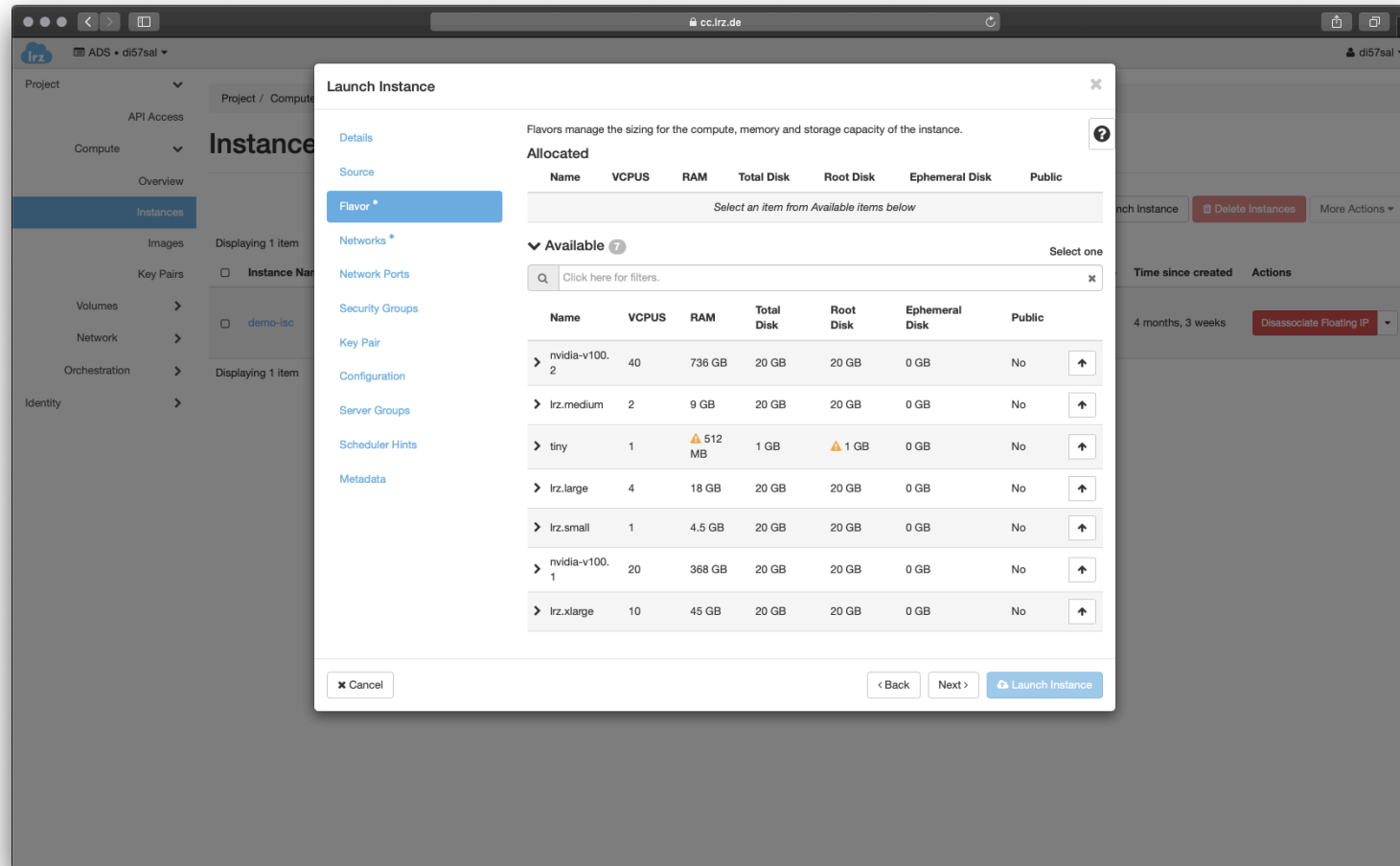
Name	Updated	Size	Type	Visibility
Select an item from Available items below				

Available 15 Select one

Click here for filters.

Name	Updated	Size	Type	Visibility
> CentOS-6	5/22/19 12:07 AM	8.00 GB	raw	Public
> CentOS-7	5/21/19 5:58 PM	8.00 GB	raw	Public
> cirros	5/21/19 5:59 PM	44.00 MB	raw	Public
> Debian-10-buster	7/31/19 6:57 PM	2.00 GB	raw	Public
> Debian-8-jessie	3/22/19 3:35 AM	2.00 GB	raw	Public
> Debian-9-stretch	5/21/19 5:58 PM	2.00 GB	raw	Public
> Fedora-29	5/21/19 5:53 PM	4.00 GB	raw	Public
> Fedora-30	7/31/19 6:55 PM	4.00 GB	raw	Public
> FreeBSD-11.2	5/21/19 5:58 PM	31.00 GB	raw	Public

Creating an Instance using the Web Interface



Launch Instance

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

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
Select an item from Available items below						

Available 7

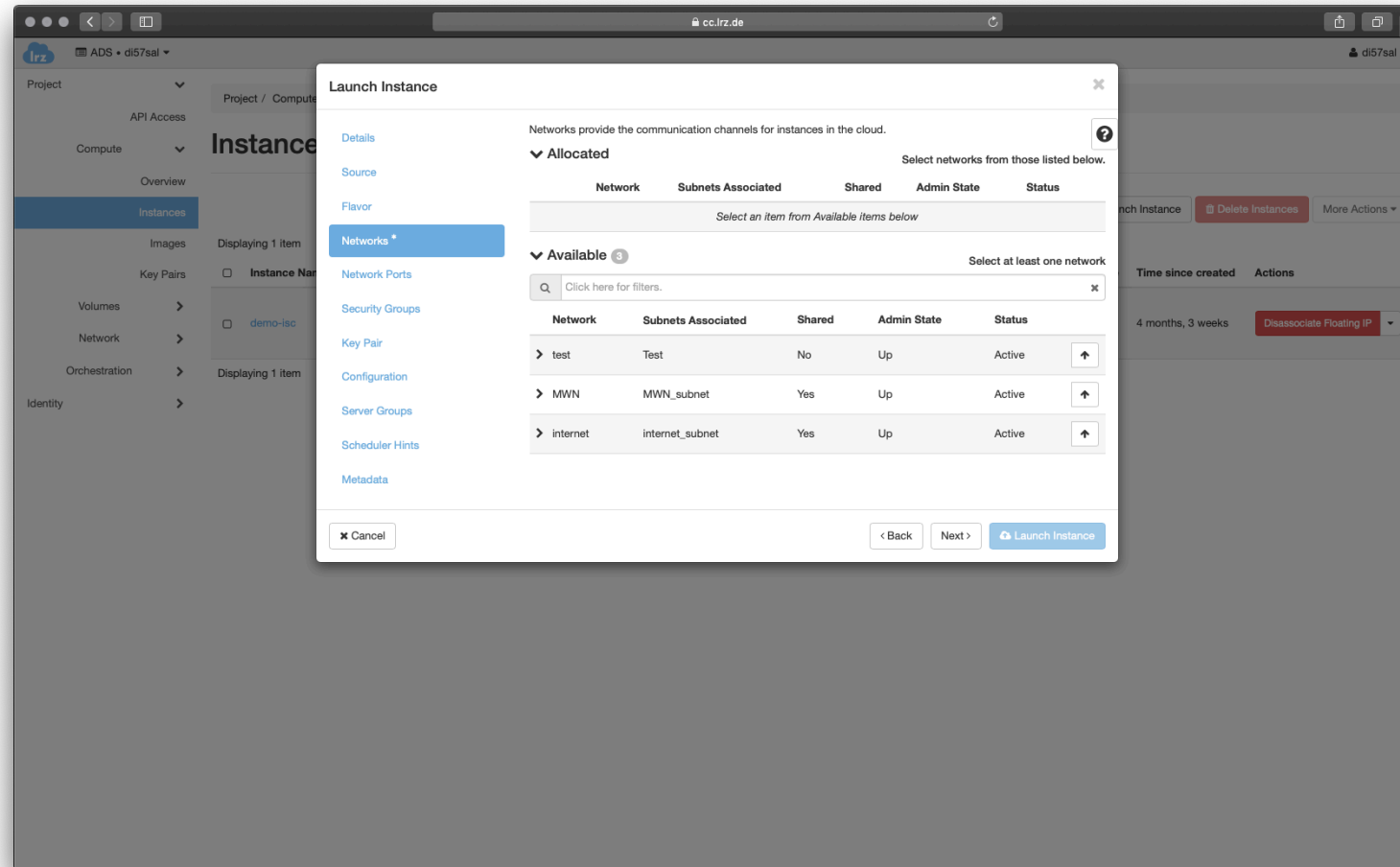
Select one

Click here for filters.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
> nvidia-v100.2	40	736 GB	20 GB	20 GB	0 GB	No
> lrz.medium	2	9 GB	20 GB	20 GB	0 GB	No
> tiny	1	512 MB	1 GB	1 GB	0 GB	No
> lrz.large	4	18 GB	20 GB	20 GB	0 GB	No
> lrz.small	1	4.5 GB	20 GB	20 GB	0 GB	No
> nvidia-v100.1	20	368 GB	20 GB	20 GB	0 GB	No
> lrz.xlarge	10	45 GB	20 GB	20 GB	0 GB	No

Cancel Back Next Launch Instance

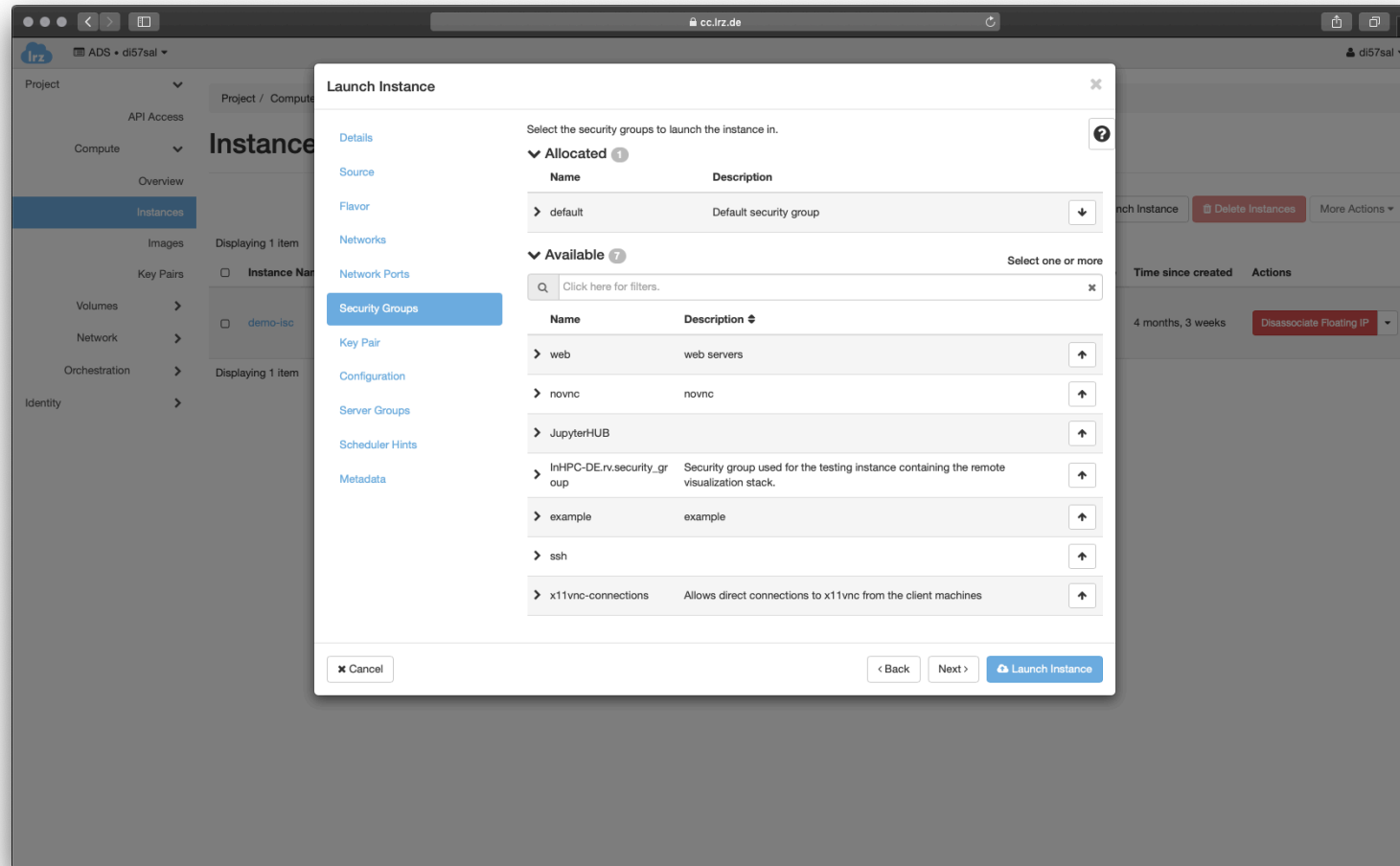
Creating an Instance using the Web Interface



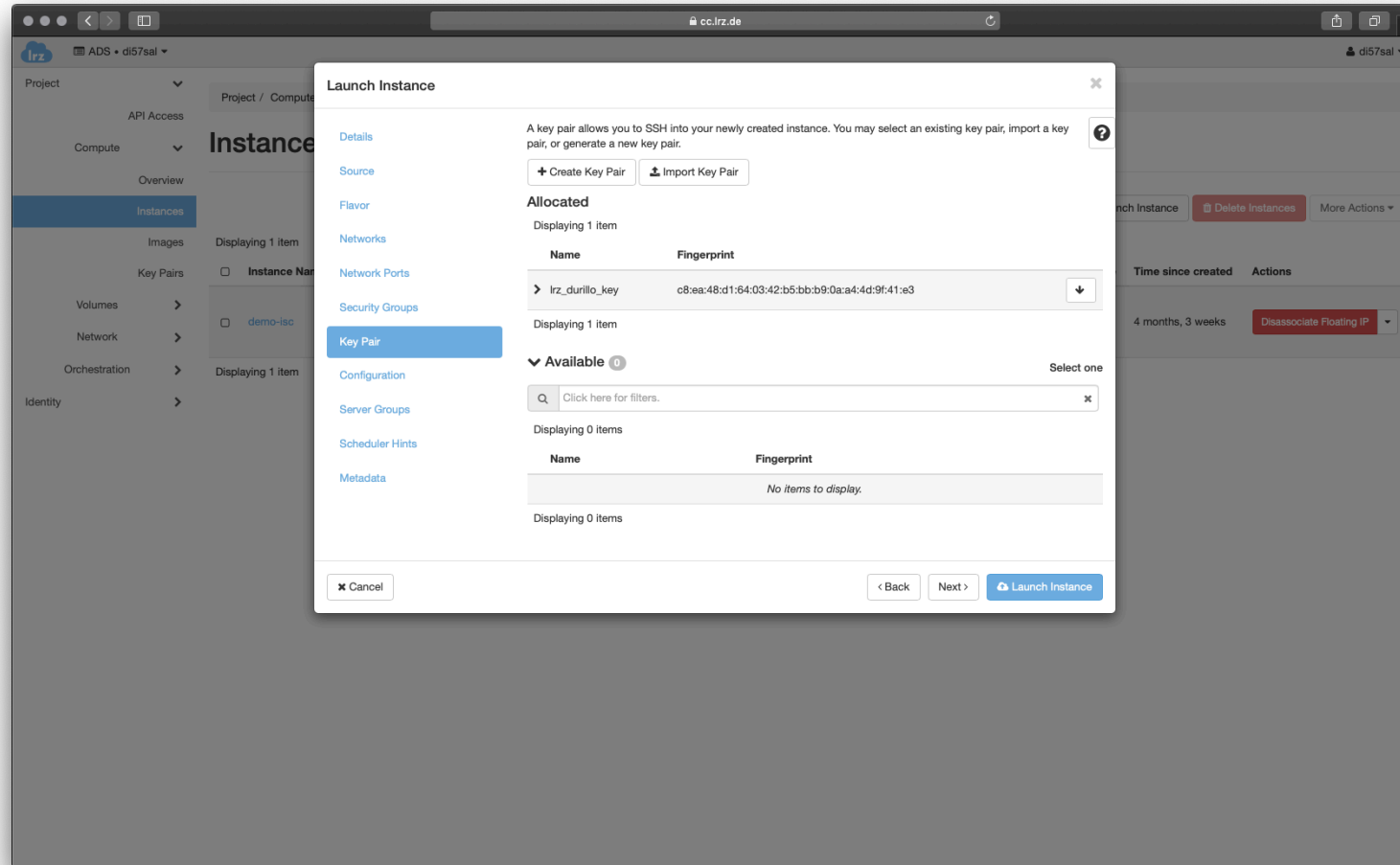
The screenshot shows the 'Launch Instance' dialog box in the OpenStack web interface. The dialog is titled 'Launch Instance' and has a close button (X) in the top right corner. The left sidebar of the dialog lists various configuration options: Details, Source, Flavor, Networks (selected), Network Ports, Security Groups, Key Pair, Configuration, Server Groups, Scheduler Hints, and Metadata. The main content area is titled 'Networks provide the communication channels for instances in the cloud.' and is divided into two sections: 'Allocated' and 'Available'. The 'Allocated' section is currently empty, with a prompt to 'Select networks from those listed below.' The 'Available' section has a search bar and a table of available networks. The table has columns for Network, Subnets Associated, Shared, Admin State, and Status. There are three rows of available networks: 'test', 'MWN', and 'internet'. Each row has a right arrow icon in the status column. At the bottom of the dialog, there are three buttons: 'Cancel', '< Back', and 'Next > Launch Instance'.

Network	Subnets Associated	Shared	Admin State	Status
> test	Test	No	Up	Active
> MWN	MWN_subnet	Yes	Up	Active
> internet	internet_subnet	Yes	Up	Active

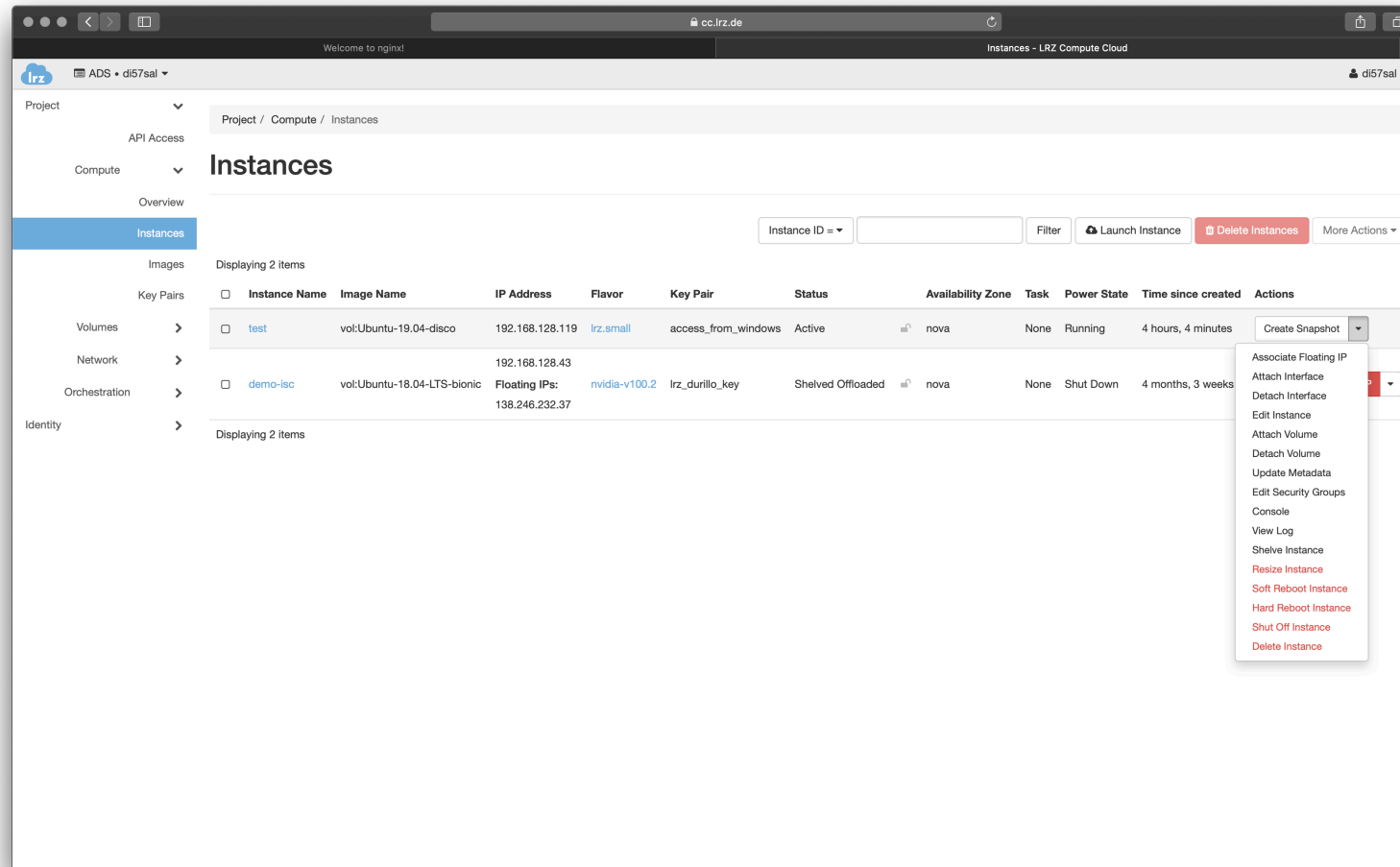
Creating an Instance using the Web Interface



Creating an Instance using the Web Interface



Creating an Instance using the Web Interface



Project / Compute / Instances

Instances

Instance ID = Filter Launch Instance Delete Instances More Actions

Displaying 2 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/> test	vol:Ubuntu-19.04-disco	192.168.128.119	lrz.small	access_from_windows	Active	nova	None	Running	4 hours, 4 minutes	Create Snapshot
<input type="checkbox"/> demo-isc	vol:Ubuntu-18.04-LTS-bionic	Floating IPs: 192.168.128.43 138.246.232.37	nvidia-v100.2	lrz_durillo_key	Shelved Offloaded	nova	None	Shut Down	4 months, 3 weeks	<ul style="list-style-type: none"> Associate Floating IP Attach Interface Detach Interface Edit Instance Attach Volume Detach Volume Update Metadata Edit Security Groups Console View Log Shelve Instance Resize Instance Soft Reboot Instance Hard Reboot Instance Shut Off Instance Delete Instance

Creating an Instance using the Web Interface



The screenshot shows the 'Security Groups' page in the lrz web interface. The page title is 'Security Groups' and it displays a table with 8 items. The table has columns for Name, Security Group ID, Description, and Actions. The 'default' security group is highlighted in blue.

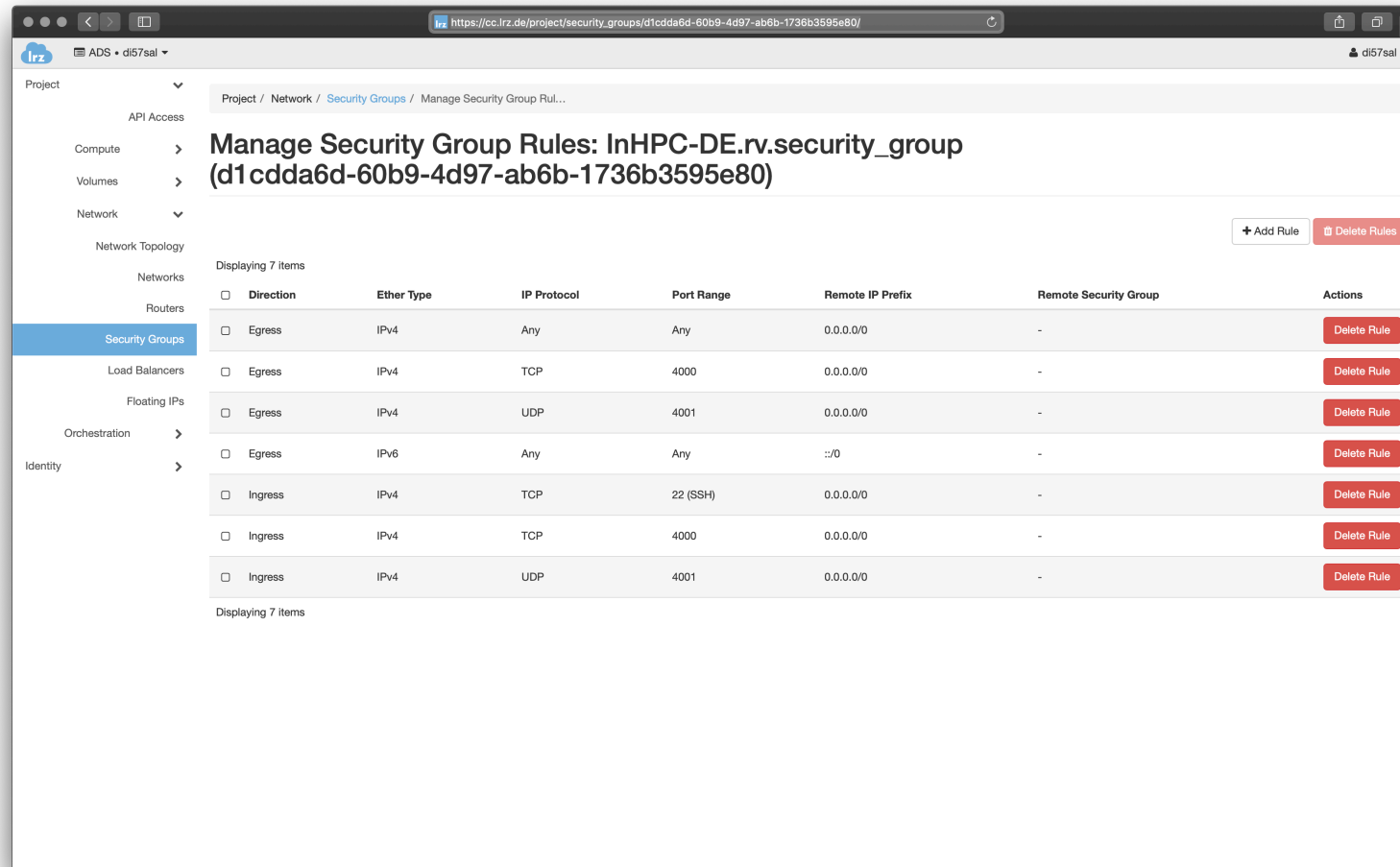
<input type="checkbox"/>	Name	Security Group ID	Description	Actions
<input type="checkbox"/>	InHPC-DE.rv.security_group	d1cdda6d-60b9-4d97-ab6b-1736b3595e80	Security group used for the testing instance containing the remote visualization stack.	Manage Rules
<input type="checkbox"/>	JupyterHUB	8474f2ab-1c1c-4477-aa24-cb4f6f848ef8		Manage Rules
<input type="checkbox"/>	default	2a0b66d5-7a9f-49ae-8d30-f11ddea6f968	Default security group	Manage Rules
<input type="checkbox"/>	example	d926448f-7830-4ec0-8888-a056ff9d4a1a	example	Manage Rules
<input type="checkbox"/>	novnc	65143f52-5536-4e8b-b977-8d0e36f2e056	novnc	Manage Rules
<input type="checkbox"/>	ssh	f4ec8f5f-8797-488a-9cd3-3da12a068cd5		Manage Rules
<input type="checkbox"/>	web	189b2ecd-d0bf-4b9c-a3fc-3f734d57dbe3	web servers	Manage Rules
<input type="checkbox"/>	x11vnc-connections	fb86cef1-828e-4172-a3f5-4346bb75e1e2	Allows direct connections to x11vnc from the client machines	Manage Rules

Creating an Instance using the Web Interface

The screenshot displays the OpenStack web interface for creating a security group. A modal dialog titled "Create Security Group" is open, featuring a "Name" input field and a "Description" text area. The description text reads: "Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group." Below the dialog, a table lists existing security groups with columns for Name, ID, and Description.

Name	ID	Description
InHPC-DE.rv.security_group		
JupyterHUB		
default	2a0b66d5-7a9f-49ae-8d30-f11ddea6f968	Default security group
example	d926448f-7830-4ec0-8888-a056ff9d4a1a	example
novnc	65143f52-5536-4e8b-b977-8d0e36f2e056	novnc
ssh	f4ec8f5f-8797-488a-9cd3-3da12a068cd5	
web	189b2ecd-d0bf-4b9c-a3fc-3f734d57dbe3	web servers
x11vnc-connections	fb86cef1-828e-4172-a3f5-4346bb75e1e2	Allows direct connections to x11vnc from the client machines

Creating an Instance using the Web Interface



The screenshot shows a web browser window displaying the LRZ management interface. The page title is "Manage Security Group Rules: InHPC-DE.rv.security_group (d1cdda6d-60b9-4d97-ab6b-1736b3595e80)". The interface includes a left-hand navigation menu with categories like Project, Compute, Volumes, Network, and Security Groups. The main content area displays a table of security group rules with columns for Direction, Ether Type, IP Protocol, Port Range, Remote IP Prefix, Remote Security Group, and Actions. There are buttons for "+ Add Rule" and "Delete Rules" at the top right of the table.

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	TCP	4000	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	UDP	4001	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	4000	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	UDP	4001	0.0.0.0/0	-	Delete Rule

A Guided Example

Subtask: Access the created instance via SSH

- On the Computer Cloud Web Interface
 - **Create a security group that allow ingress connections to port 22!**
 - Add this security group 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

A Guided Example

- Install pip



```
$ sudo apt update
$ sudo apt install python3-pip
```

- Install jupyterlab using pip



```
$ pip install jupyterlab
```

- Run jupyter-lab to listen on the private IP of the machine



```
$ jupyter-lab --ip=192.168.3.140
```

You can check the IP using ifconfig (you need to install net-tools)

A Guided Example

- Connect to the showed URL with a browser. Why is not working?

A terminal window with a dark background and a grey title bar containing three colored circles (red, orange, green). The terminal text is as follows:

```
$ jupyter-lab --ip=192.168.3.140
Or copy and paste one of these URLs:

http://192.168.3.140:8888/lab?token=d61e9775638219bcdc0260b1cd58
4767467eed3f7d440ff9
or
http://127.0.0.1:8888/lab?token=d61e9775638219bcdc0260b1cd584767
467eed3f7d440ff9
```

- On the Compute Cloud Web Interface
 - **Create a security group that allow ingress connections to port 8888!**
 - Add this security group to the instance

A Guided Example

- Connect to the showed URL with a browser. Why is still not working?

A terminal window with a dark background and a grey title bar containing three colored window control buttons (red, orange, green). The terminal text is white and shows the command to start Jupyter Lab with a specific IP address, followed by instructions to copy and paste one of two URLs. The first URL uses the IP 192.168.3.140 and port 8888. The second URL uses the IP 127.0.0.1 and port 8888. Both URLs include a long alphanumeric token.

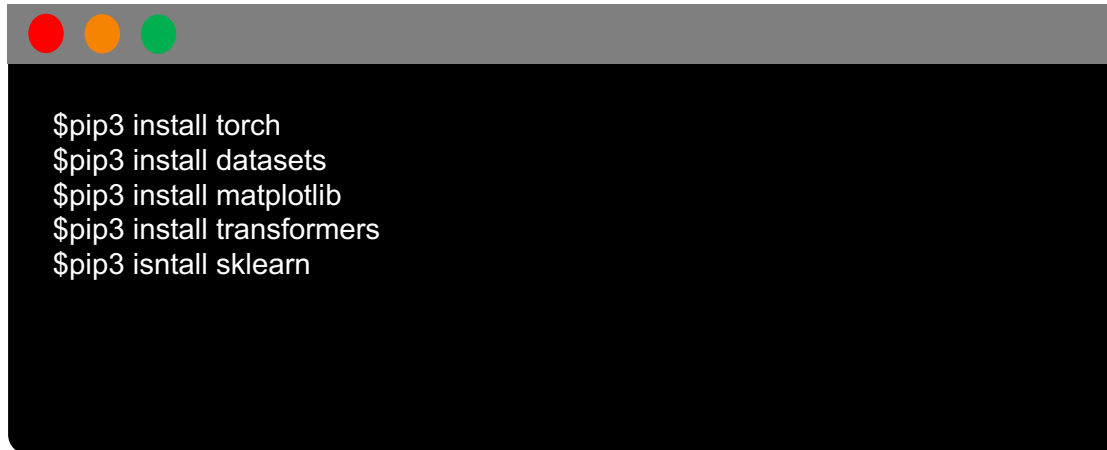
```
$ jupyter-lab --ip=192.168.3.140
Or copy and paste one of these URLs:

http://192.168.3.140:8888/lab?token=d61e9775638219bcdc0260b1cd58
4767467eed3f7d440ff9
or
http://127.0.0.1:8888/lab?token=d61e9775638219bcdc0260b1cd584767
467eed3f7d440ff9
```

- The showed URL is relative to the Compute Cloud instance private network.
- Substitute in that URL the IP with the floating IP of the instance and try accessing the instance now

A Guided Example

- Not done: A few libraries are missing for executing our Jupyter Notebook Example



```
$pip3 install torch
$pip3 install datasets
$pip3 install matplotlib
$pip3 install transformers
$pip3 isntall sklearn
```

Summary

- Overview of the LRZ Resources for ML/DL Workloads
- Focus session on the LRZ AI System, a scenario intended for ML/DL training
- Focus session on the LRZ Compute Cloud, a scenario intended for ML/DL development and inference