Date: 2017/3/25
Author: @cloudsskyone

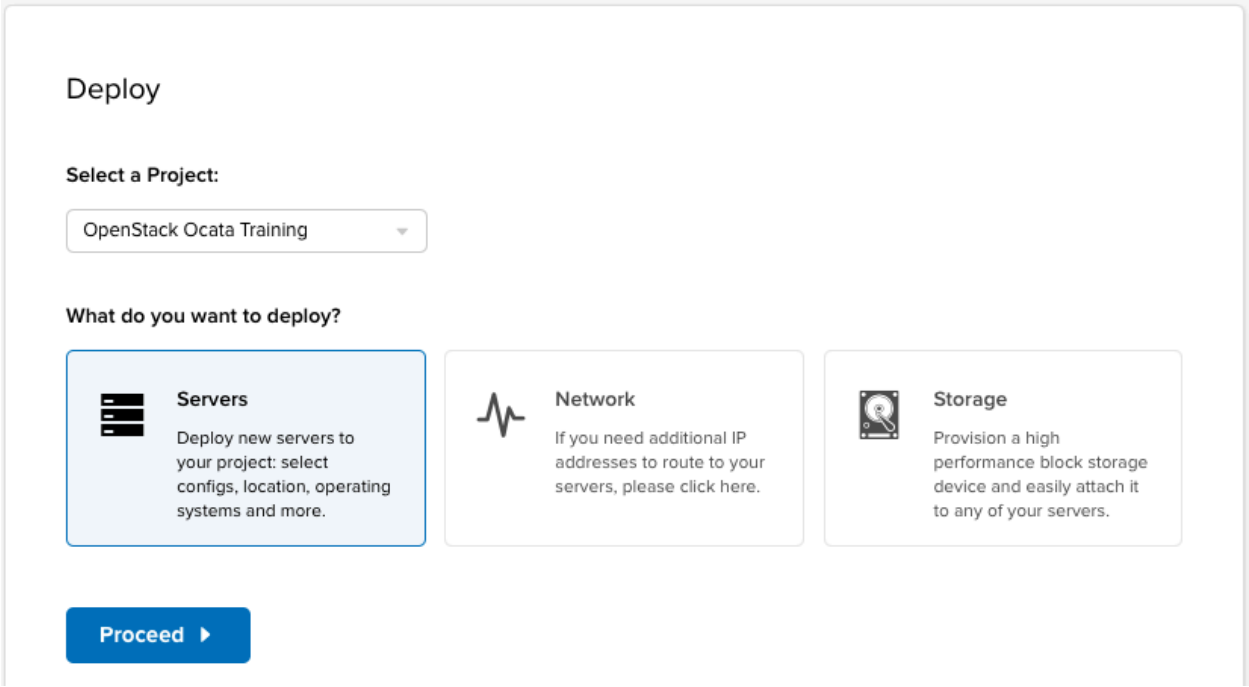
Multi-Node OpenStack Ocata Deployment on Packet.net with RDO Packstack

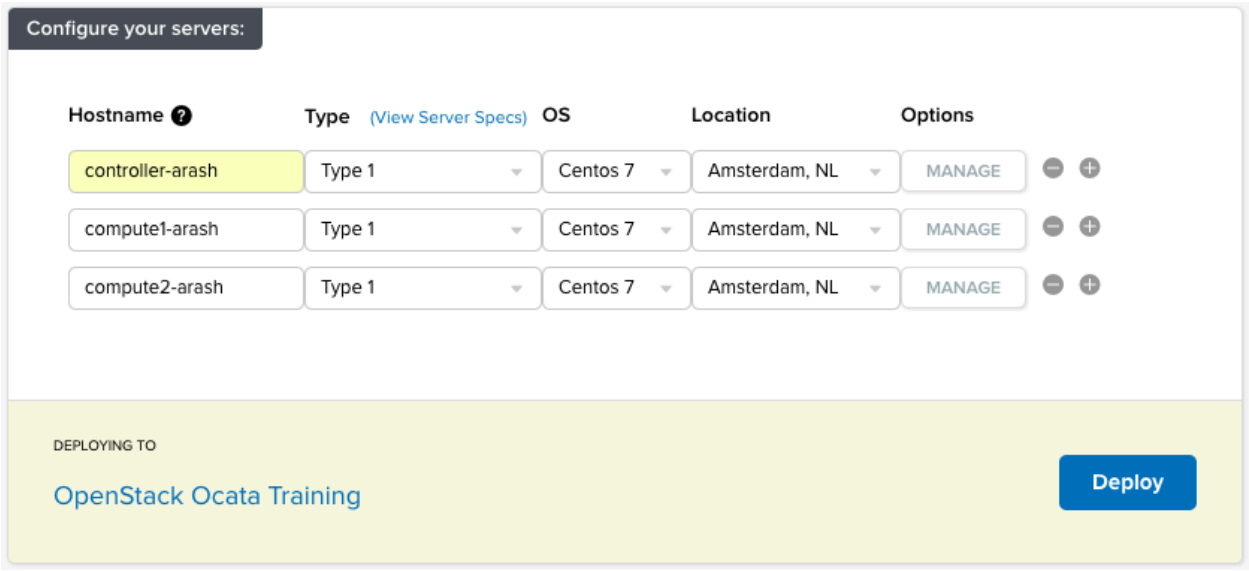
I'm writing this short howto for our todays Hands On OpenStack Ocata Deployment Online session which is going to take place every last Saturday each 2 month through our OpenStack Cologne Meetup group. We have decided to go with Packet.net for several reasons such as the great support from packet team, low prices for development and testing compared to other bare metal cloud providers and the fact that Packet is built for developers and operators and provides a clean api and supports our much beloved Terraform for provisioning servers.

For our training we're going to provide 3 CentOS 7 servers of Type 1 with 4 CPU, 32 GB RAM and 120 GB of SSD in RAID 1 for each Stacker attending to our online deployment session and trainings.

On Packet we've created a project called "OpenStack Ocata Training", imported our SSH public key and have created an API Key, which we can use in our upcoming sessions to create servers with Terraform programmatically. For this deployment we're using Packstack from RDO project, since this is the fastest and the best way to deploy a production ready OpenStack environment for development, testing and training. For larger enterprise grade deployments we recommend to use RDO TripleO with RHOSP or Kolla-Ansible or hopefully soon the Kolla-Kubernetes project.

Creating Servers on Packet.net is a breeze, it takes less than 5 minute, that's awesome (you don't need to do that for our training, since the servers are already provisioned for you), but if you'd like to use your own servers and keep them for a while, what you need to do is to create an account, a project and click on "Deploy" and select "Servers":



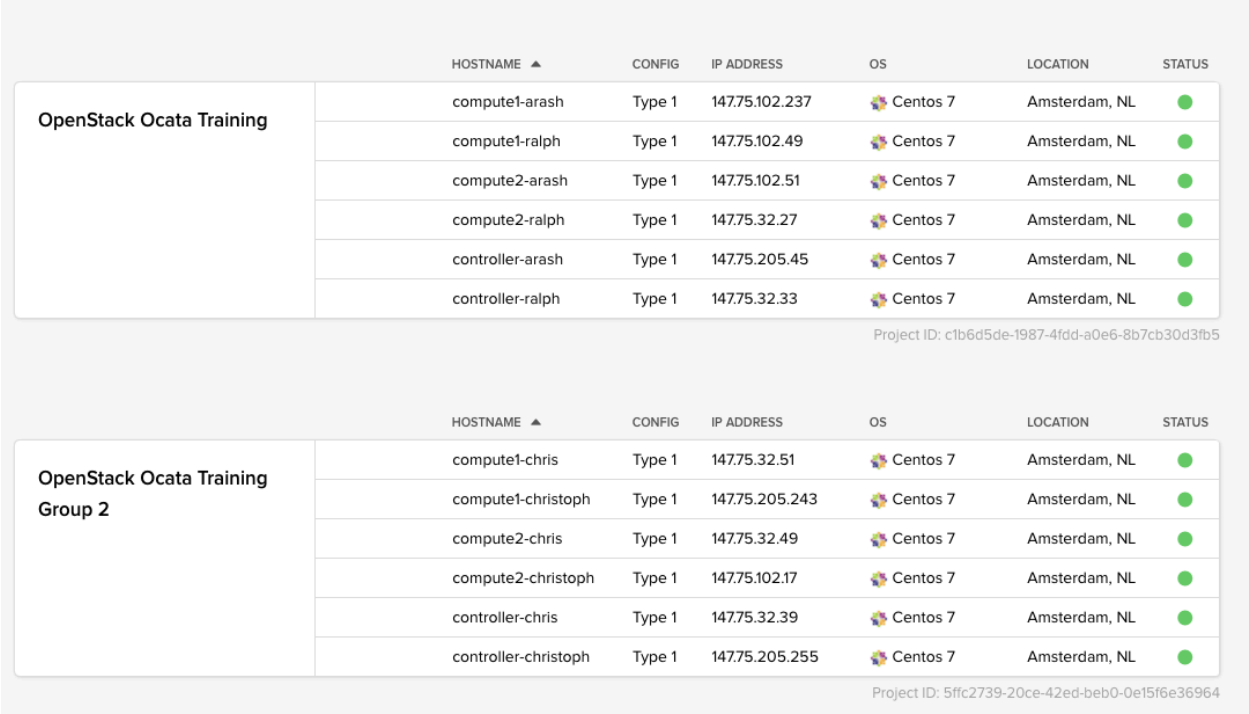


For our 3 node deployment we're using 1 controller and 2 compute nodes, after creating the servers we need to login to the servers with our private key with:

$ ssh -i private-key.pem root@<ip of the controller or compute nodes>

On Packet we've created 12 servers for each participant in 2 different groups, the IP's have was provided to each user through the meetup messaging dashboard.

Each participant will have access to 1 controller and 2 compute nodes:



	HOSTNAME	CONFIG	IP ADDRESS	OS	LOCATION	STATUS
OpenStack Ocata Training	compute1-arash	Type 1	147.75.102.237	Centos 7	Amsterdam, NL	
	compute1-ralph	Type 1	147.75.102.49	Centos 7	Amsterdam, NL	
	compute2-arash	Type 1	147.75.102.51	Centos 7	Amsterdam, NL	
	compute2-ralph	Type 1	147.75.32.27	Centos 7	Amsterdam, NL	
	controller-arash	Type 1	147.75.205.45	Centos 7	Amsterdam, NL	
	controller-ralph	Type 1	147.75.32.33	Centos 7	Amsterdam, NL	

Project ID: c1b6d5de-1987-4fdd-a0e6-8b7cb30d3fb5

	HOSTNAME	CONFIG	IP ADDRESS	OS	LOCATION	STATUS
OpenStack Ocata Training Group 2	compute1-chris	Type 1	147.75.32.51	Centos 7	Amsterdam, NL	
	compute1-christoph	Type 1	147.75.205.243	Centos 7	Amsterdam, NL	
	compute2-chris	Type 1	147.75.32.49	Centos 7	Amsterdam, NL	
	compute2-christoph	Type 1	147.75.102.17	Centos 7	Amsterdam, NL	
	controller-chris	Type 1	147.75.32.39	Centos 7	Amsterdam, NL	
	controller-christoph	Type 1	147.75.205.255	Centos 7	Amsterdam, NL	

Project ID: 5ffc2739-20ce-42ed-beb0-0e15f6e36964

Controller Setup

On the controller node we need to update the server, install the latest RDO CentOS Ocata Release packages and the Packstack tool, generate the packstack answer file, enable root login and set the root password.

yum update -y (has been done already for you for the training)
reboot (has been done already for you for the training)

Run these commands on the controller during our training session:

 # yum install centos-release-openstack-ocata -y
 # yum update -y
 # yum install -y openstack-packstack
 # packstack --gen-answer-file=ocata-2-node-provider-flat

The following has been done already for you for the training:

vi /etc/ssh/sshd_config
Set PermitRootLogin yes
Uncomment PasswordAuthentication yes
Comment out #PasswordAuthentication no

Restart sshd service
systemctl restart sshd.service

Set root password
passwd

reboot

Compute Nodes Setup

Skip this section for our training, this has been done for you:

yum update -y
reboot

vi /etc/ssh/sshd_config
Set PermitRootLogin yes
Uncomment PasswordAuthentication yes
Comment out #PasswordAuthentication no

Restart sshd service
systemctl restart sshd.service

Set the root password
passwd

Adapt the Packstack Answer File

Before we start the deployment, we have to adapt the packstack answer file with the IPs of our compute nodes and change some values for some few keys in the packstack answer file as shown below.

 # vi ocata-2-node-provider-flat

 # We want to use our own tenant / project and user, disable DEMO project / user
CONFIG_PROVISION_DEMO=n
 # SSL doesn't work for Horizon out of the box (is disabled by default)
CONFIG_HORIZON_SSL=n

CONFIG_CONTROLLER_HOST=147.75.205.45
 # Provide the IPs of your compute nodes
CONFIG_COMPUTE_HOSTS=147.75.102.237,147.75.102.51
 # We're using the controller as network node as well (don't change it)
CONFIG_NETWORK_HOSTS=147.75.205.45
 # Define NTP servers for time synchronization
CONFIG_NTP_SERVERS=0.pool.ntp.org,1.pool.ntp.org,2.pool.ntp.org,3.pool.ntp.org
 # Using provider network (was set to br-ex, which doesn't make any sense)
CONFIG_NEUTRON_L3_EXT_BRIDGE=provider
CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS=extnet:br-ex
CONFIG_NEUTRON_OVS_EXTERNAL_PHYSNET=extnet
 # Set the right interface (eth0) for Open vSwitch bridge on the controller / network node
CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-ex:eth0
# Increase the Cinder Volume Size, was set to 20G
CONFIG_CINDER_VOLUMES_SIZE=100G

Run Packstack and start the deployment

Now we can run packstack with our answer-file as follow and start our multi-node deployment:

 # packstack --answer-file=ocata-2-node-provider-flat

The deployment will take only 15 minutes, the complete output has been truncated here, at the end you shall get something like this:

```
Copying Puppet modules and manifests                [ DONE ]
Applying 147.75.205.45_controller.pp
147.75.205.45_controller.pp:                        [ DONE ]
Applying 147.75.205.45_network.pp
147.75.205.45_network.pp:                           [ DONE ]
Applying 147.75.102.51_compute.pp
Applying 147.75.102.237_compute.pp
147.75.102.237_compute.pp:                          [ DONE ]
147.75.102.51_compute.pp:                           [ DONE ]
Applying Puppet manifests                           [ DONE ]
Finalizing                                          [ DONE ]

 **** Installation completed successfully ******

Additional information:
 * File /root/keystonerc_admin has been created on OpenStack client host 147.75.205.45.
 * To access the OpenStack Dashboard browse to http://147.75.205.45/dashboard .
Please, find your login credentials stored in the keystonerc_admin in your home director
 * Because of the kernel update the host 147.75.102.51 requires reboot.
 * Because of the kernel update the host 147.75.205.45 requires reboot.
 * Because of the kernel update the host 147.75.102.237 requires reboot.
 * The installation log file is available at: /var/tmp/packstack/20170325-101624-g3WPX9/
 * The generated manifests are available at: /var/tmp/packstack/20170325-101624-g3WPX9/m
```

After the initial deployment it might be a good idea to reboot all servers (I didn't do it and I'm not sure if this is really needed).

To be able to log into the Horizon dashboard, you'd need to grab the admin password from the keystonerc_admin file:

```
[root@controller-arash ~]# cat keystonerc_admin
unset OS_SERVICE_TOKEN
export OS_USERNAME=admin
export OS_PASSWORD=xxxxxxxxxxxxxxxxxxxxxxxxx
export OS_AUTH_URL=http://147.75.205.45:5000/v3
export PS1='[\u@\h \W(keystone_admin)]\$ '

export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
```

Log into Horizon Dashboard as the admin user with the password provided above for your installation (replace the IP with the IP of your controller):
http://147.75.205.45/dashboard

Create Networks

Before creating public and private networks, we need to source our keystonerc_admin file:

 # source keystonerc_admin

Create the public network

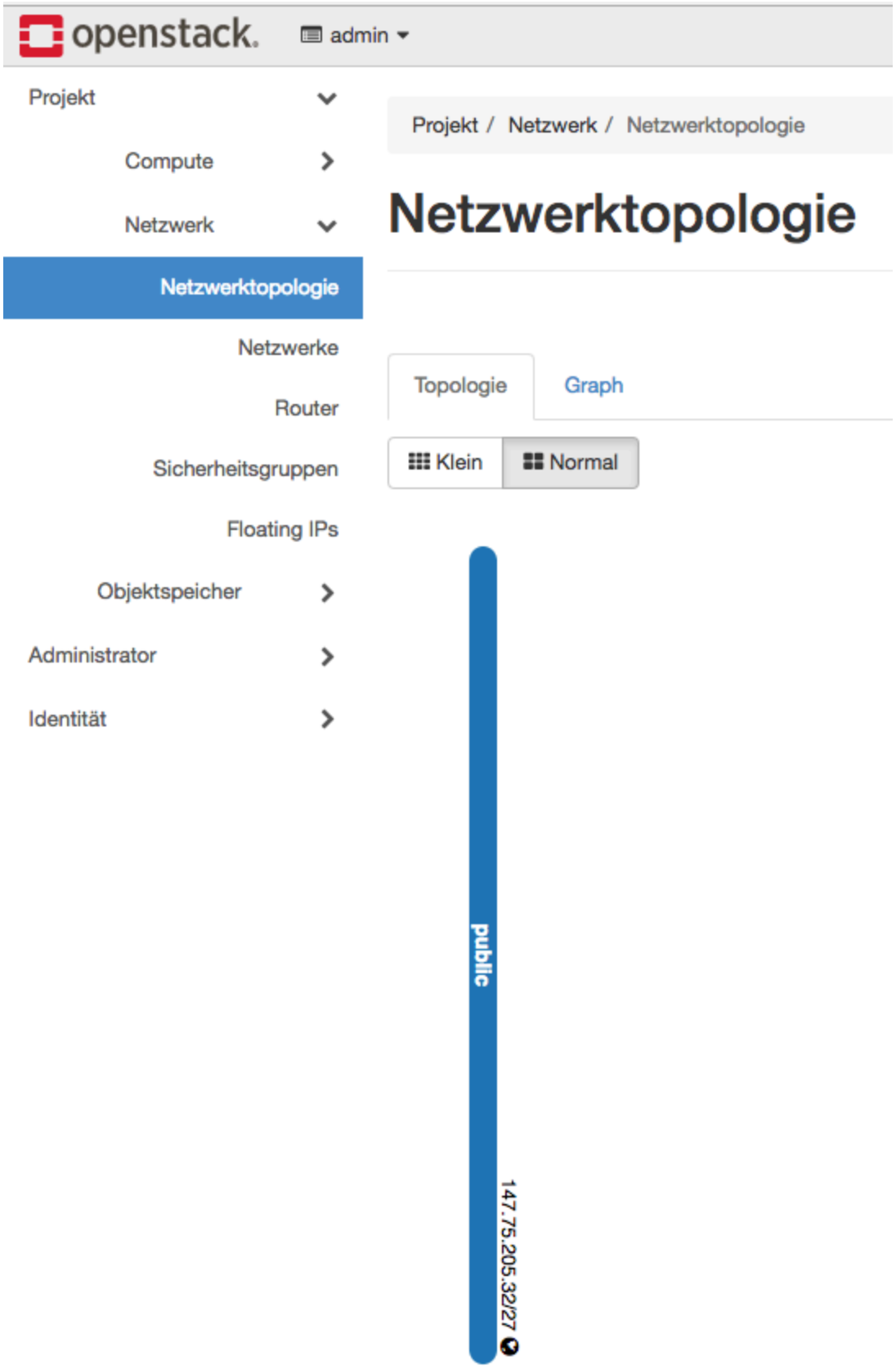
 # neutron net-create public --provider:network_type flat --provider:physical_network extnet --router:external

Create the public subnet

To create the public subnet, one needs to request an IP block from packet.net, but for our example here we assume we can use a floating IP range from 147.75.205.46 to 147.75.205.50 with the network 147.75.205.32/27

 # neutron subnet-create --gateway 147.75.205.44 --allocation-pool start=147.75.205.46,end=147.75.205.50 --disable-dhcp --name public_subnet public 147.75.205.32/27

After creating the public network, head to the Horizon Dashboard and verify under the network if the public network is showing up properly:


openstack.
admin
Projekt
Compute
Netzwerk
Netzwerktopologie
Netzwerke
Router
Sicherheitsgruppen
Floating IPs
Objektspeicher
Administrator
Identität
Projekt / Netzwerk / Netzwerktopologie
Netzwerktopologie
Topologie
Graph
Klein
Normal
public
147.75.205.32/27

Create a private network with a subnet

First we'll create a private network named private with the subnet private_subnet and the CIRD 192.168.11.0/24 and enable DHCP and define the name servers

 # neutron net-create private

 # neutron subnet-create private 192.168.11.0/24 --name private_subnet --enable-dhcp=True --dns-nameserver 8.8.8.8 --dns-nameserver 8.8.4.4

And create a router named router1:

 # neutron router-create router1

Set the gateway for this router to the public network:

 # neutron router-gateway-set router1 public

And finally set the router interface to the private subnet:

 # neutron router-interface-add router1 private_subnet

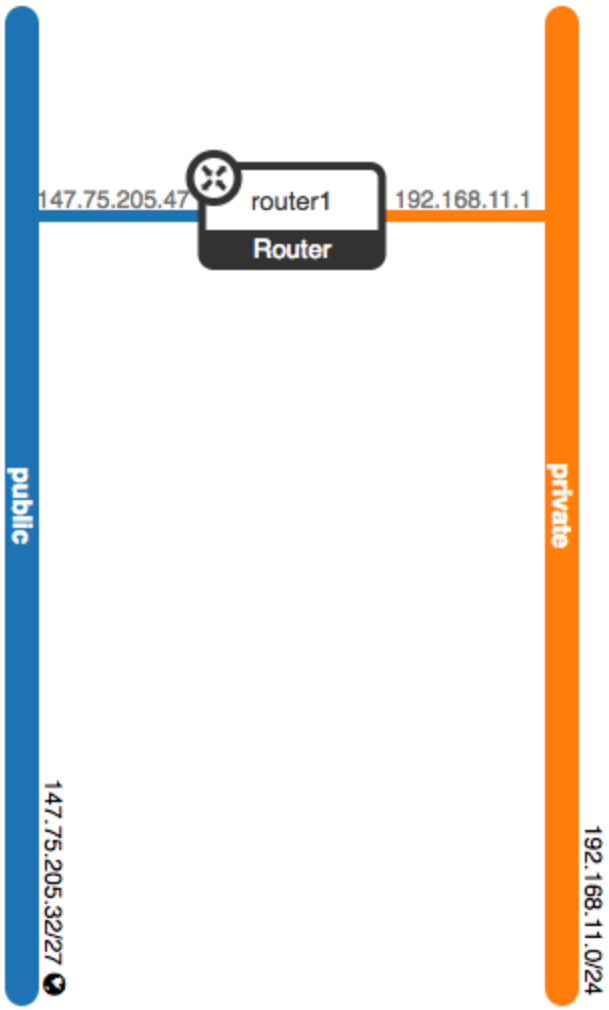
In Horizon we shall see our new private subnet and the router:

Projekt / Netzwerk / Netzwerktopologie

Netzwerktopologie

Topologie | Graph

Klein | Normal



Add an Image to glance

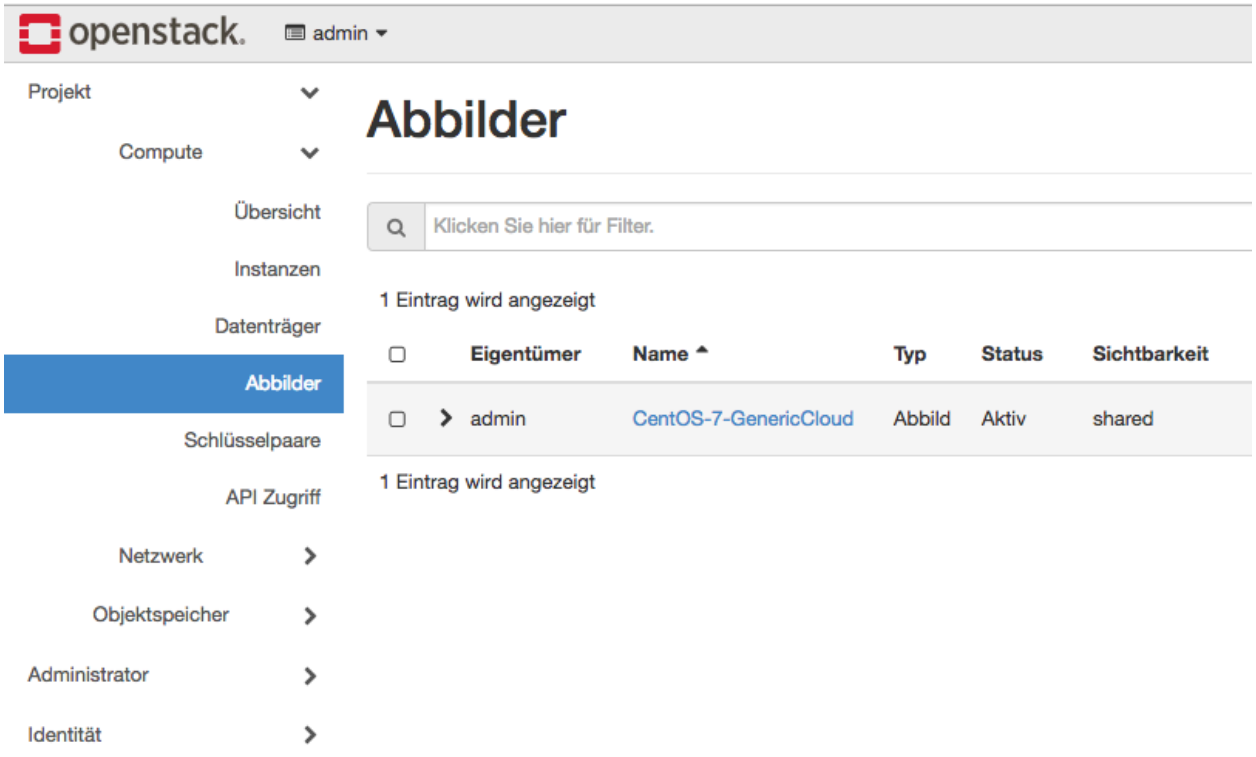
Download the latest CentOS QCOW2 Image and extract the qcow2 image:

 # wget http://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud.qcow2.xz
 # xz -d CentOS-7-x86_64-GenericCloud.qcow2.xz

Add the image to glance:

 # glance image-create --container-format=bare --disk-format=qcow2
--name=CentOS-7-GenericCloud < CentOS-7-x86_64-GenericCloud.qcow2

In Horizon navigate to Compute → Images (in German it's named Abbilder) and verify if your image is listed there:



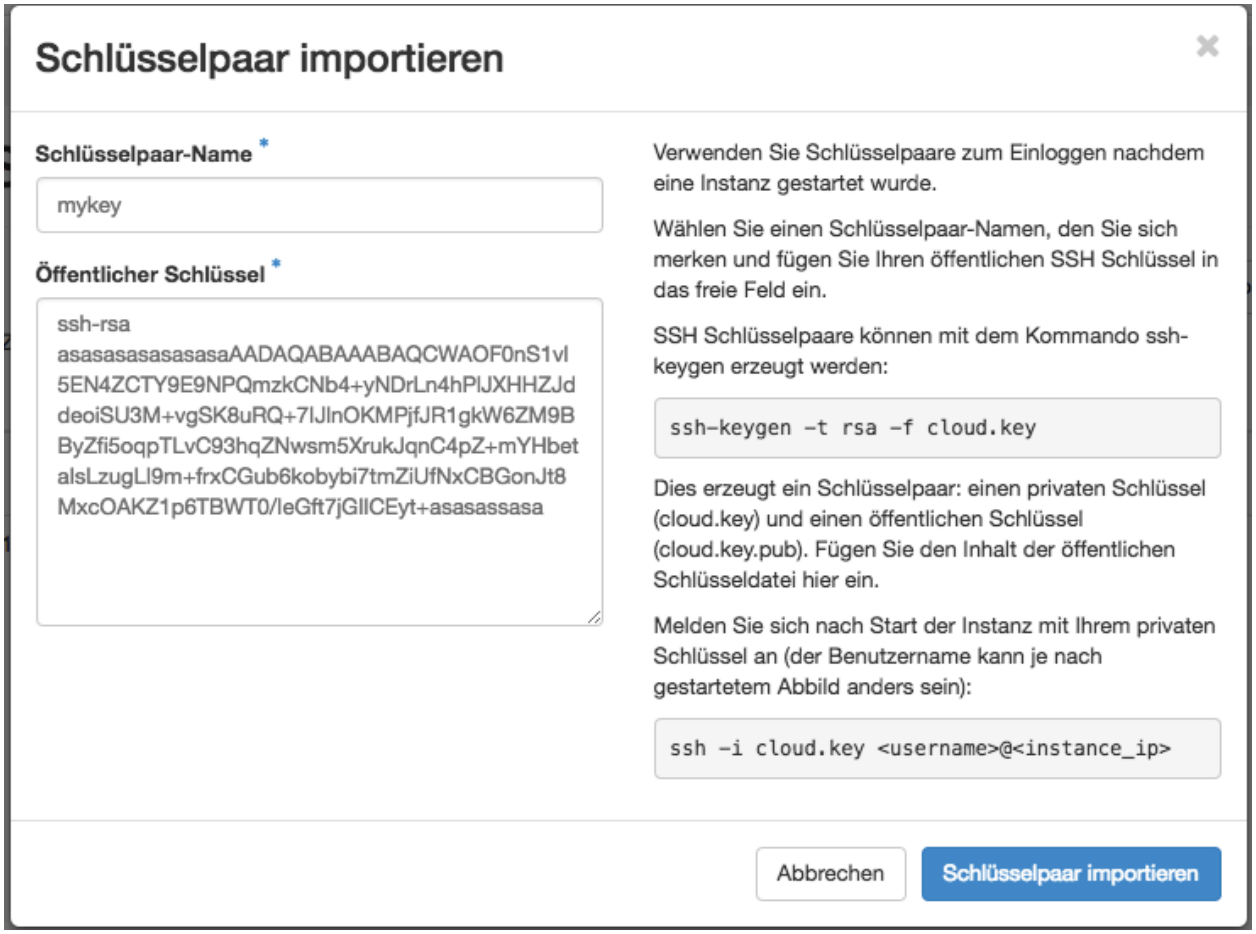
Run an instance through Horizon Dashboard

Before running an instance we need to create a key pair, or if we have already one key pair, we can import our existing public key through the dashboard:

To do so, navigate in Horizon to Projekt → Compute → Schlüsselpaare (Keypair) and click on “Schlüsselpaar importieren” (import keypair) and provide a name “mykey” and your public key.

Alternatively you can create a new keypair by clicking on:

(create keypair and download the private key):



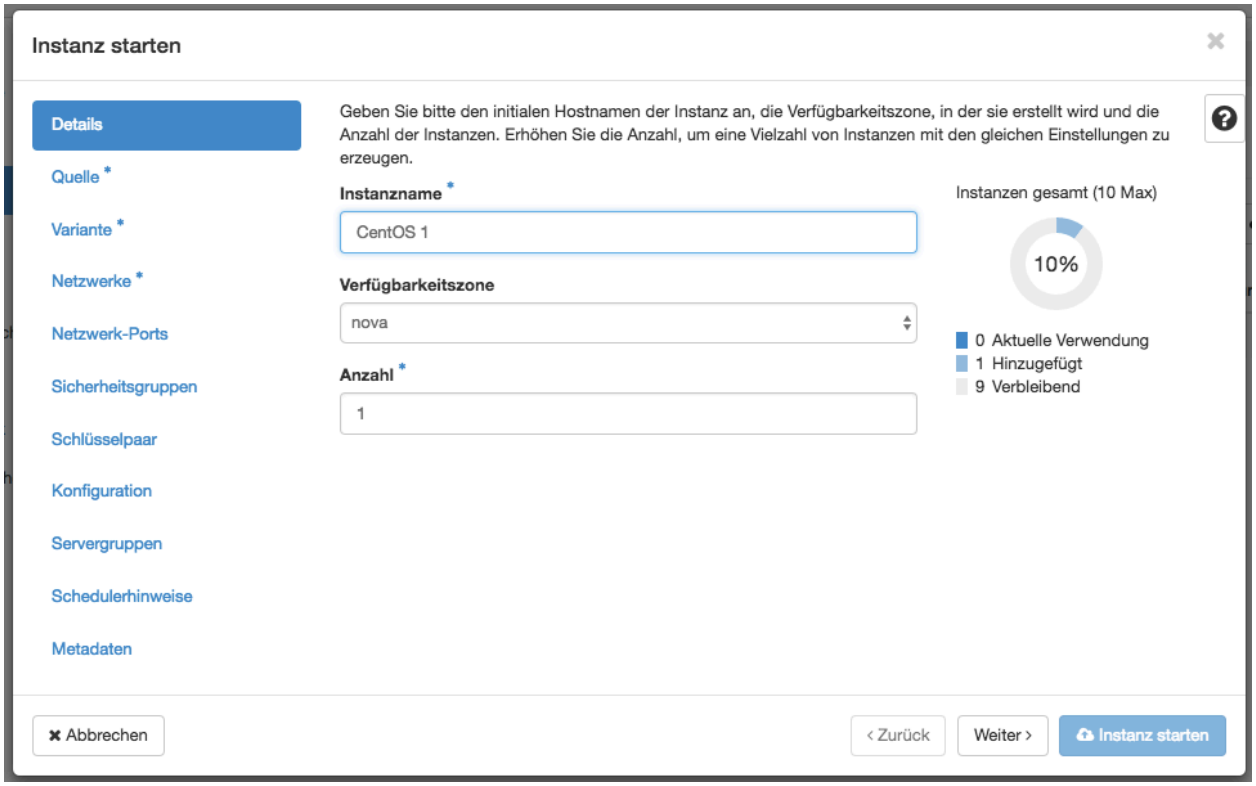
Finally we can start a new instance from our CentOS QCOW2 Image by navigating to:

`Projekt → Compute → Instanzen`

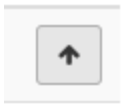
And click on:

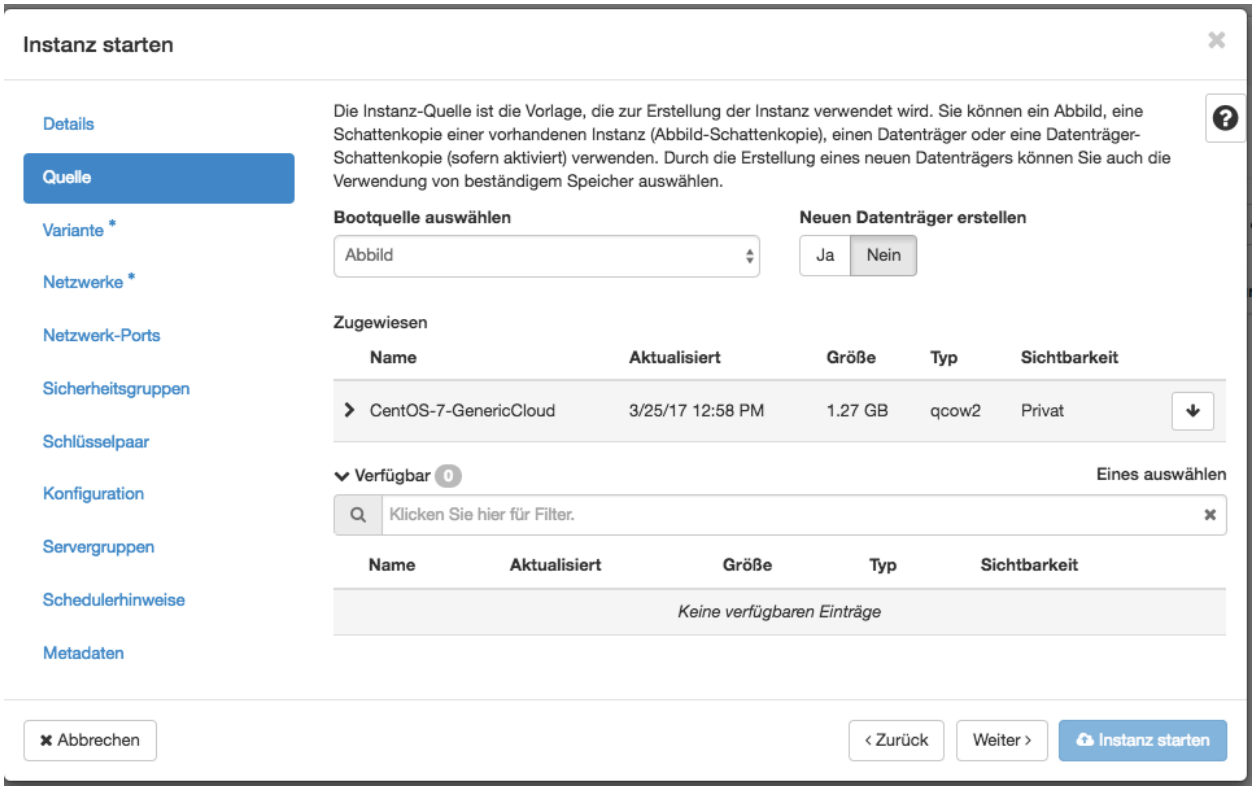
And follow the next steps provided by the wizard:

Provide a name for the instance (e.g. CentOS 1)

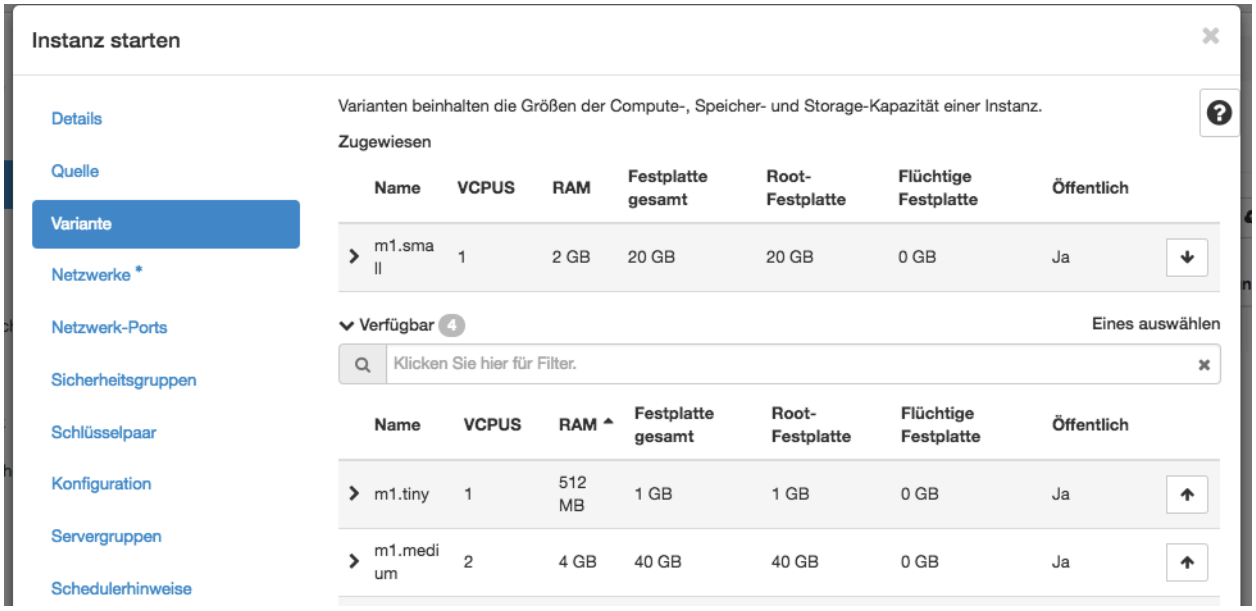


Select the image (Abbild) and select “Nein” for “Neuen Datenträger erstellen” (create new volume), unless the instance creation will fail! Add the image with the Up arrow under the available (Verfügbar) images:

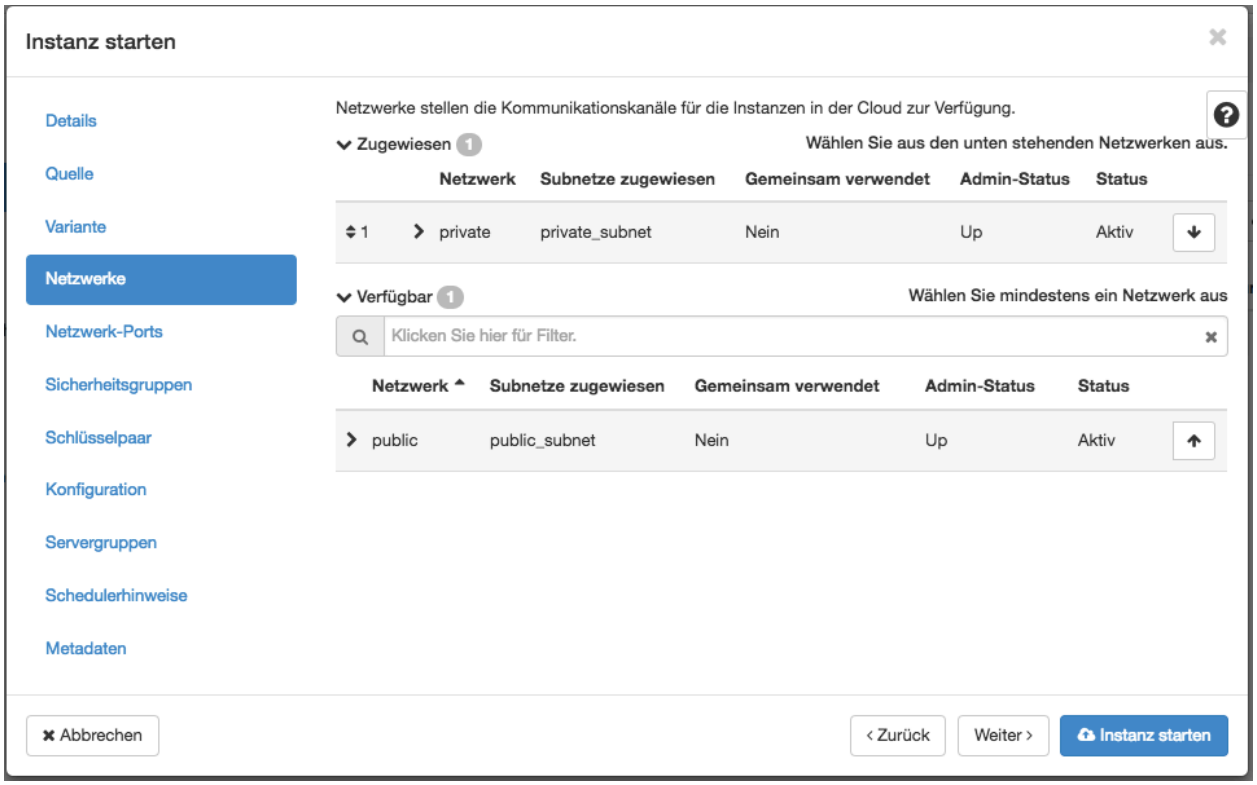




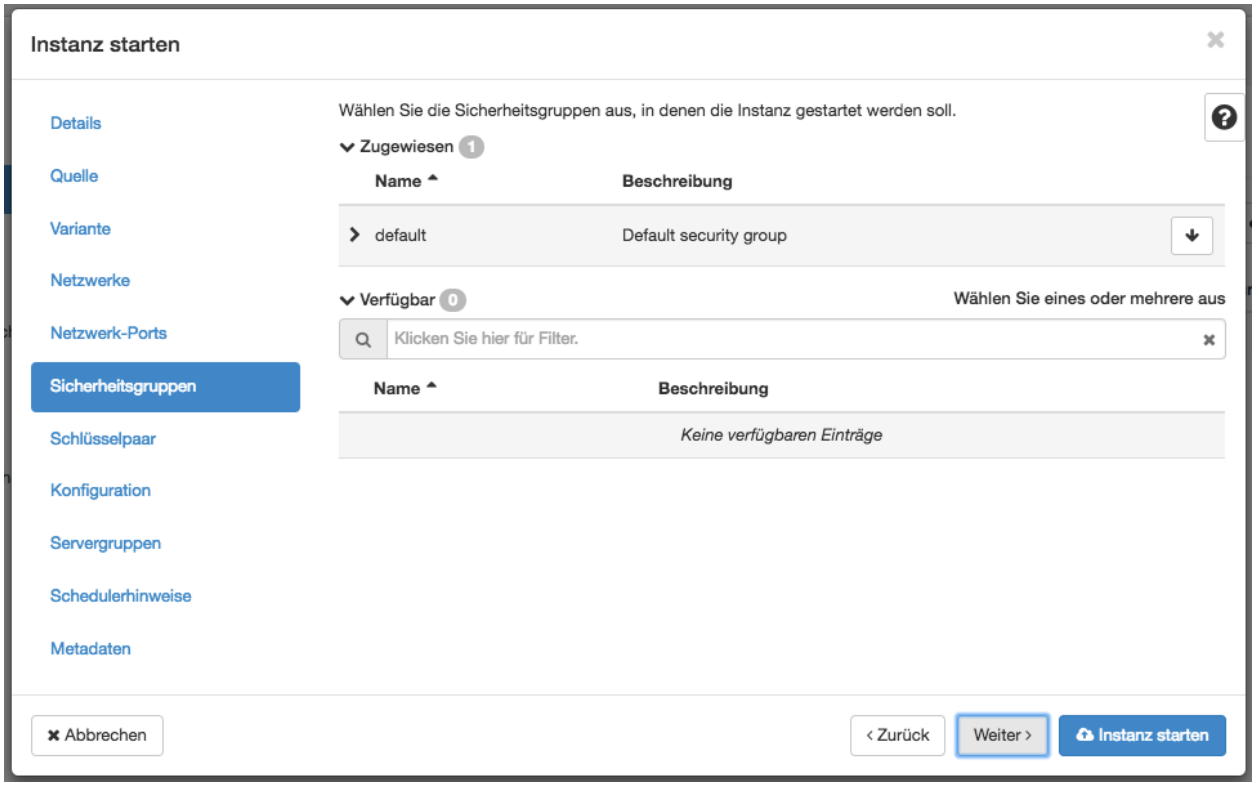
Select the m1.small flavor type (Variante), use the up arrow:



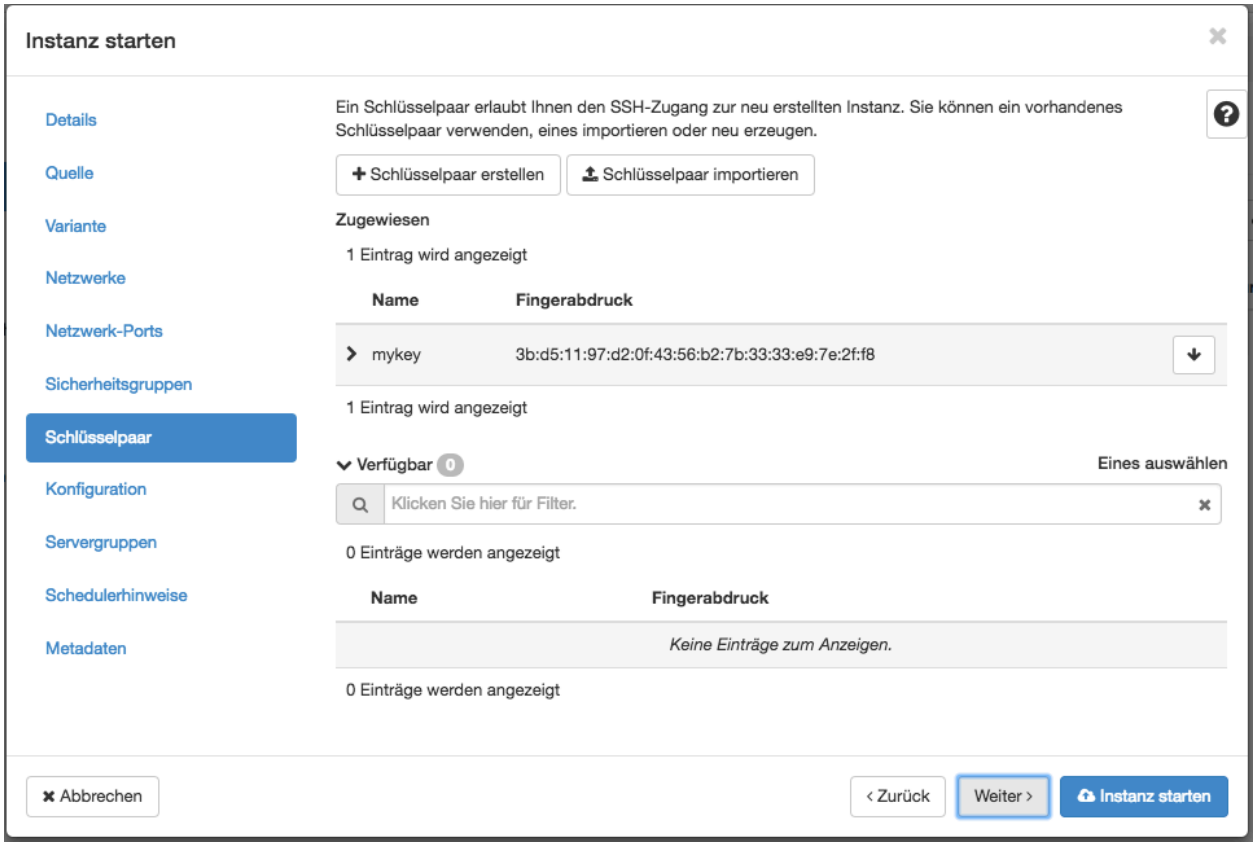
Assign the private network to the instance to use:



Jump over Network ports to the Security Group (Sicherheitsgruppen), the Default Security Group is selected by default (since this is the only security group which was created during the installation):

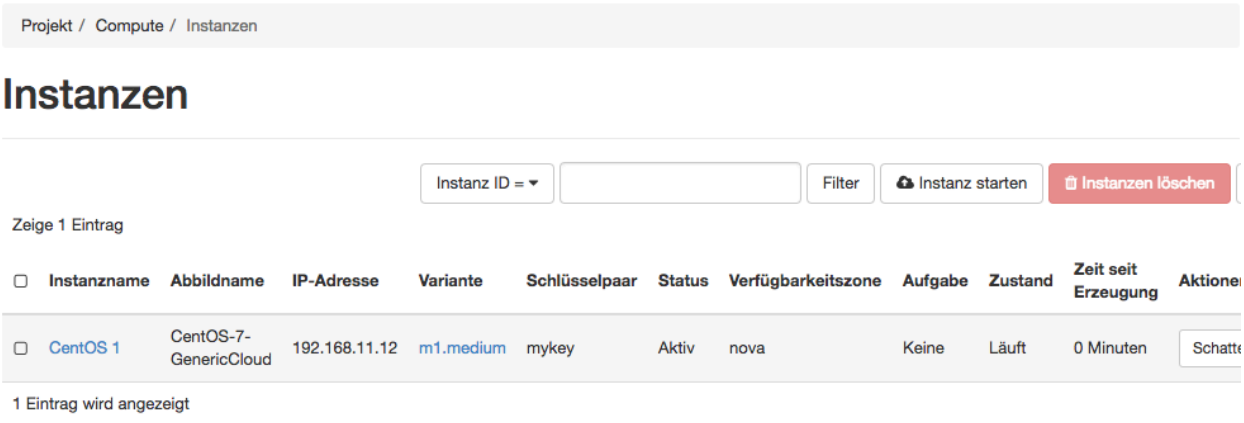


Our key will be selected by default as well:

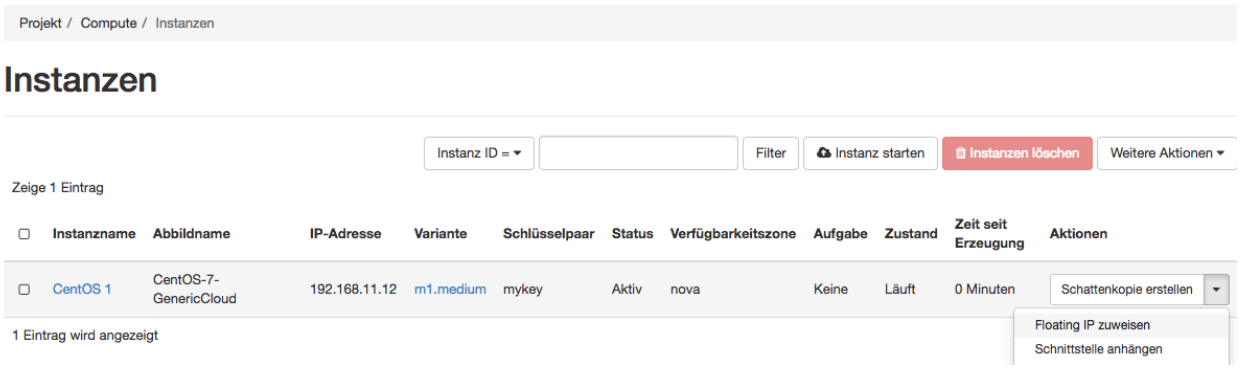


We can jump over the next steps and launch our instance by clicking on:

After few seconds we'll see that our instance is running:

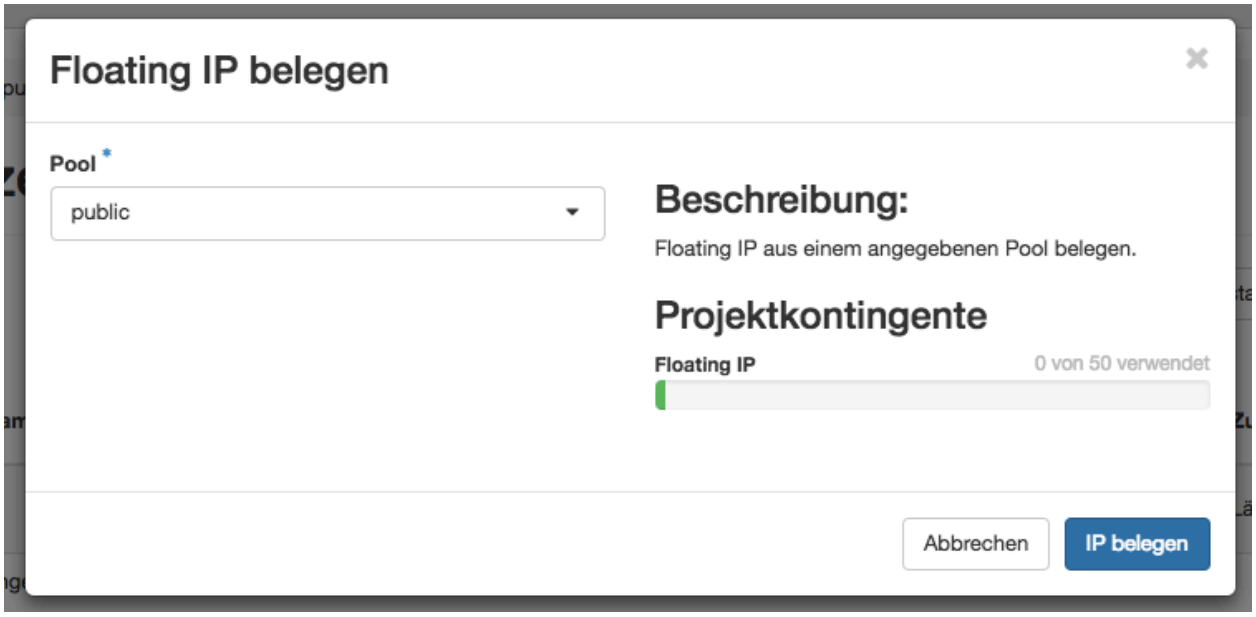


Now we can click on “Floating IP zuweisen” (assign floating IP)

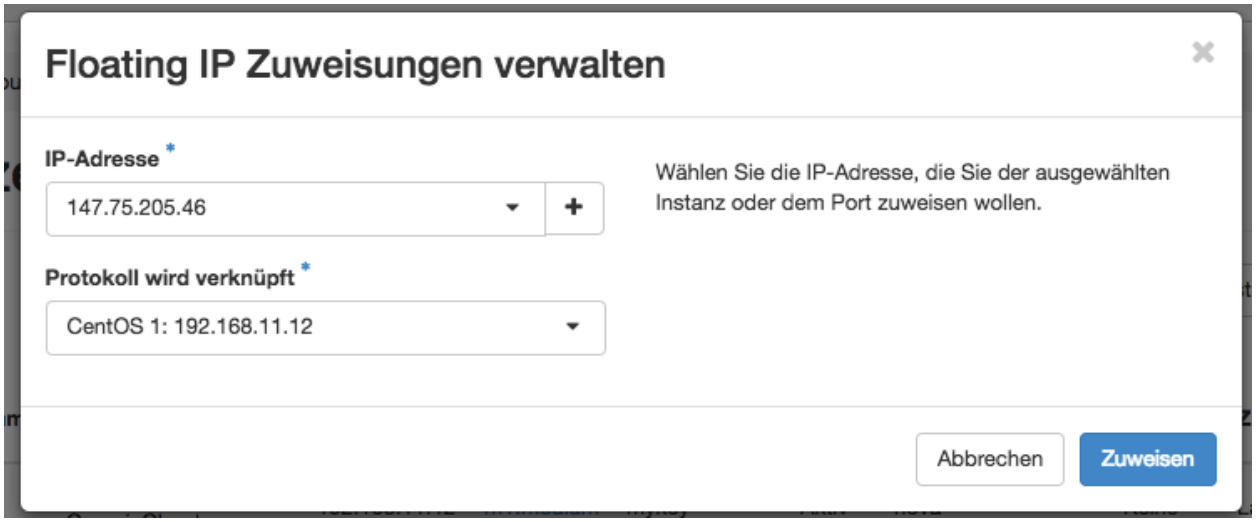


Click on the  icon:

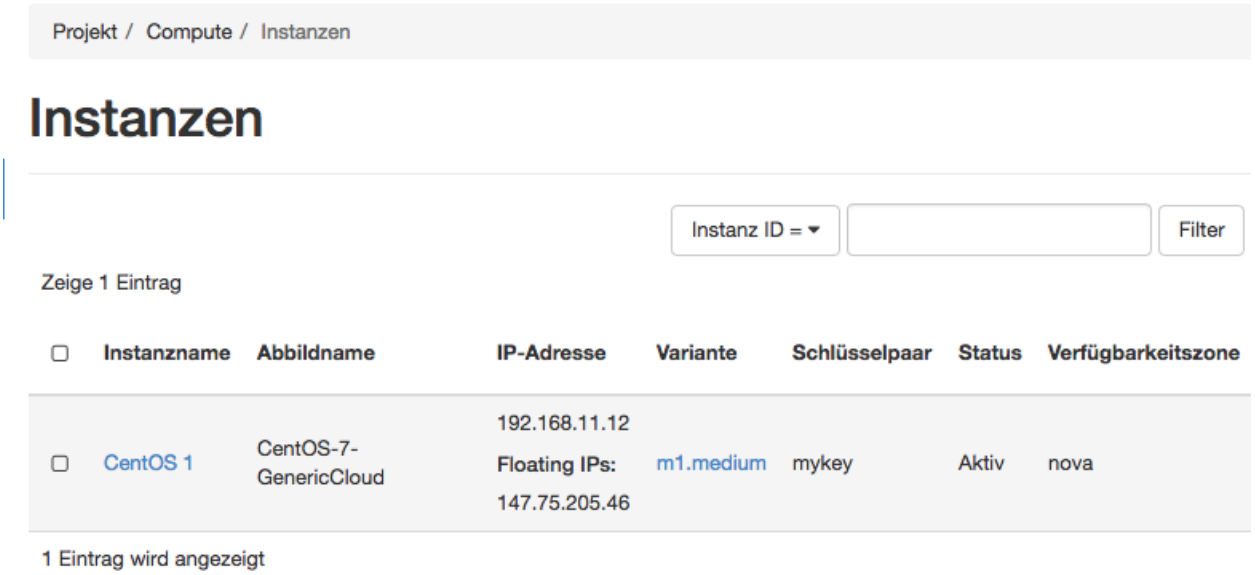
Click on “IP belegen” (assign IP):



A floating IP will be created from the pool which we defined during the public net creation.
And finally click on “Zuweisen” (assign):

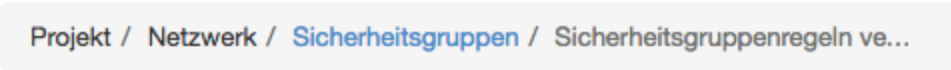


Now we can see that the floating IP is assigned to our instance:

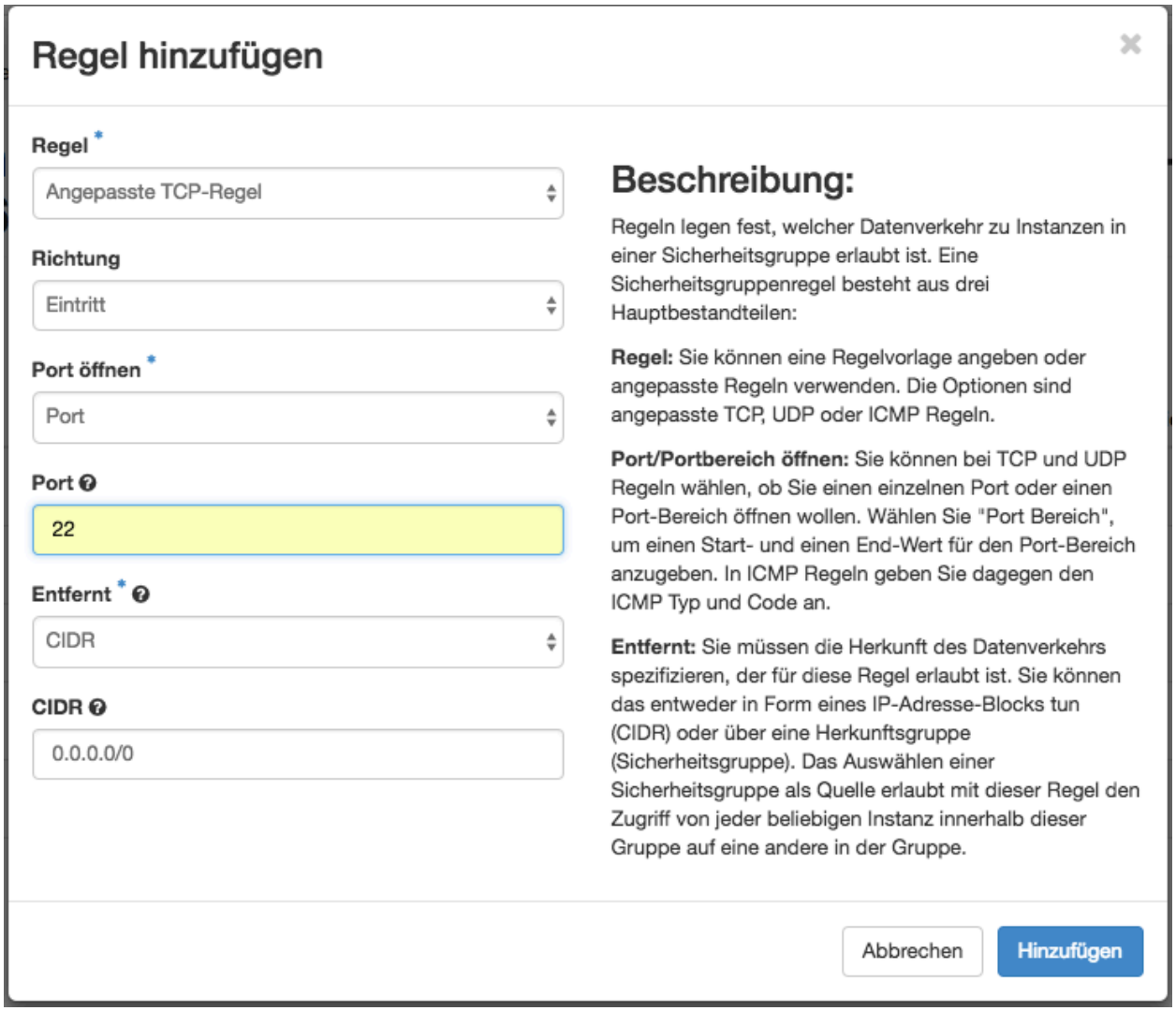


Note: since on Packet.net we don't have any floating IPs for our training, we won't be able to ssh into our instance, but we can achieve this through the private IP of our instance as follow.

First we need to allow SSH through port 22 by defining the ingress rule in our default security group by navigating to (Project → Network → Security Groups):



And add a new rule as follow:



Now on the controller we can examine with “ip netns” the dhcp namespace:

[root@controller-arash ~(keystone_admin)]# **ip netns**
qrouter-1cbc5b78-eb54-417f-a174-bcd229316345
qdhcp-8372e1a3-5c7c-4a93-b697-2ce9b90a3163

And ssh into the instance with our private key “mykey.pem” and the default user “centos”:

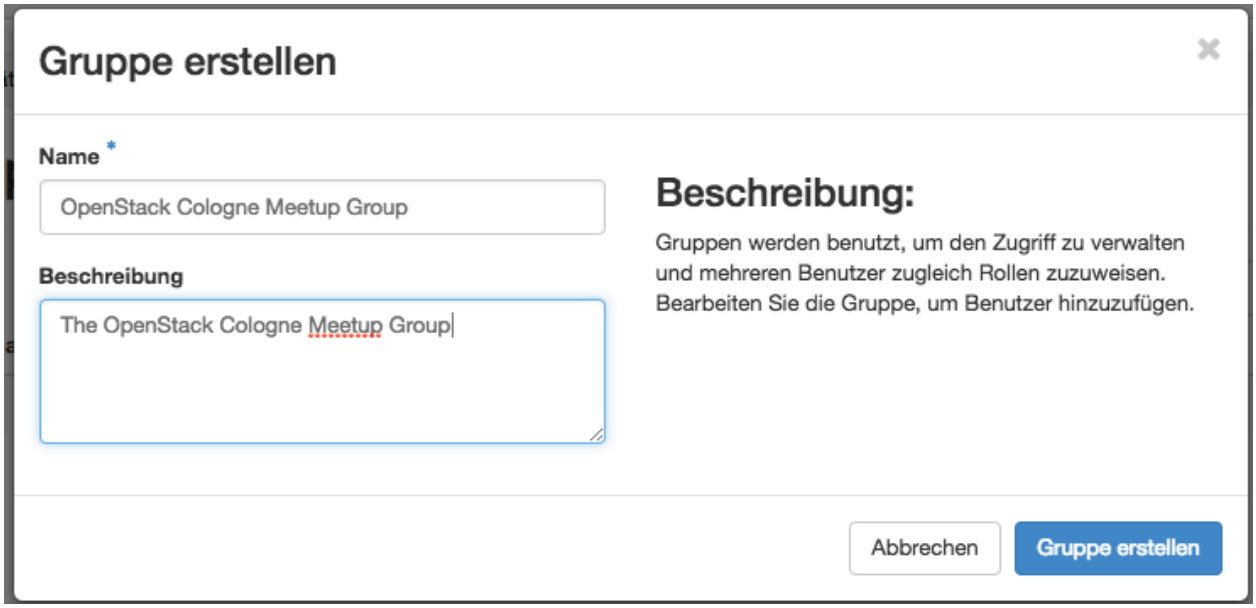
[root@controller-arash ~(keystone_admin)]# **ip netns exec qdhcp-8372e1a3-5c7c-4a93-b697-2ce9b90a3163 ssh -i mykey.pem centos@192.168.11.12**

Last login: Sat Mar 25 13:16:03 2017 from 192.168.11.2
[centos@centos-1 ~]$;)))

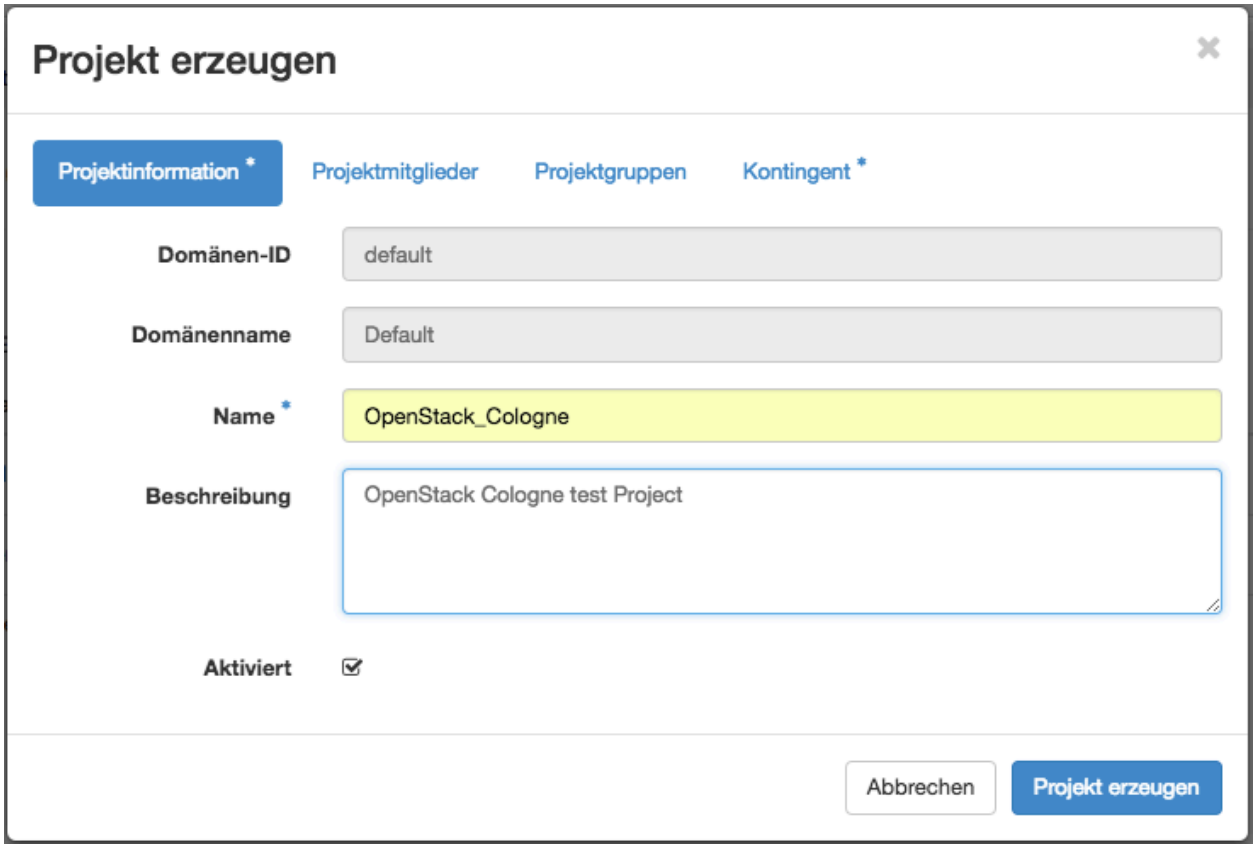
As you can see our VM is happy ;-)

Create a new Group, Project (tenant) and User and run a new instance for this user

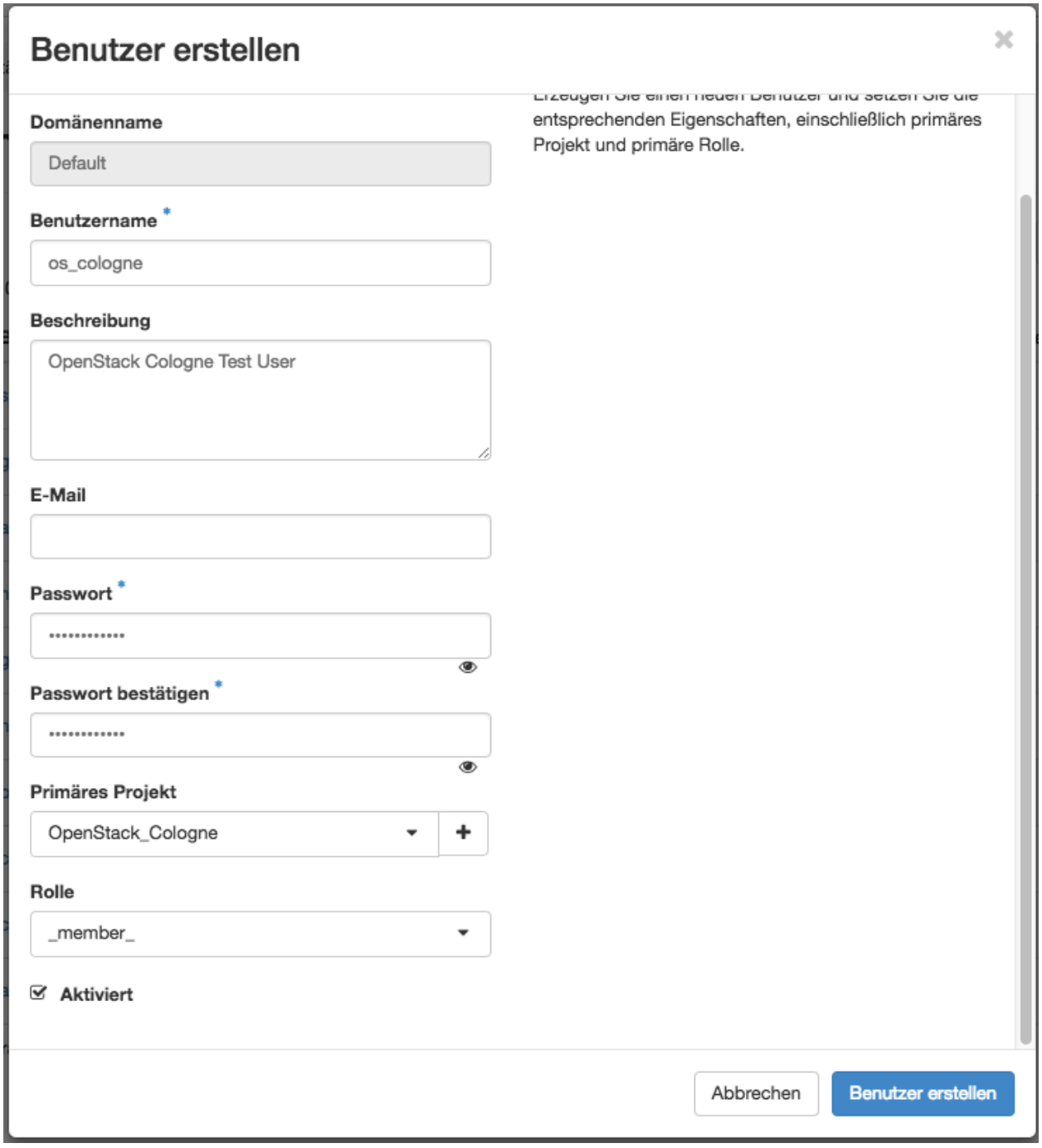
In Horizon navigate to Identität → Gruppen (Identity → Groups) and create a group:



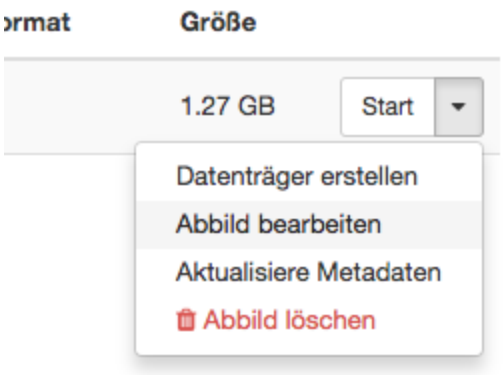
Create a project:



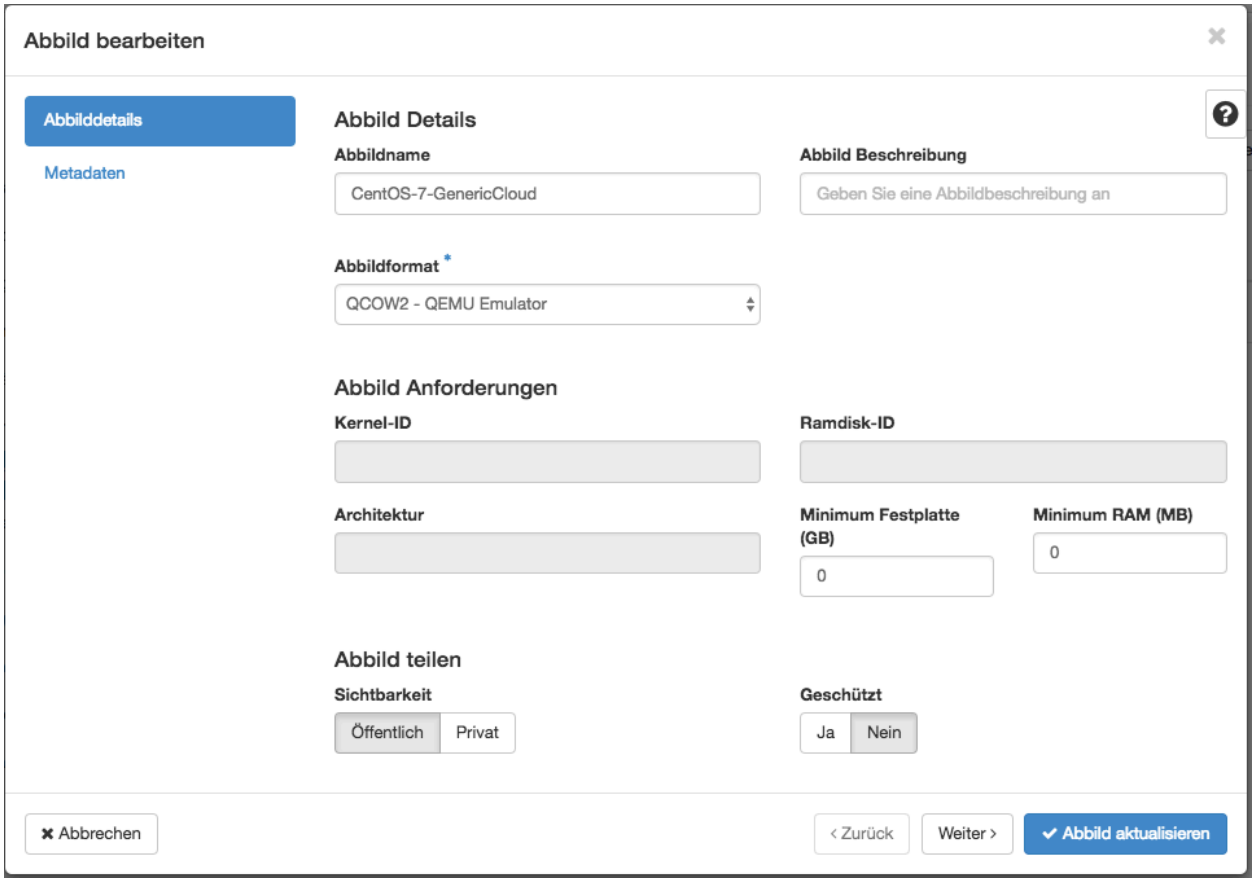
Create a Test User os_cologne and assign the none-admin member role to the user in the project OpenStack Cologne:



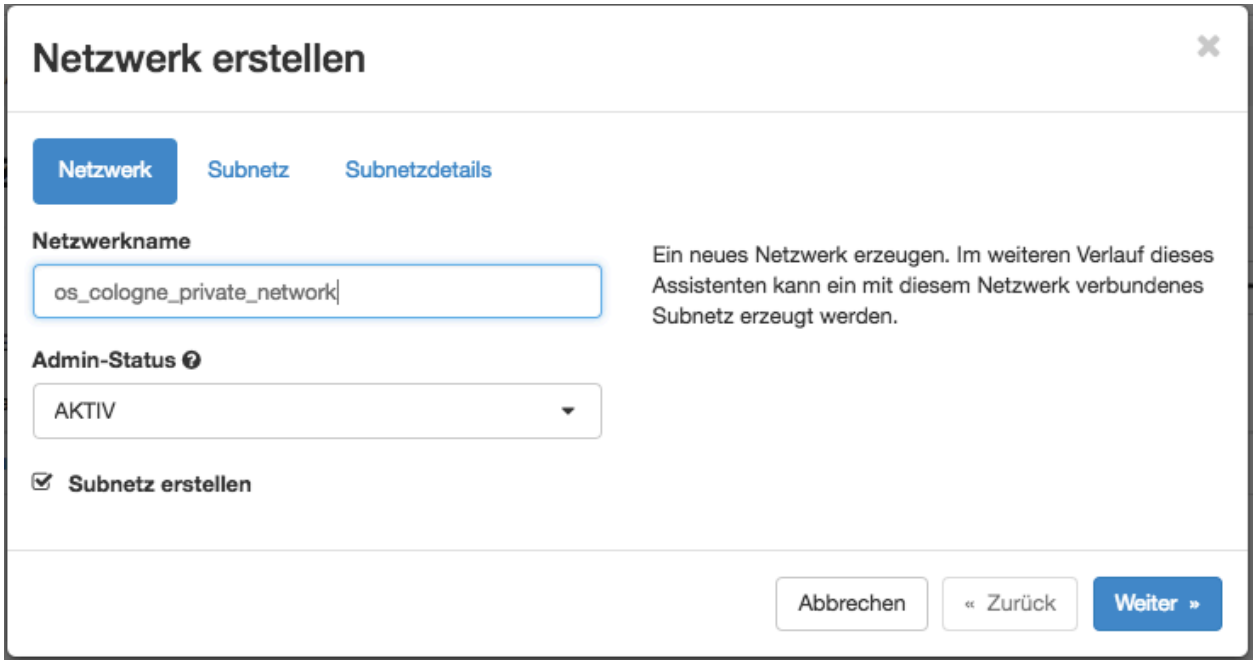
Make the CentOS image public, so that our user os_cologne with the member role can use this image, to do so, please navigate to Administrator → Abbild (images) and edit the property of the image (Abbild bearbeiten):



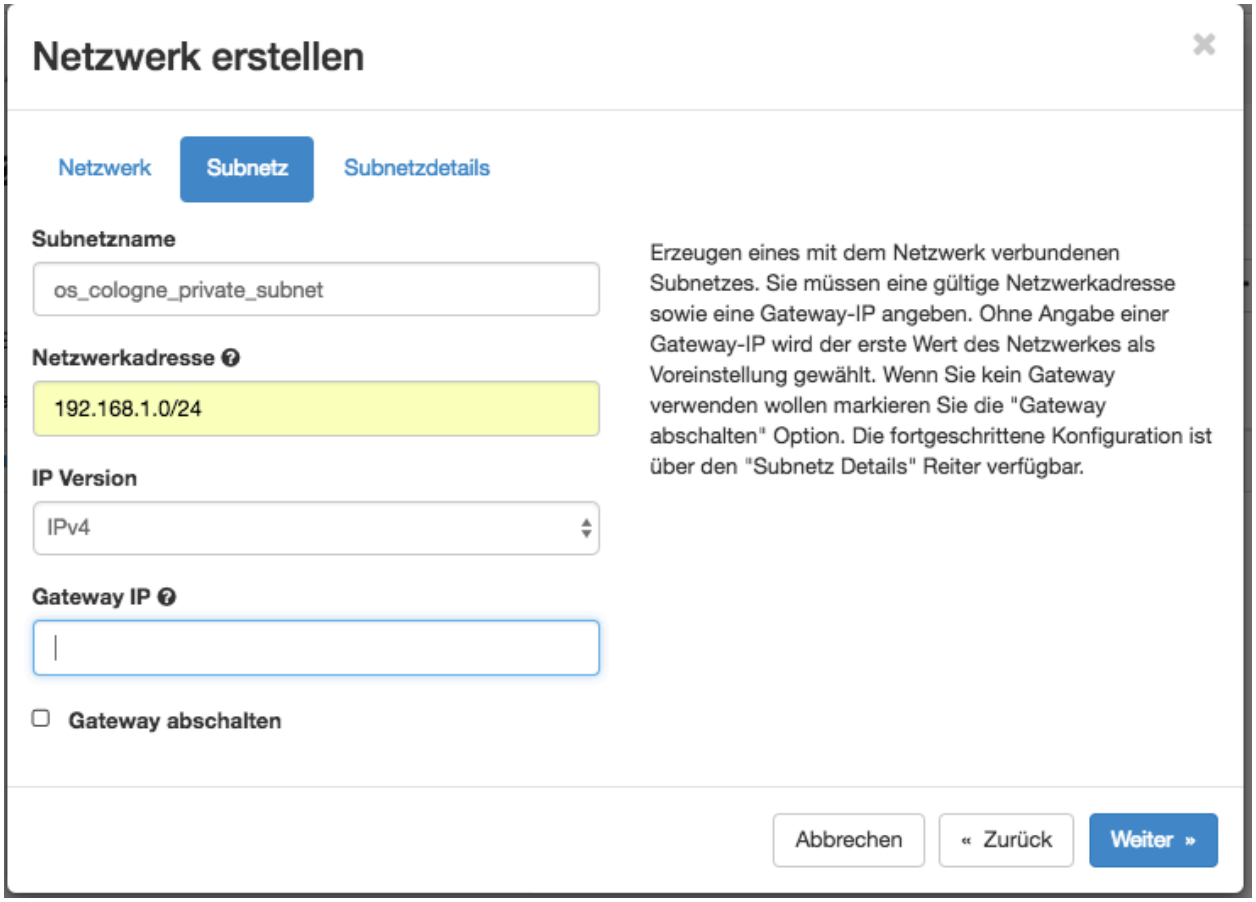
And share the image as public (öffentlich):



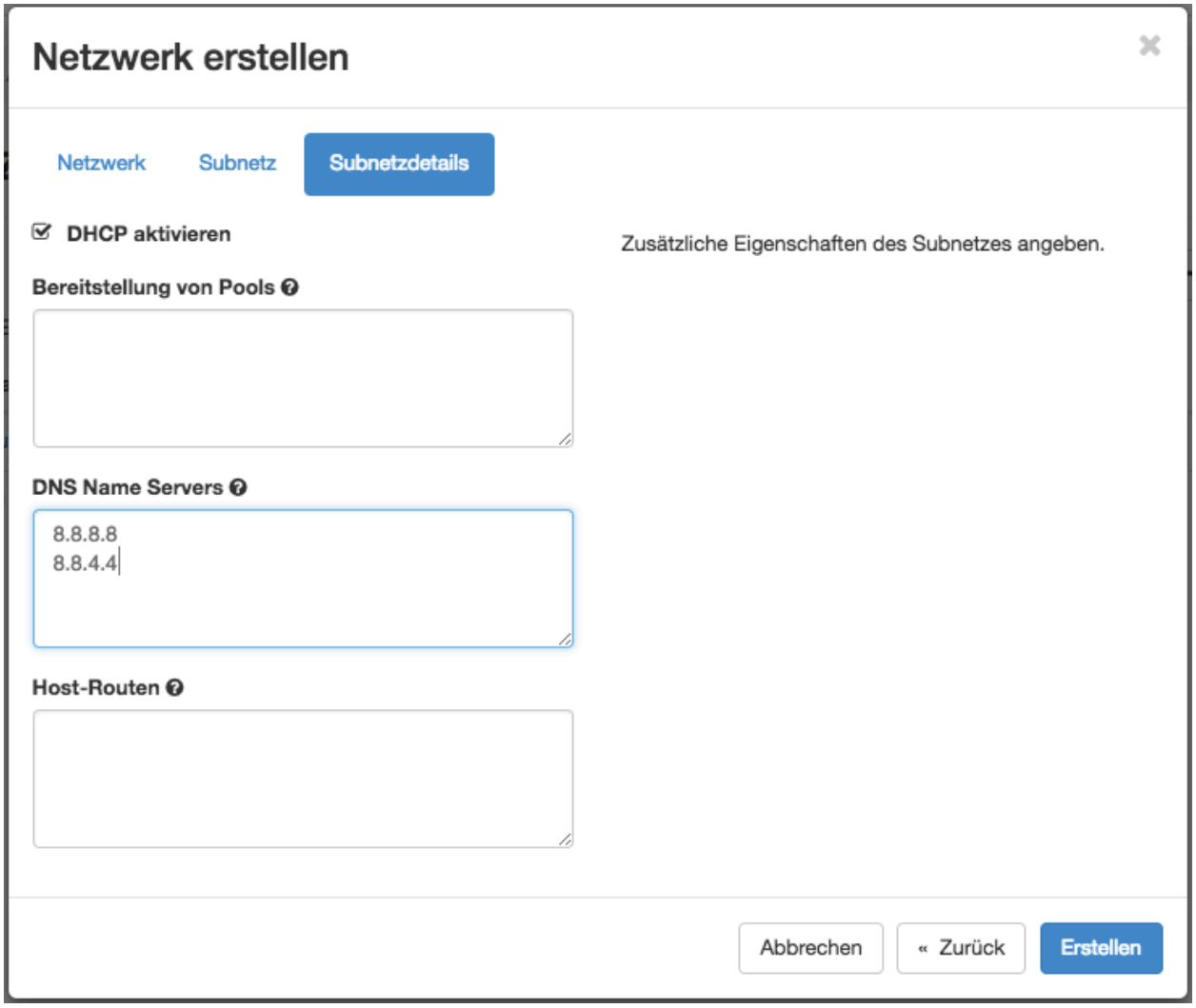
Logout and login as os_cologne user and create a network:



Create a subnet:

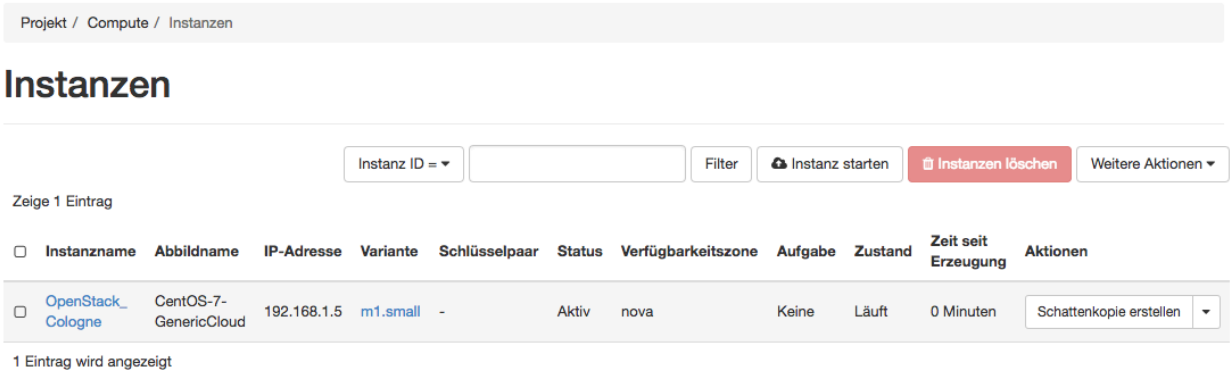


Let DHCP activated and only provide the Nameservers and create the private network:

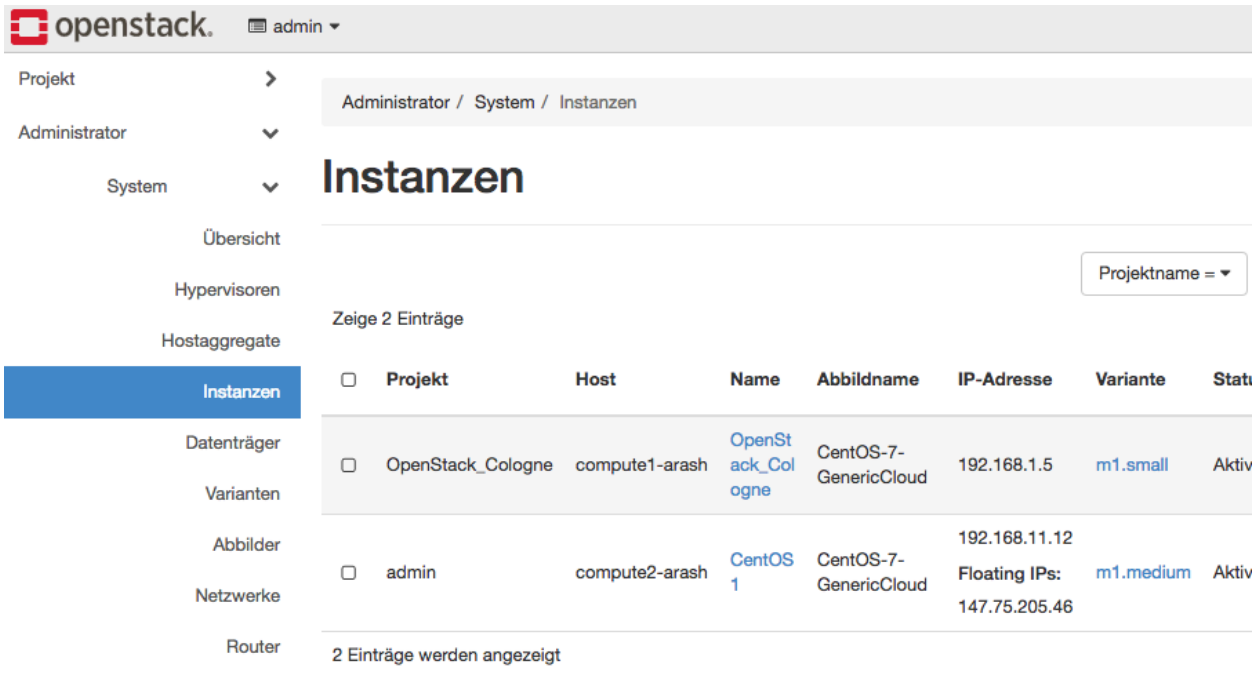


Please create an instance as we did for the admin user.

You should be able to use the same CentOS image and use your own private network, the instance shall be created successfully:



Login again as the admin user and verify if the 2 instances are created on different compute nodes:



Related article:

If you'd like to enjoy Kubernetes with OpenShift Origin on OpenStack, please refer to this doc https://goo.gl/gbnMDD

Questions?
If you've any questions or if you find some gotchas, please drop me a PM @cloudsskyone

Thanks!
Arash