Do not quote. Work in progress!

Fedora Server Edition on rented root server hardware

The example of Hetzner

Abstract (✔)

Installing a standalone server on rented hardware in an external data center is suitable for small and medium-sized companies and organizations, as well as for self-employed entrepreneurs and individuals. However, the installation is subject to some restrictions and peculiarities that arise from the accommodation in a remote and inaccessible data center as well as from specific security measures of the providers. There are many descriptions for other mainstream distributions. For Fedora/CentOS/RHEL, the information is sparse and often outdated. This article uses the Hetzner data center as a blueprint to show how these functions can be accomplished using Fedora resources only. Other providers, as e.g. OVH, Strato, zap hosting, you name it, have the same issues in principle, even if the solutions differ slightly.

1 Why Rent a Dedicated Server? (✔)

Even though cloud services, virtual servers, SAAS are on everyone's lips, complete control over hardware and software is attractive or even indispensable for companies and also private individuals in many cases. Data protection and confidentiality considerations, special software needs, or simply the joy of experimentation are reasons for doing so.

Rental servers in a data centre offer the advantage of a professional set-up and, above all, a *technically optimal internet connection*, as only large corporations can afford it on one's own premises. Various companies offer this kind of product. This guide applies to the *Hetzner* offering, a large European company. Tey offers among others quite affordable options through its "Server Auction". In principle, the instructions can be adopted to offers from other providers.

2 Planning ahead (✔)

The objective is to install and maintain a server that is not directly accessible locally. Various publicly accessible services are to be offered securely and reliably. These include standard services such as mail and web applications, but also sector-specific software.

The most important guideline is to carefully seal off access to the server as far as possible. It must be prevented by all means that it can be compromised or even hijacked. Due to the remote, not directly accessible location, a failure of unobstructed access is comparatively laborious to remedy and thus "expensive". Appropriate measures are:

- Access to the server exclusively via ssh and key-based identification
- The server operates exclusively as a "host" for public services without offering them directly
- All services such as mail or web are encapsulated and run either in virtual machines or a container (frontend as guest system)

- No accounts or only a few accounts required for system maintenance are created on the host itself. A guest system accommodates all other accounts..
- The server may perform all internal processing of data without direct access from the public network (backend). This is particularly useful for I/O-intensive applications such as a database for performance reasons.
- Communication between VMs, containers and backend takes place via a protected internal, non-public network.

2.1 Storage (✔)

The aforementioned provisions are largely "best practice" for any server installation today, but become even more important when outsourcing off premise.

Hetzner rental servers usually two disks by default. A disk fault is rare, but nevertheless it is advisable to run them in a RAID array. The strict separation of system and user data under this condition also serves to make administration as simple as possible. The system administrator must be able to maintain the system area, i.e. the operating system including installed utility programs and software such as a database system, completely independently of the storage of user data. System maintenance must not jeopardise user data under any circumstances. If necessary, it must be possible to unmount user data.

For precisely this reason, Fedora Server creates by default a small /boot partition and in the remaining area a partition with a volume group (VG). Therein, it creates a logical volume of approx. 15 GB (the exact value depends on the disk capacity) for the operating system and its software. The other available space remains free for the creation of logical volumes (LVs) for user data, which are mounted in the appropriate positions in the directory tree of the system area (details later).

We will go a little further and create another small partition and VG for the operating system in addition to the partition for /boot (sysvg approx. 30 GiB). Therein, we create LVs for the root directory tree and its runtime environment. We leave some free space for disposal as needed. The remaining area of the hard disk is filled by a large partition and VG for user data (usrvg). Similar to the standard partition, all user data lives as LVs in usrvg mounted in corresponding directories of the system area. This is the maximum possible separation of system and user data unless you have some external storage or multiple disks available.

2.2 Network (✔)

It is one peculiarity of the Hetzner infrastructure that, although each server is technically located in a subnet of different sizes, the network connection component prevents a server from communicating directly with its neighbors in the subnet. The reason for this is to prevent a system administrator from accidentally “hijacking” a foreign IP address due to a typo. If a sysadmin fails to take this into account and sets up a standard network configuration, it can result in surprising network blockages. Hetzner uses various network sizes, so the larger the subnet used, the greater the likelihood of this happening.

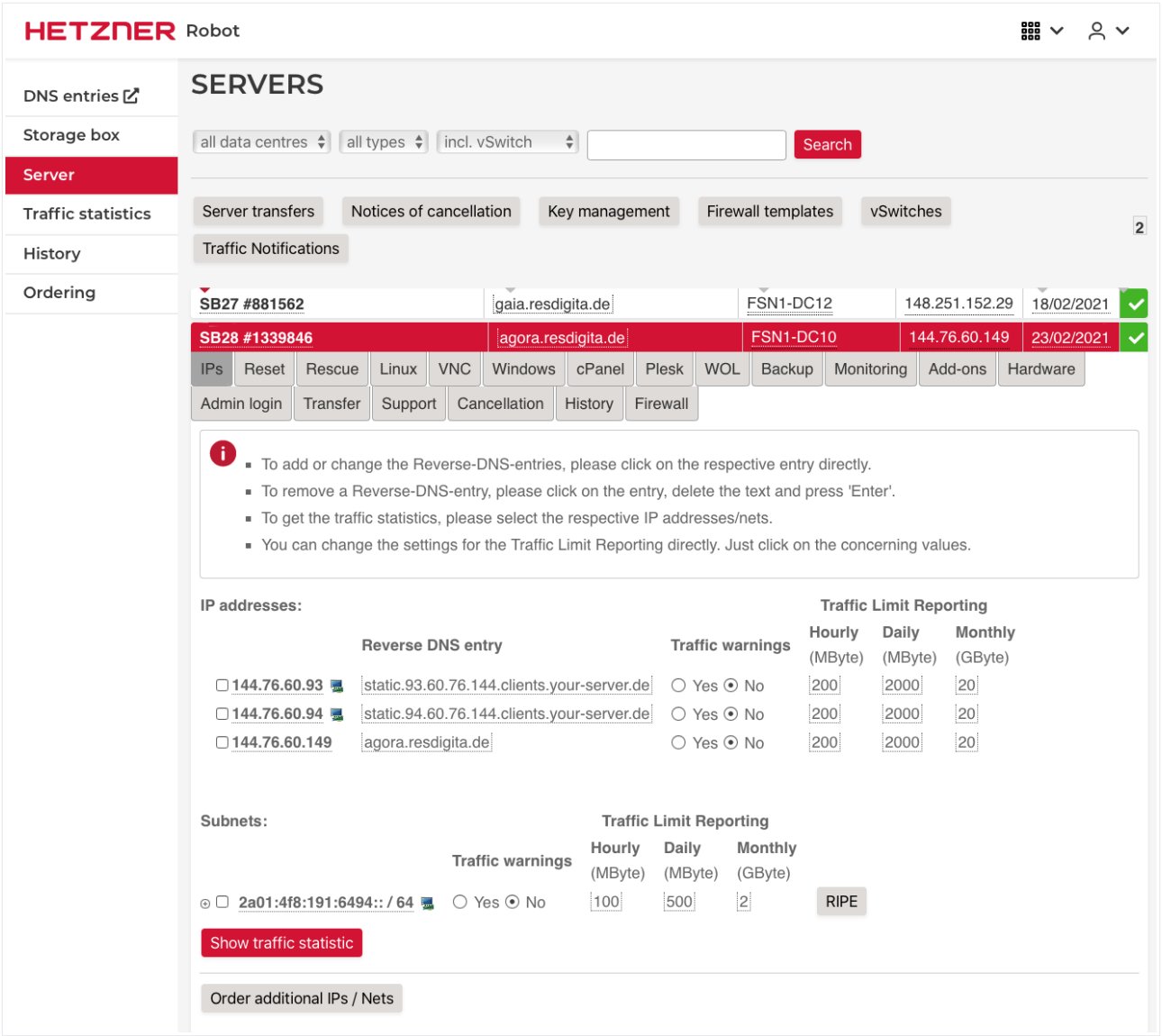
To prevent any surprises, Hetzner configures a peer-to-peer connection to the router in its customer server setups. System administrators who install and configure their own server must take care of this themselves.

The other peculiarity of the Hetzner infrastructure is that it routes all IP addresses assigned to a server to its physical Ethernet interface, i.e. always to the same MAC address. This is a problem specifically for virtual machines with an IPv4 and a IPv6 address, which are connected via a virtual bridge in the server. For IPv4, Hetzner could provide different Mac addresses for each IPv4 address, so you could configure a conventional bridge which needs a unique MAC address for each IP address. However, for IPv6 Hetzner does not provide separate Mac addresses. So this option is not viable for a server that is supposed to be universally accessible to the public and therefore requires a dual stack configuration nowadays.

The solution is a special bridge that routes based on IP addresses, a routing bridge (brouter). The configuration is a bit more complex. The basis is a bridge that does not integrate the server network interface as a slave (and take over its IP address), but instead receives its own IP address. With forwarding enabled, incoming other IPs are forwarded from the interface to the bridge. For an IPv4 address, the virtual machines connect to this bridge via a peer-to-peer connection. IPv6 handles this out of the box and requires no special configuration.

3 Preparations

At Hetzner, servers are ordered fully automatically via the web interface. It is important to deselect the setup included in many offers. Within minutes, you will receive IP addresses and access data by email, and the server will be visible in your account and immediately available. An IPv4 address



and a /64 IPv6 subnet are included as standard. As explained above, if you plan to offer publicly available services in virtual machines with both IPv4 and IPv6 (dual stack, what you definitely want these days), you need more IPv4 addresses. You have to order these separately, either various single IP addresses or an IPv4 subnet, using the order button at the bottom. You get those IPs nearly instantly and can use them immediately. A subnet has no advantages within the server, but broadcast and network IPs still cost the same as any other IP. Here we will use additional single IP addresses. On the screenshot above you see 2 single IPs added.

For IPv6, the host receives a /64 subnet range. The host address is fixed u.v.w.x::2.

Ensure that your rented server's IPv4 and IPv6 addresses are known in the DNS. Carefully note down all IP addresses details, host IPs, IPv4 subnet range, gateway IPs, and name server IPs. You will need these details in several individual steps.

4 Check the server

Hetzner delivers a server without setup in „Rescue" mode. The server boots with a temporary minimal operating system based on Debian. Hetzner sends the address and a root password by email.

If you want to install a server which is already configured in some way, you have to activate the rescue mode yourself. In the Web interface select the „Rescue" tab, select Linux and the keyboard layout, and activate the rescue mode. You get a new form with the auto-generated root password. Then select the „Reset" tab, select „Execute an automatic hardware reset", and submit the form. Wait for a mail with the information about the Reset done

Then you can access your server, using the user name "root" and the following password: wRVx9RuMeXLhtG

Log in to the server using the provided access data.

```
[…]$ ssh root@${YOUR_SERVER_IP}
```

(a) Check the discs

Ensure, the disks are empty without any partition or partition table.

```
[…]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0   7:0    0  3.4G  1 loop
sda     8:0    0  1.8T  0 disk
sdb     8:16   0  1.8T  0 disk
[…]$ fdisk /dev/sda
…
Device does not contain a recognized partition table.
…
<q>
[…]$ fdisk /dev/sdb
…
Device does not contain a recognized partition table.
…
<q>
```

If there partitions or a partition table delete everything using dd

```
[…]$ dd if=/dev/zero of=/dev/sda bs=4096 status=progress
```

There is no need to write to the complete disk. Use <CTRL><c> after a some time to stop the process. You just need to overwrite the partition table.

(b) Check the systems boot mode

When partitioning the disks you need to know whether the system uses UEFI or BIOS boot mode. Determine the boot mode

```
[…]$ [ –d /sys/firmware/efi ] && echo UEFI || echo BIOS
BIOS
```

So it is a BIOS boot system and you need to configure a BiosBoot partition. Otherwise you would need an EFI partition.

5 Basic Installation

Hetzner offers various ways to install the operating system.

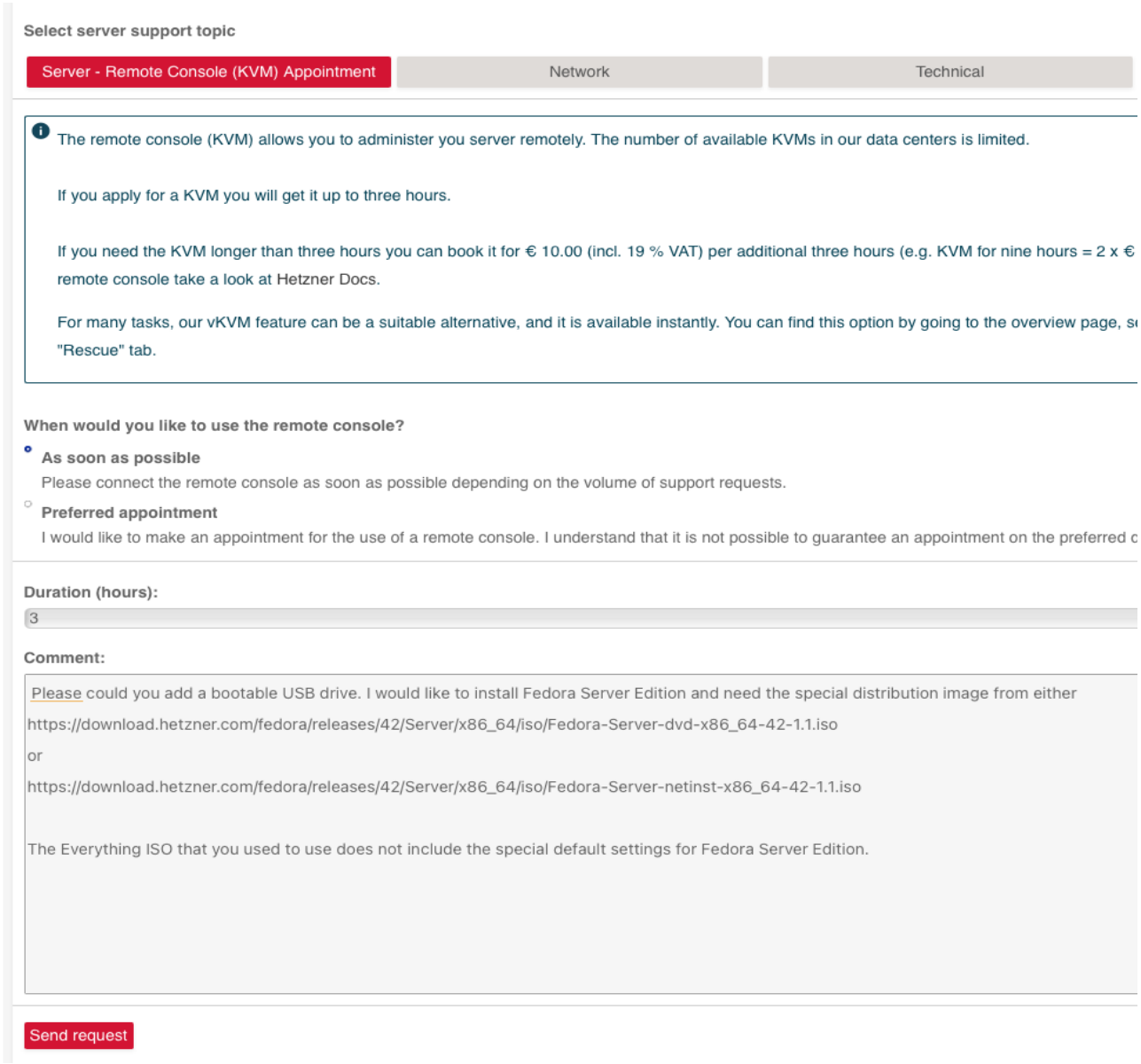
- Script-based Installation
  Configuration and Adaptability is very limited and omits all the features that make Fedora attractive. For Fedora it is useless. Currently it is not on offer but may be again in the future.
- VNC (Virtual Network Computing)
  Hetzner starts Anaconda on your server in a VNC session and sends the password by email. That would be very convenient, but almost never offers the latest version, and sometimes even only EOL versions. It used to use the everything media which provides a ‚Fedora Server‘ option. But it is not the Fedora Server Edition. It misses various Server Edition specific defaults.
- KVM remote console (Keyboard, Video and Mouse over-IP)
  Upon request, Hetzner Support will temporarily connect a virtual console that can be operated via a browser. This also includes a free choice of USB boot image. Any downloadable Fedora version is fine, even Rawhide. The service is free of charge, up to a connection time of 3 hours max. This is quite sufficient for an installation.

Using the KVM console is the better and recommended choice.

5.1 Working with the Hetzner KVM Console

(a) Order a KVM console

Click on „Support“ in the servers Web interface. In the Form., product type „Server“ and the corresponding server id are already selected. At the bottom click on „Remote Console“. The lower part of the screen expands to an order form.

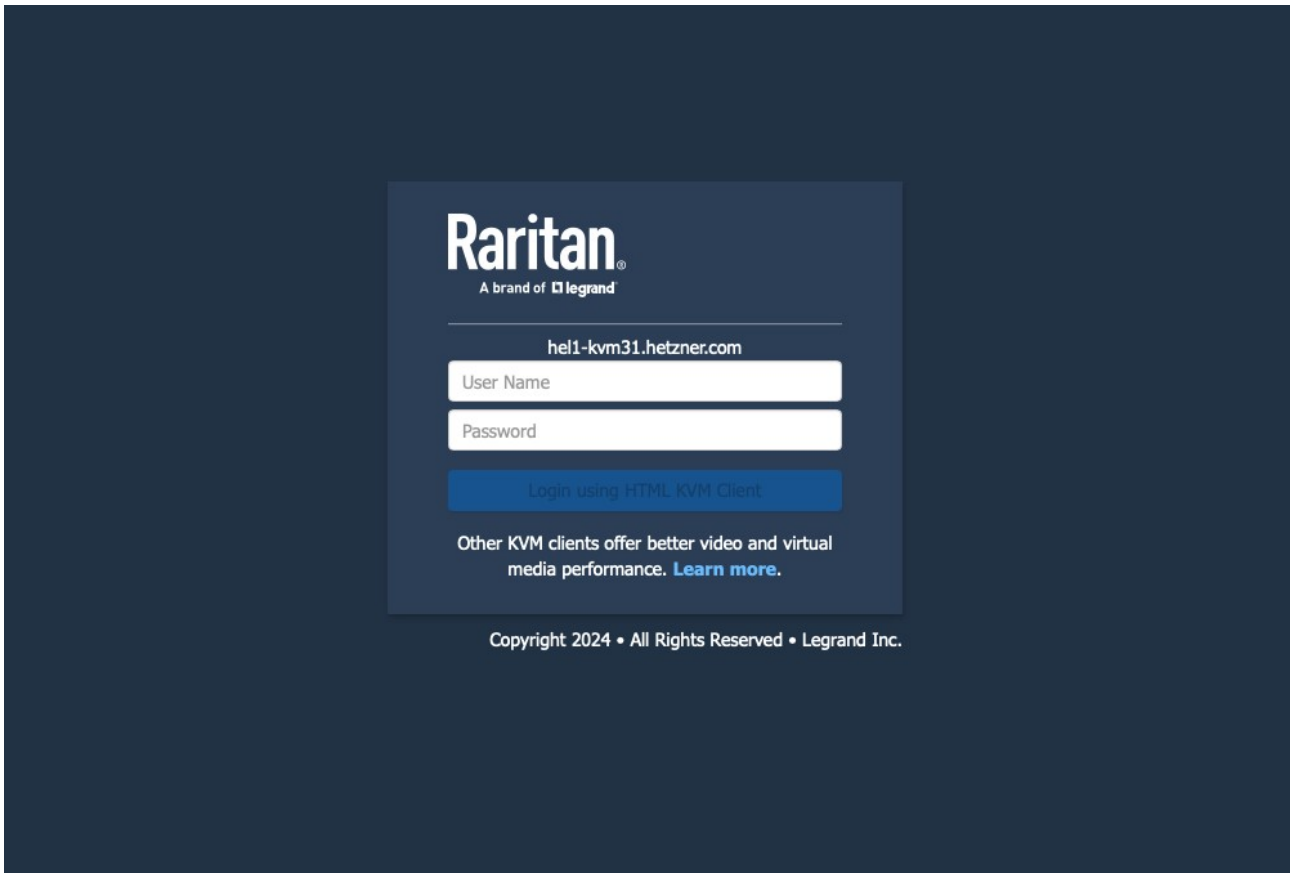


Fill in the form as needed and submit the request. The Hetzner mirror includes all Fedora versions. The access data are in the „server ready“ mail.

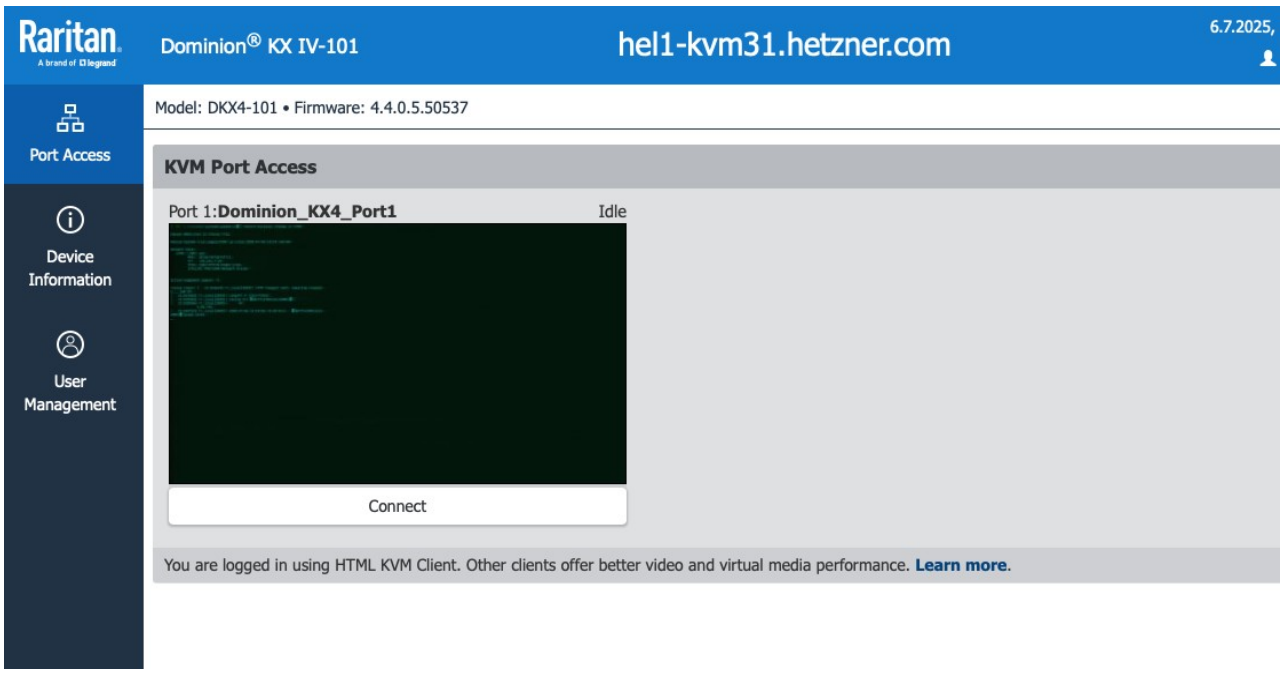
Hetzner provides 2 different KVM console models: Raritan and Lantronics Spider. The Lantronics seems not to work correctly with some browsers. Firefox works fine. With Raritan, you can choose between the html browser or a Java client, requiring Java 8. You get an email from Hetzner support when your KVM console is connected and ready.

(b) Using Raritan KVM console

1. Enter the provided KVM address from the 'remote console is ready' email into your WEB browser und you get the Login screen.

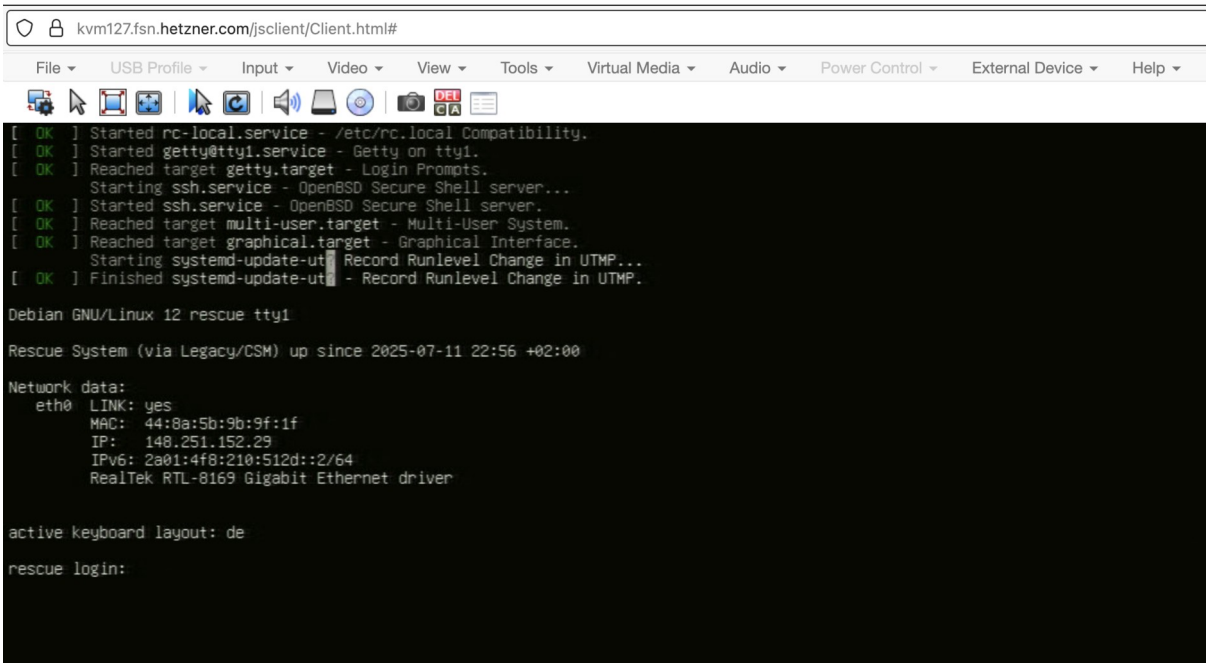


2. Login and you get the Raritan Console.

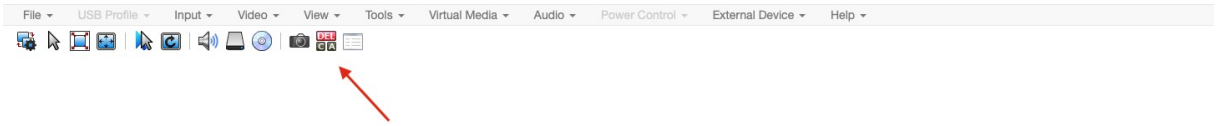


Select *Connect*.

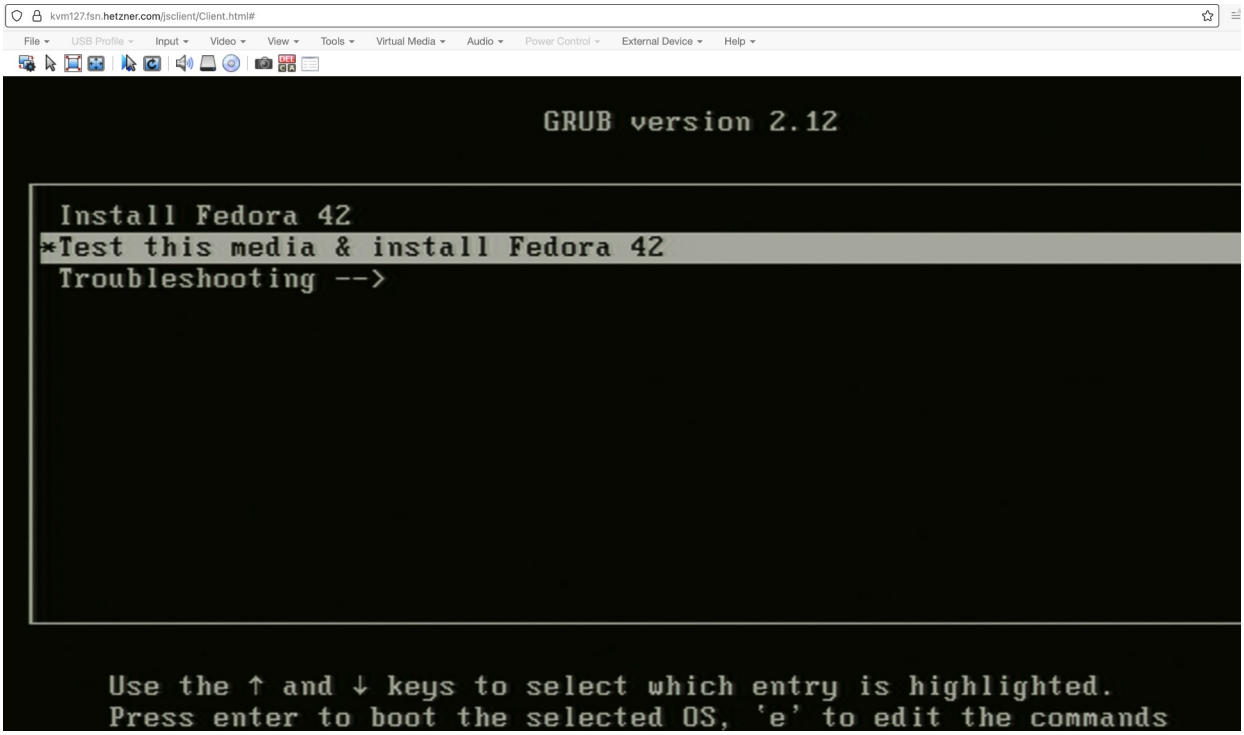
3. A new window opens showing the current state of your servers console. With a new server, Hetzner support will usually reboot your server when connecting the KVM. Then you will already see the familiar Anaconda Language selection screen. Otherwise you get the current terminal.



In this case, as in the example above, reboot the server using the button in the top icon bar.



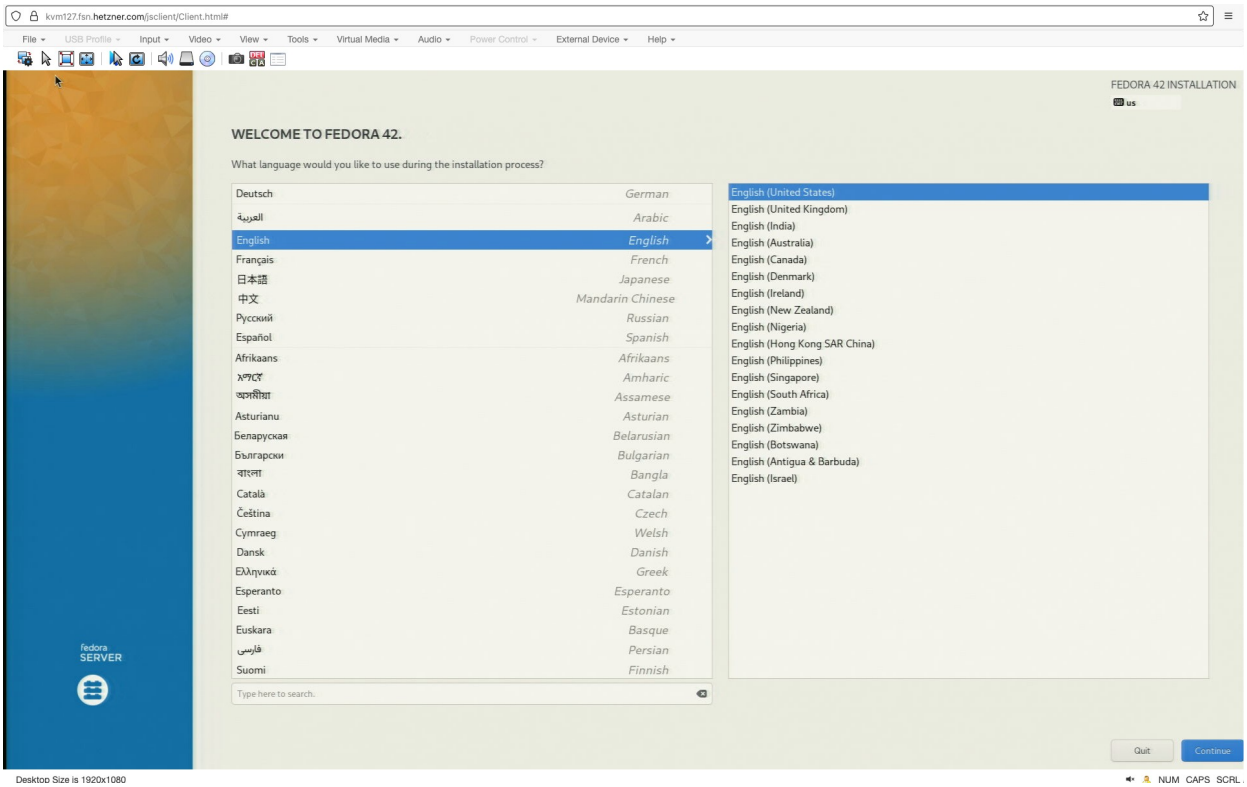
You will get some messages saying „No video output“ and some BIOS messages flickering along the screen. Don't bother, do not do anything. Finally you will get the GRUB screen.



Again, do nothing. After a short time Fedora will start the media test. Be patient, it takes some time and you may see the anoying message „no video from target server“.

If the test fails, inform Hetzner support by replying to the KVM ready mail and ask politely to fix the issue.

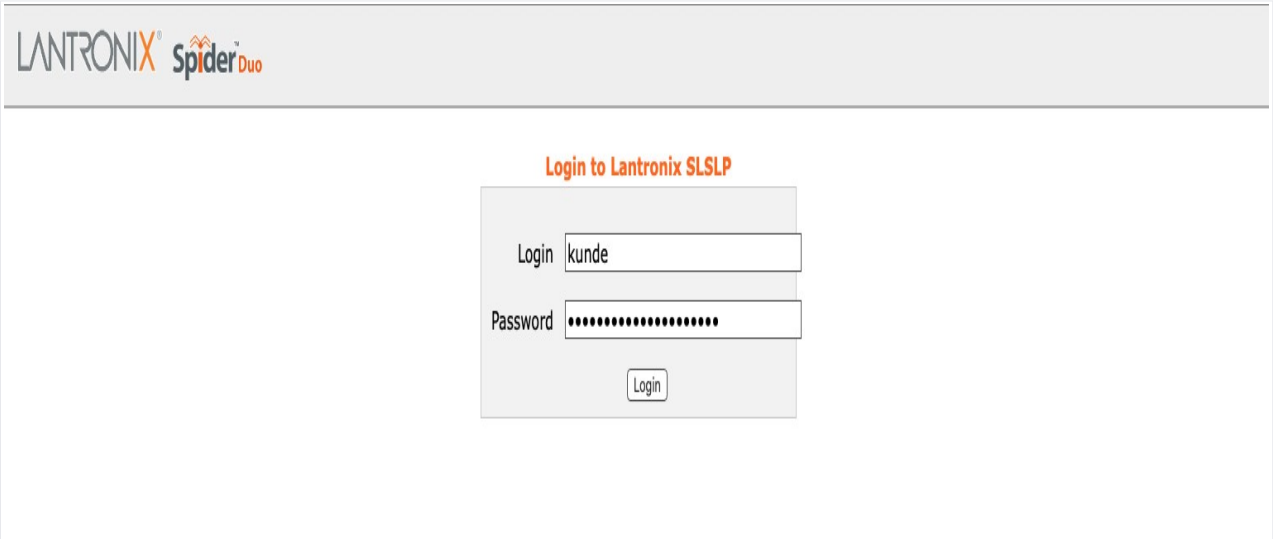
4. Finally you get the Anaconda language selection screen.



Select your language and keyboard setting. From now on you are in the standard Anaconda interactive installation.

(c) Using the Lantronics Spider

1. Enter the provided KVM address from the 'remote console is ready' email into your WEB browser und you get the Login screen.



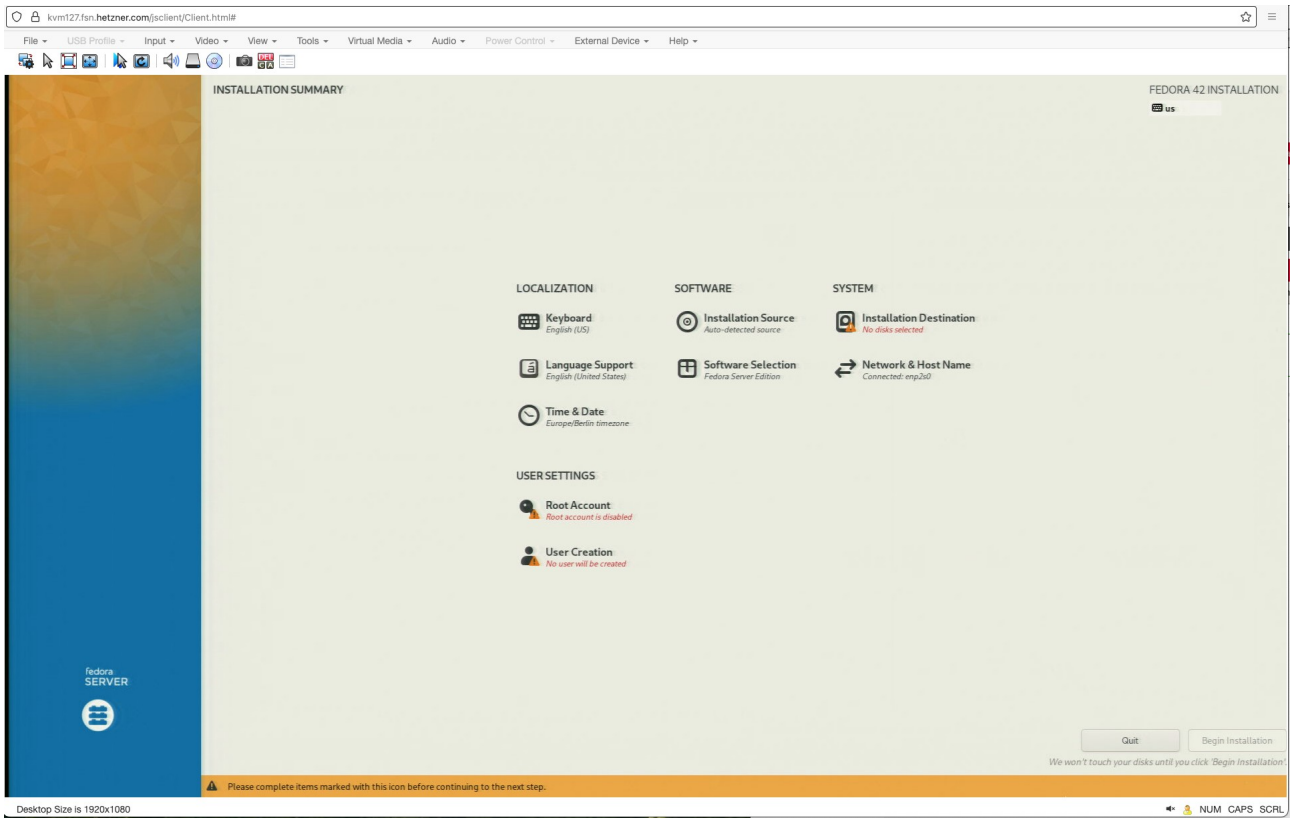
2. Upon successful login, the console automatically connects to the server console.

Clicking into the thumb image activates the console connection. The thumb image is replaced with an active screen that fills the entire view port.

3. Usually Hetzner Support reboots the system after connecting the KVM. With the disks being empty and non-bootable, the server boots the Fedora installation system from the provided USB drive and you will either see various boot messages scrolling by or you are directly greeted by the well known Anaconda language selection screen. Otherwise reboot the server using the reset function of the Web interface.
4. From now on you work in the familiar Anaconda and perform a interactive installation. Continue further down with the „Interactive Installation“ chapter.

5.2 Working in Anaconda interactive system installation

After selecting the language, you will be taken to the familiar initial Anaconda installation summary. From here on, follow the well known, interactive installation procedure as described in the Fedora Server Edition documentation. Unfortunately, the function keys described there do not work in a KVM console. Otherwise, everything works as usual.



Most of it is already correctly preset. It is nevertheless useful to briefly check the items "Keyboard Layout", "Language Support", "Software Selection" (Fedora Server without additional options), and "Time & Date" (Check that an NTP server is configured as well). Don’t modify „Installation Source“.

However, two elements require special attention here: the hard disk and the network configuration.

(a) Disk configuration

As stated at the beginning, due to the remote location, access to the server via SSH is a top priority. This must also be retained in the event of a hard disk failure. It is therefore advisable to mirror the disks and create a RAID 1. Additionally, separate system and user data as strict as possible. For details see the Fedora Server Edition documentation about this topic.

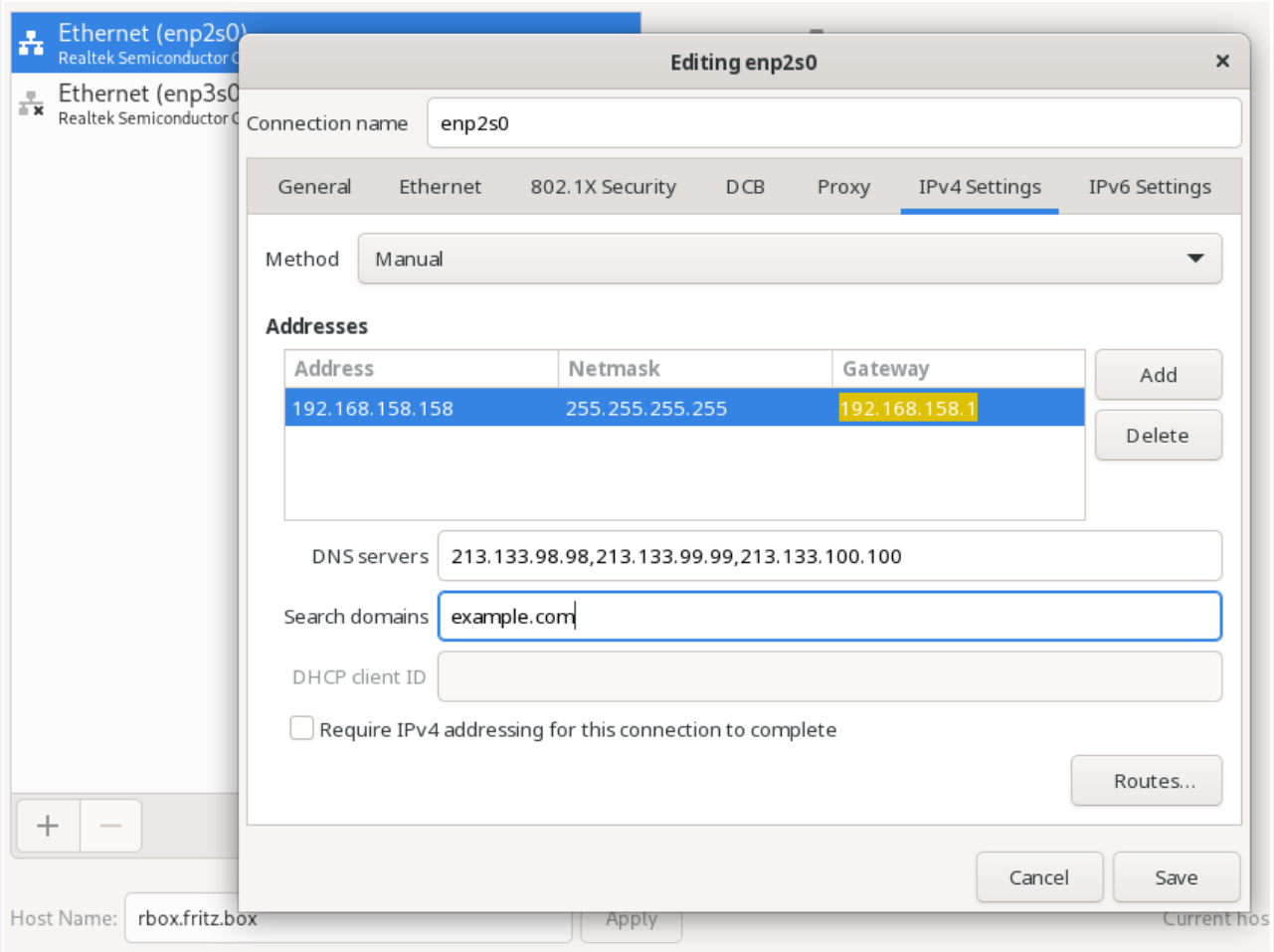
To prevent Anaconda from changing the order of the partitions as it sees fit, use the “Advanced Custom” installation option. For details see the Fedora Server RAID installation documentation.

(b) Network connectivity

The next step is to configure the network connection. Hetzner provides DHCP so you get a working network connection. If your server includes an IPv4 connection, it is already set up via DHCP. An IPv6 connection maybe get automatically configured, too. But sometimes it is not.

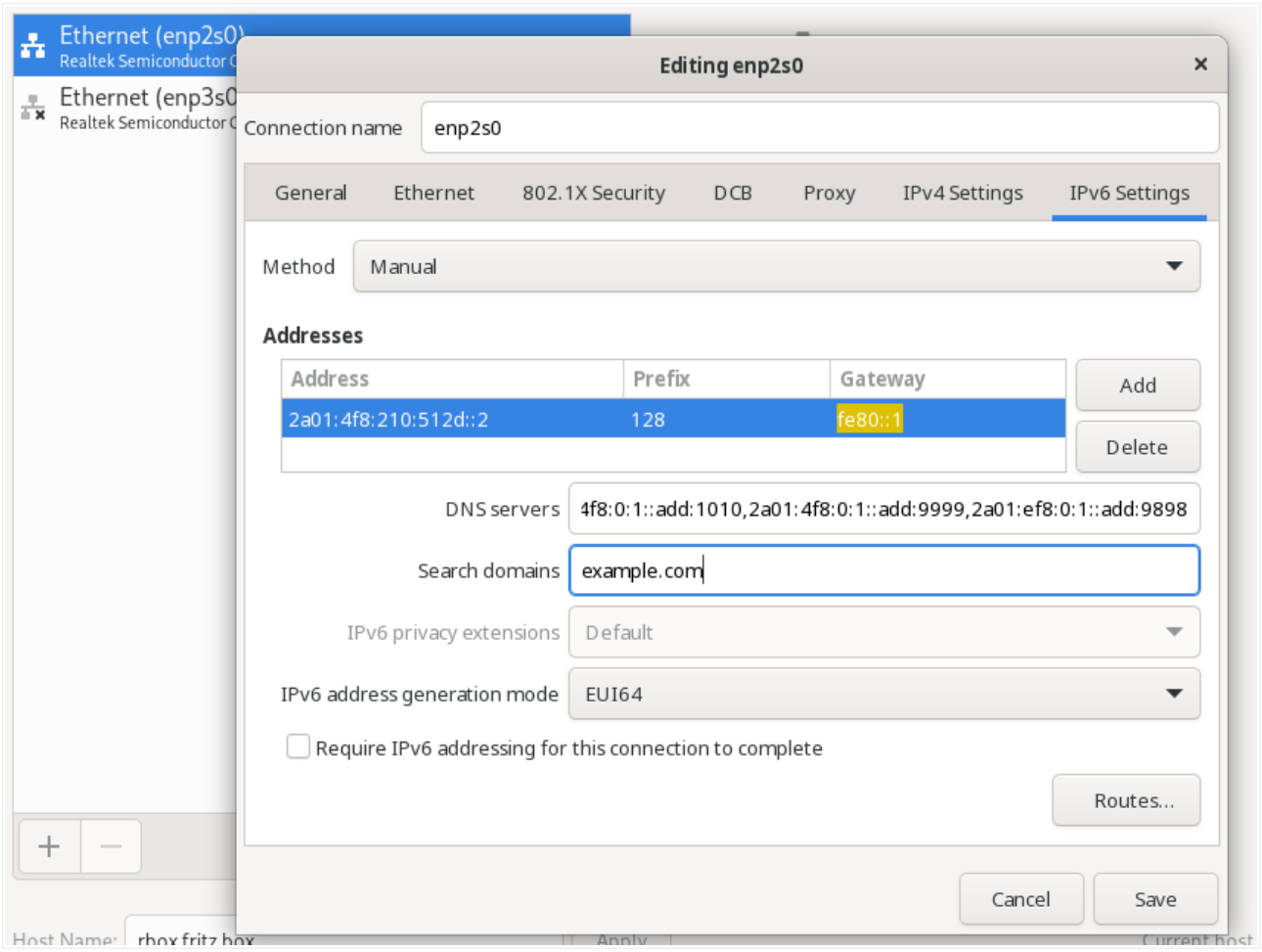
In any case, we have to adjust the nework configuration to take the Hetzner network specifics into account that we explained at the beginning. You have 2 options here: you can either use the anaconda interface to modify the network configuration or you can postpone it to post-installation work.

If you don't like to postpone it, select „Network & Host Name“ from the summary screen and then the active Ethernet connection. In the editing form select the IPv4 Serttings. Select Manual in the Method drop box and then „Add“ to create a new, static IPv4 address.



Fill out the form as shown in the example above. Unlike the usual information, you must enter 255.255.255.255 or /32 as the network mask in order to configure a peer-to-peer connection. The gateway information is displayed when you hover over the IP address in the Hetzner server management Web interface.

Select the IPv6 tab.



Again, fill out the form as shown in the example above. The IPv6 gagteway address is always fe80::1, the host address is ::2 appended to your network address. As prefix length enter 128 if you plan to install virtual maschines, otherwise use the well known 64.

Save the modifications. Anaconda will immediately reconfigure aund restart the network under the hood.

Before you proceed, test the connection using ping and ping6 from your workstation!

(c) Completing the installation

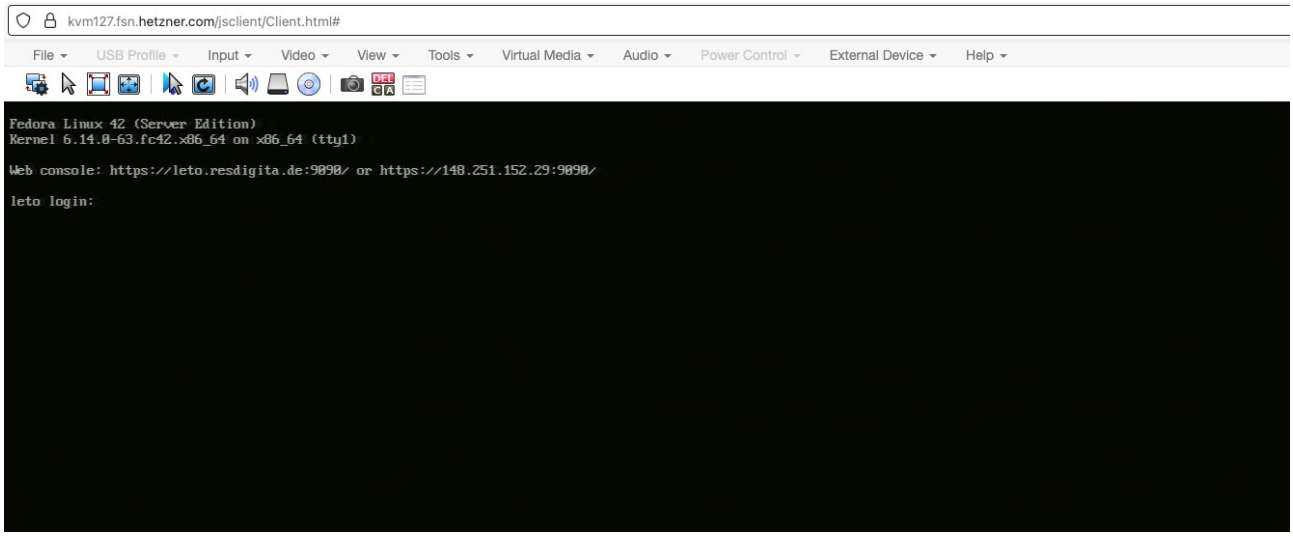
Carry out the final configuration steps, in particular the creation of a regular user with administration authorization. Finally, launch the installation process.

Do not release the KVM console yet, it may still be needed for a subsequent step!

5.3 Checkout the installation

After the installation is complete, the system restarts. The KVM screen goes blank.

After a while you can follow the boot process at the KVM console and finally get the regular login prompt.



In the KVM console, login with your credentials to ensure it works as expected.

Afterwards, login via ssh (non root) as well as via the Web Admintool Cockpit to check, if everything is available. If you didn't adjust the network configuration in Anaconda, it's time to get it done now.

1. Login to the server using the administrative user account and acquire root permissions,

   ```
   […]$ ssh hostmin@example.com
   […]$ sudo -i
   ```

2. Control of the IP addresses

   ```
   […]# ip a
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc n
   …
   2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER
   …
   ```

3. Check NetworkManager configuration

   ```
   […]# nmcli con
   NAME   UUID                                  TYPE      DEVICE
   enp2s0 ccdabaa33b-25b0-3bfd-8a74-b6b40847a7a4 ethernet  enp2s0
   ```

   Sometimes, you get something like

   ```
   […]# nmcli con
   NAME                  UUID                                  TYPE      DEVICE
   'Wired connection 2'  8d971f49-033f-398a-9714-3a4e848178fb  ethernet  enp2s0
   ```

   If so, fix it for convenience. Adjust the naming appropriately.

   ```
   […]# nmcli con mod 'Wired connection 2' connection.id enp2s0
   ```

4. Configure a static IPv6 address. You find the information in your Server Web interface in the IP section. The address of the server is always ::2 and the default gateway is fe80::1.

   ```
   […]# nmcli con mod 'enp2s0' ipv6.method manual \
     ipv6.addresses <YOUR_IPv6_PRAEFIX>::2/64 \
     ipv6.gateway fe80::1 \
     ipv6.dns "2a01:4f8:0:1::add:1010 2a01:4f8:0:1::add:9999"
   ```

```
[…]# nmcli con up <NAME>
[…]# nmcli con reload
```

5. Check the Configuration

```
[…]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc n
…
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER …
   inet <YOUR_IPv4_address>/27 brd …
   ipv6.addresses <YOUR_IPv6_PRAEFIX>::2/64 scope …
   …
```

   Open a ssh connection from another box to ensure it works

```
[…]# ssh hostmin@<YOUR_IPv6_PRAEFIX>::2
```

6. Configure a static IPv4 peer-to-peer connection

```
[…]# nmcli con mod 'enp2s0' ipv4.method manual \
  ipv4.addresses <YOUR_IPv4_ADDRESS>/32 \
  ipv4.gateway <YOUR_IPv4_GATEWAY> \
  ipv4.dns "213.133.98.98 213.133.99.99 213.133.100.100"
[…]# nmcli con up <NAME>
[…]# nmcli con reload
```

7. Again, check the configuration

```
[…]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc n
…
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER …
   inet <YOUR_IPv4_address>/32 brd …
   ipv6.addresses <YOUR_IPv6_PRAEFIX>::2/64 scope …
   …
```

   Open a ssh connection from another box to ensure it works

```
[…]# ssh hostmin@<YOUR_IPv4_ADDRESS>
```

8. Finally, you may check the generated configuraion file. It should show the entries you made in the previous steps.

```
[…]# cat /etc/NetworkManager/systems/enp2s0.nmconnection
[connection]
id=enp2s0
uuid=e039253a-…
type=ethernet
autoconnect-priority=-999
interface-name=enp2s0
timestamp=1752…

[ethernet]

[ipv4]
address1=148.251.152.29/32
dns=213.133.98.98;213.133.99.99;213.133.100.100;
gateway=148.251.152.1
method=manual

[ipv6]
addr-gen-mode=eui64
address1=2a01:4f8:210:512d::2/64
dns=2a01:4f8:0:1::add:1010;2a01:4f8:0:1::add:9999;
gateway=fe80::1
method=manual
```

Of cause, adjust the address values to your server.

If everthing works as expected, release the KVM console.

5.4 Post-installation tasks

Finally, walk through the post-installation tasks as specified in the Fedora Server Edition documentation.

Your Server is now ready for production.

6 Infrastructure for Virtual Machines

As already outlined at the beginning, it makes sense to place all publicly accessible services in a virtual machine for a server located in a non-accessible, remote data center.

The first part is identical to the usual installation and is not specific to Hetzner. The next parts, connecting the virtual machines to the public network and installing virtual machines, require a special configuration.

6.1 Install virtualization support

The installation does not require any special consideration. The easiest way is to follow the Fedora Server Edition documentation about installing virtualization support.

The article recommends installing the cockpit module cockpit-machines. Unfortunately, this module installs a lot of graphical desktop software as a dependency, e.g. Wayland, X11 or Poppler, which you would rather not have lying around on a server and which does not fit in at all with the claim of a lightweight, web-based administration tool. Server administrators may want to do without it, especially as the additional benefit is rather small and cannot by any means substitute Command Line tools.

6.2 Setting Up a Bridge for Public Network Access

As outlined at the begining, it is a peculiarity of the Hetzner infrastructure that all IP addresses assigned to a server are routed to its physical Ethernet interface, i.e. to the same MAC address. So we have to route the data packages using a brouter.

1. Check the forwarding configuration

   To be able to route data packages, forwarding has to be active.

   ```
   […]# cat /proc/sys/net/ipv4/ip_forward
   […]# cat /proc/sys/net/ipv6/conf/default/forwarding
   ```

   Both cases must return 1. The IPv4 part might already have forwarding active because of Libvirt basic configuration, but probably not IPv6. If necessary, activate forwarding temporarily.

   ```
   […]# echo 1 > /proc/sys/net/ipv4/ip_forward
   […]# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
   ```

   For permanent setup, create the following file:

```
[…]# vim /etc/sysctl.d/70-enable-forwarding.conf
# local customizations
#
# enable forwarding for dual stack
net.ipv4.ip_forwarding=1
net.ipv6.conf.all.forwarding=1
```

2. If not already done during server installation, modify the IPv6 subnet to trigger forwarding.

```
[…]# nmcli con mod enp2s0 ipv6.addresses '2a01:4f9:4b:53eb::2/128'
[…]# nmcli con up enp2s0
```

3. Create the bridge

   To clearly describe the function of the bridge, we choose a name analogous to the public Ethernet interface. With an interface enpXsY, the bridge would get the name vbrXsY. The IP addresses are the same as of the external interface, but *different network prefix* to trigger routing! IPv4 is a point-to-point network, so we use the corresponding subnet as provided by Hetzner for the bridge. We configured the IPv6 with a previx of /128, so we use the full /64 for the bridge. Replace the placeholder by the actual IPs.

```
[…]# nmcli con add  con-name vbrXsY  ifname vbrXsY  type bridge  stp off \
     ipv4.method manual  ipv4.addresses 'xx.yy.zz.ww/27'  \
     ipv6.method manual  ipv6.addresses '2a01:xxx:yyy:zzzz::2/64' \
     ipv6.addr-gen-mode eui64 \
     connection.zone FedoraServer


     Connection 'vbrXsY' (...) successfully added
```

   The bridge must reside in the *same zone* as the public interface. This is very important for correct firewall permissions!

4. Add IPv4 routing information

   While iPv6 determines the routing itself, IPv4 explicitly requires a routing entry for each additional IPv4 address.

```
[…]# nmcli con mod vbrXsY +ipv4.routes "xx.yy.zz.uu/32"
[…]# nmcli con mod vbrXsY +ipv4.routes "xx.yy.zz.vv/32"
[…]# nmcli con mod vbrXsY +ipv4.routes "xx.yy.zz.ww/32"
```

5. Finally activate the bridge

```
[…]# nmcli con up vbrXsY
[…]# systemctl restart  NetworkManager
```

   You can only test the bridge after creating at least one VM. But at least you should no longer need to reconfigure the network components, which could potentially cause errors.

6. Check on the local desktop if ping6 and ping -4 work

```
[…]# ping6 <YOUR_IPv6_PRAEFIX>::2
[…]# # e.g. ping6 2a01:4f8:191:6494::2

[…]# ping <YOUR_IPv4>
```

7. Reboot to activate the changed network configuration completely and replicably.

```
[…]# reboot
```

6.3 Setting Up a Virtual Machine

There are two things to keep in mind when setting up a virtual machine:

- The connection device to the public network is the routing bridge.
- The virtual machines' gateway to the public network is the IP address of the host, not the address specified by Hetzner when assigning the IP.

The easiest way to set up a virtual machine with Fedora Server Edition is to use the prebuild Fedora Server Edition virtual disk image. You can then follow the documentation Creating a virtual machine using Fedora Server Edition disk image with two modifications:

1. In the section Import efficiently via CLI, the command for instantiating the VM must be modified to use the routing bridge for public network access:

   ```
   […]# virt-install --name ${YOUR_SERVER_NAME} \
    --memory 4096 --cpu host --vcpus 2 --graphics none \
    --os-variant fedora41 \
    --import \
    --disk /var/lib/libvirt/images/${YOUR_SERVER_NAME}.qcow2,format=qcow2,bus=virtio \
    --network bridge=${vbrXsY},model=virtio \
    --network bridge=virbr0,model=virtio
   ```

   In case you prefer to use Cockpit, you would follow the section Import comfortably via Cockpit. In Step 4, „*Adjust the network connection setup*“, you would modify the Interface type to „Virtual network“ instead of Direct Attachment, and select the Routing Bridge created in the previous paragraph in the „Source“ selection box. Leave the other two input fields unchanged.

   TDB: Einfügen eines Screenshots!!

2. As part of the Minimal initial configuration of the new virtual machine there is the task *Consolidate the network configuration* (step 5). Hetzner needs here a static peer-to-peer connection between the VM and the host. Even if Hetzner DHCP is available, it is unusable because neither the subnet address nor the gateway address is correct. Anyway, a static IP setup (or a corresponding local, host-specific DHCP setup) is a must.

   The instruction in step 5c then looks as follows:

   ```
   […]# nmcli con mod enp1s0 ipv4.method manual \
        ipv4.address "${YOUR_VM_IPv4}/32" \
        ipv4.gateway "${YOUR_HOST_IPv4}" \
        ipv4.dns "213.133.98.98;213.133.99.99;213.133.100.100" \
        ipv4.method manual \
        ipv6.addresses ${YOUR_VM_IPv6}/64 \
        ipv6.gateway ${YOUR_HOST_IPv6} \
        ipv6.dns "2a01:4f8:0:1::add:1010;2a01:4f8:0:1::add:9999"  \
        connection.zone "FedoraServer"
   […]# nmcli con up enp1s0
   […]# systemctl  restart  NetworkManager
   ```

3. You can create as many VMs as the host capacity allows. They can be configured with different combinations of network connections. If, for example, a VM is only to be operated with IPv6, you would only enter the following for IPv4: `ipv4.method disabled`

The host and VMs are now ready for use. Application software can be installed as usual and the server can go into production.