

Kathará

Single-host

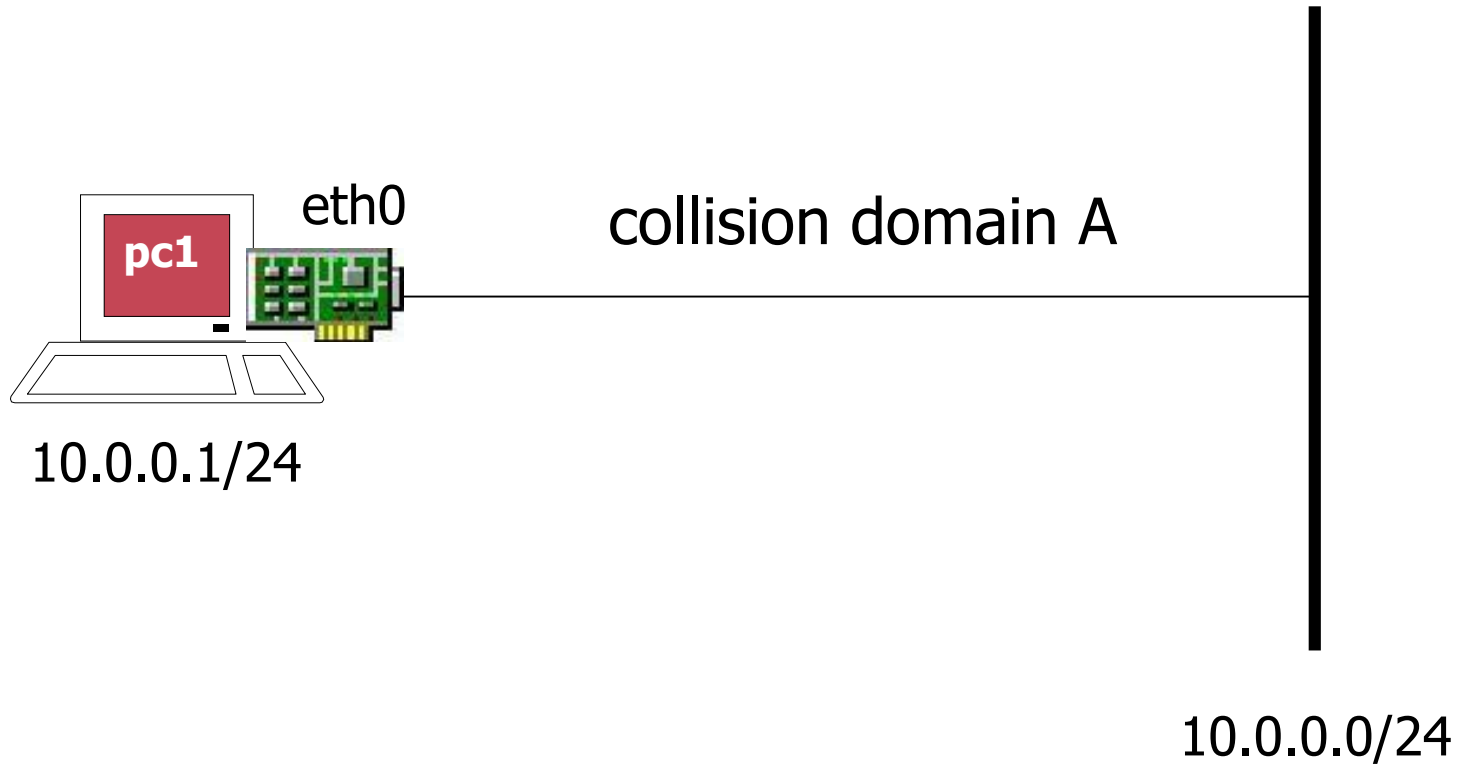
Updated by K. Kosek-Szott
AGH University

Version	1.0
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	how to set up and manage a single virtual machine

copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material “as is”, with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.

single host



step 1 – creating the vms

host machine

```
user@localhost:~$ kathara vstart --eth 0:A -n pc1
```

```
===== Starting Device =====
```

```
Deploying collision domains...
```

```
|#####| 1/1
```

```
Deploying devices...
```

```
|#####| 1/1
```

pc1 is created and a terminal window opens for pc1

step 2 – logging on pc1

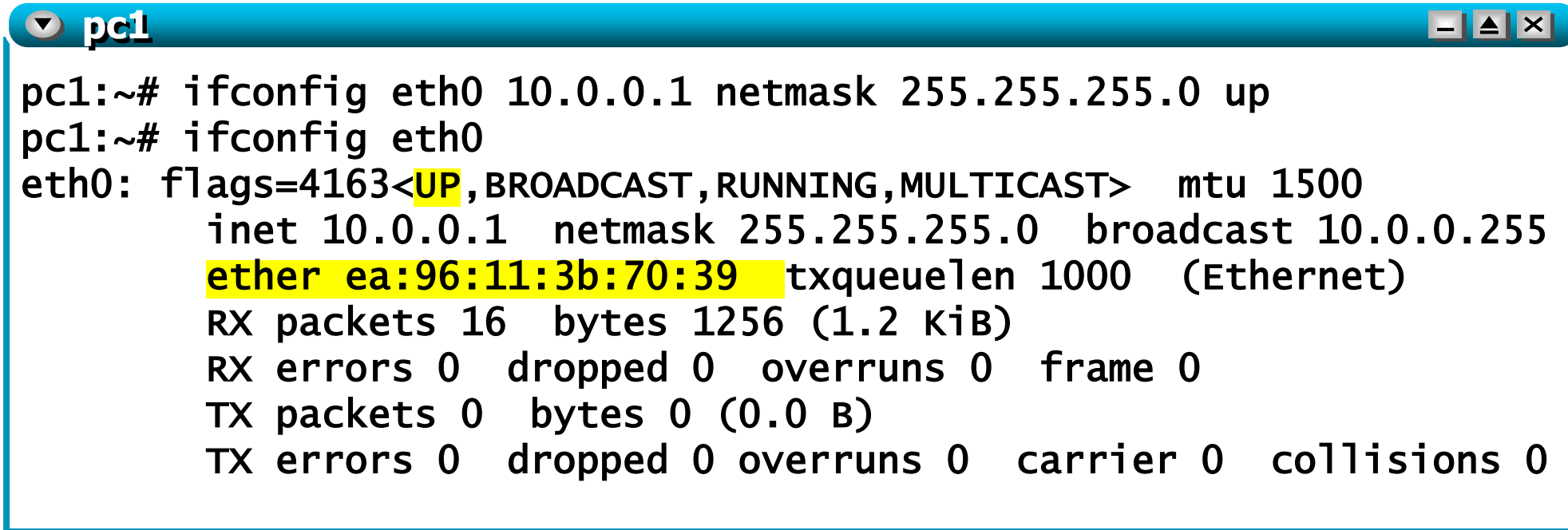
- now you are the administrator (root) of pc1



```
host machine
user@localhost:~$ kathara list
```

- check also the docker Containers/Apps, you should see your pc1 container running

step 4 – configuring the network interface of pc1



```
pc1:~# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 up
pc1:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.0.1  netmask 255.255.255.0  broadcast 10.0.0.255
    ether ea:96:11:3b:70:39  txqueuelen 1000  (Ethernet)
    RX packets 16  bytes 1256 (1.2 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- eth0 has an automatically assigned MAC (ether) address
- eth0 interface is up and running

step 5 – checking the routing table

```
pc1
pc1:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
```

- gateway is set to 0.0.0.0
- the if status is **UP**

step 6 – shutting down pc1 terminal



- Click the x button in the pc1 terminal
- This shuts down the terminal window (halts pc1), but the container is still running (!)
- You can check this in the docker application or by using the following command



step 6 – shutting down the vm

▼ host machine



```
user@localhost:~$ kathara vclean -n pc1
```

- Stops and deletes a single Kathara device
- The pc1 container will be deleted from the docker

step 7 – permanent configuration

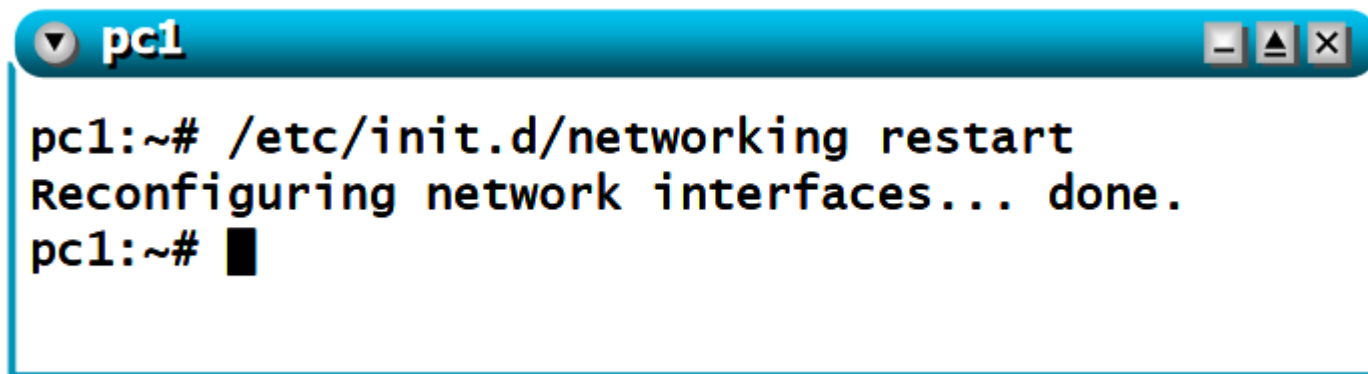
- Permanent configuration can be obtained, e.g., by editing `/etc/network/interfaces` (inside the virtual `pc1`) and appending the following lines:

```
auto eth0
iface eth0 inet static
    address 10.0.0.1
    network 10.0.0.0
    netmask 255.255.255.0
```

- tips:
 - Permanent settings in a virtual machine can be configured by editing same files that would be used in a real linux system.
 - The `interfaces` file typically contains the configuration of various networking functions
 - Use linux text editor, like `vi`, `vim` or `nano`
 - Do not delete pre-existing contents of the `interfaces` file
 - Simple change of the `interfaces` file does not immediately affect the operating system behaviour. You need to restart the networking services.
 - Note well that stopping or removing the `pc1` container from the docker irrevocably removes the virtual machine, along with its permanent configuration.

step 8 – restarting network services

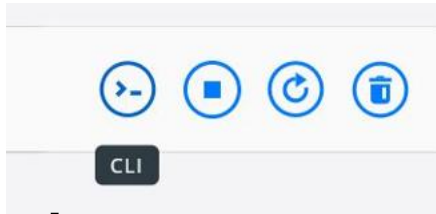
- to see the effect of new configuration on the system behaviour, please restart network services (but not the system):

A terminal window titled 'pc1' with standard window controls (minimize, maximize, close). The terminal shows a command being executed to restart network services, followed by a confirmation message and a new prompt.

```
pc1:~# /etc/init.d/networking restart
Reconfiguring network interfaces... done.
pc1:~# █
```

step 9 – opening pc1 container

- in Docker Desktop using `/bin/sh`, by selecting `>_`



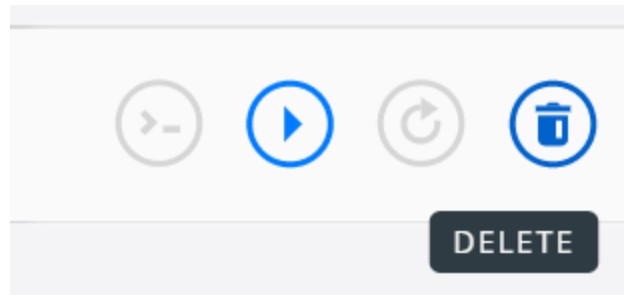
- in the host machine using `/bin/bash` in a running container

```
host machine
user@localhost:~$ docker exec -it <pc1 container name> /bin/bash
```


- **caution:** this is effective until you reboot the host operating system (your Windows 10)

step 10 – removing pc1 container

- from docker using `/bin/sh` by selecting delete button



- from host machine

A screenshot of a terminal window with a red title bar. The title bar contains a dropdown arrow, the text "host machine", and standard window control buttons (minimize, maximize, close). The terminal content shows a command prompt "user@localhost:~\$" followed by the command "kathara.exe vclean -n pc1".

```
▼ host machine
user@localhost:~$ kathara.exe vclean -n pc1
```