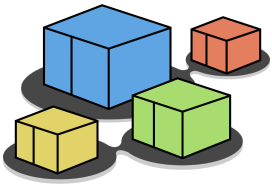


Kathará

Kathará

A container-based framework for experimenting computer networking

| | |
|--------------------|---|
| Version | 2.1 |
| Author(s) | L. Ariemma, G. Bonofiglio, G. Di Battista, V. Iovinella, G. Lospoto, M. Patrignani, M. Pizzonia, M. Rimondini |
| E-mail | contact@kathara.org |
| Web | https://www.kathara.org/ |
| Description | An introduction to the architecture, setup and usage of Kathará – based on a similar presentation of Netkit |



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.

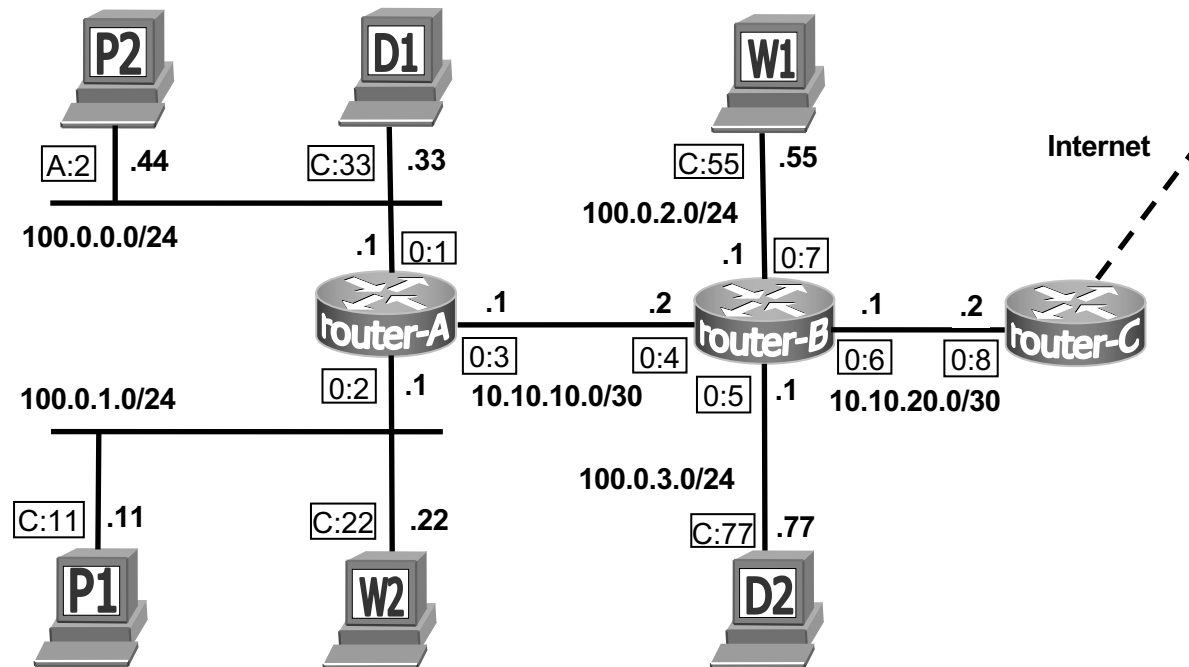


About computer networks

- Computer networks are quite complex
 - Several devices (computers, routers, etc.)
 - Several interfaces
 - Several protocols running
 - Physical interconnections originate complex topologies



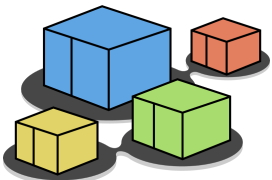
Example of a computer network





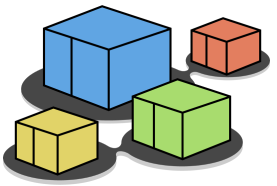
How to perform experiments?

- Performing experiments may be unfeasible
- A live network cannot be exploited for experiments
 - It hosts services that are critical for the company
 - It would be necessary to coordinate different departments of the company
- Network equipment is expensive
 - Sometimes, even for performing simple experiments, several equipment should be available in the same test bed



Simulation vs. emulation

- Emulation and simulation systems put at user's disposal a virtual environment that can be exploited for tests, experiments, measures
- **Simulation systems** aim at reproducing the **performance** of a real-life system (latency time, packet loss, etc.)
 - e.g.: ns, real, ...
- **Emulation systems** aim at accurately reproducing the **functionalities** of a real-life system (configurations, architectures, protocols), with limited attention to performance



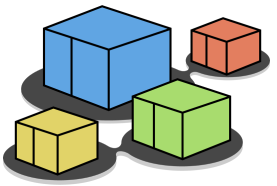
Kathará

a system for emulating
computer networks



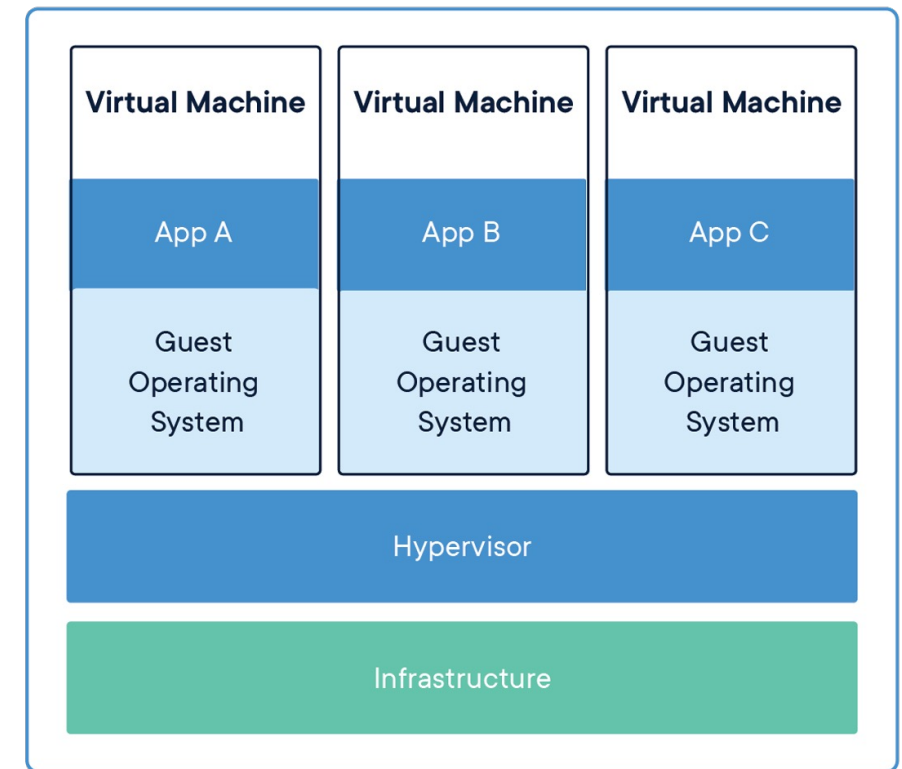
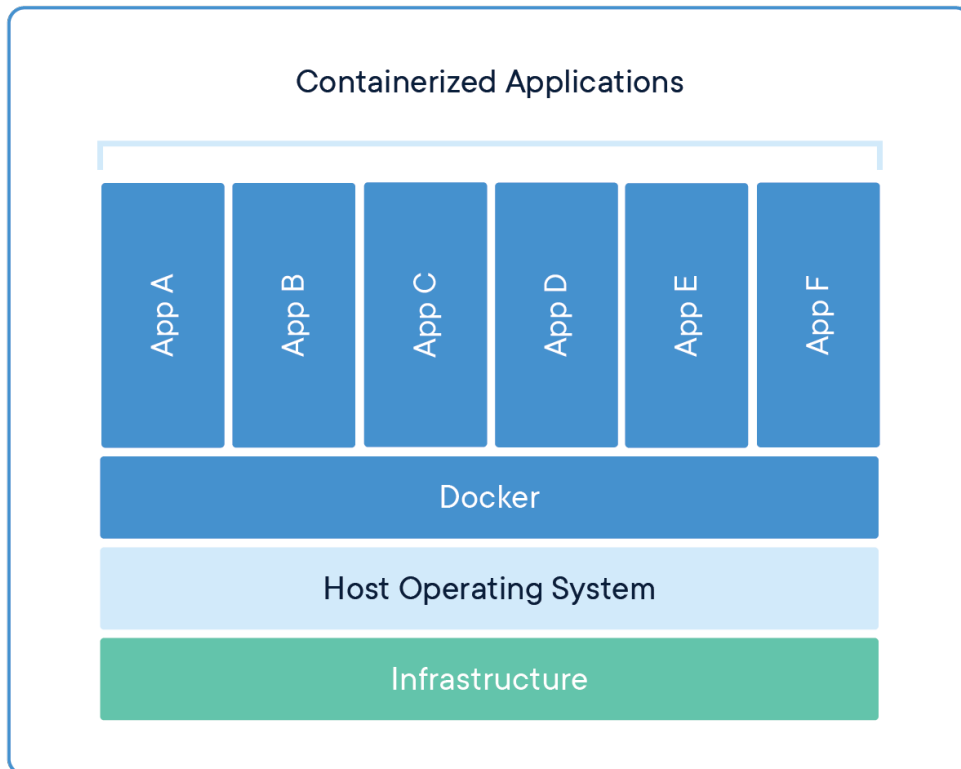
Emulating a network

- Basic idea:
 - several containers are created inside a single host machine
 - containers are connected to virtual collision domains and thus can communicate with each other
- Each container can be configured as a device that plays the role of a regular host, of a router, of a switch,

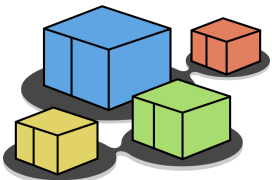


Docker and containers

A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.

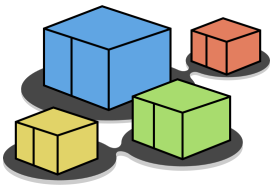


<https://www.docker.com/resources/what-container>



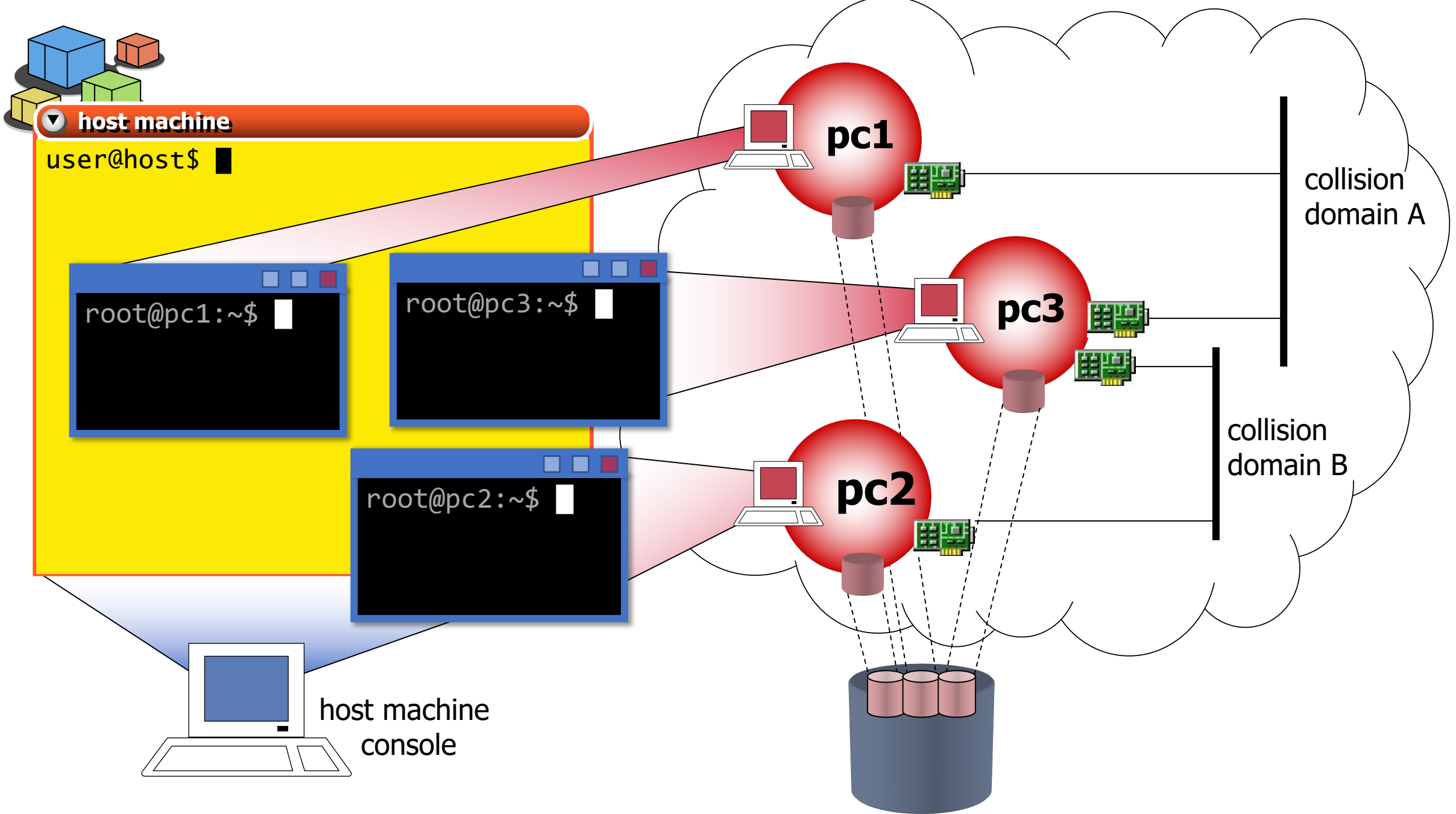
Kathará

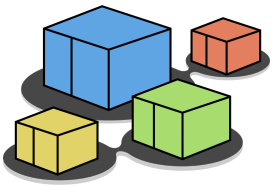
- Based on Docker
- Each emulated network device (in what follows *device*) is a container
- Note: several container images available, e.g.:
 - Base (DNS, Web Server, network utilities)
 - Quagga (standard routing protocols)
 - **FRRouting (standard routing protocols + EVPN + MPLS)**
 - Open vSwitch (Open Flow enabled switch)
 - Behavioral Model (software implementation of a P4 switch)



Emulated network devices

- Each device has:
 - A console (a terminal window)
 - A memory
 - A filesystem
 - (zero, one or more) network interfaces
- Each network interface is connected to a single (virtual) collision domain
- Each virtual collision domain can be connected to several interfaces





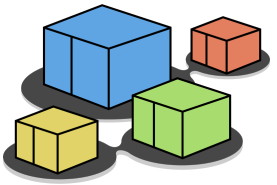
Setting up Kathará

Setup manual



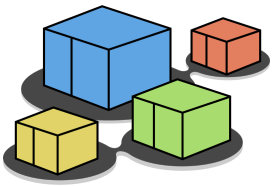
Installing Kathará

- Available for:
 - Windows
 - Linux
 - MacOS
- Download at <https://www.kathara.org/>
- Follow the wiki

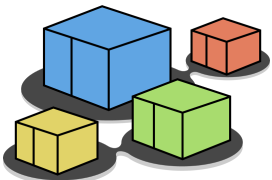


Changing the default Docker Image

- Run `kathara settings`
- Select Choose default image by pressing 2
- Select `kathara/frr` by pressing 2
- Confirm and exit



Using Kathará



Kathará commands

- Kathará provides users with three sets of commands
 - v-prefixed commands (v-commands)
 - l-prefixed commands (l-commands)
 - Global commands
- v-commands act as low-level tools for configuring and starting up a single device
- l-commands provide an easier-to-use environment to set up complex labs consisting of several devices
- Global commands are mainly management commands



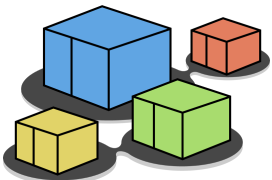
Kathará v-commands

- Allow to startup a single device with arbitrary configurations (network interfaces, etc.)
 - **vstart**: starts a new device
 - **vconfig**: attaches network interfaces to a running device
 - **vclean**: halts a device



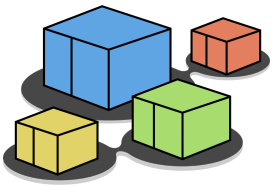
Kathará I-commands

- Ease setting up complex labs consisting of several virtual machines
 - **lstart**: starts a Kathará lab
 - **lclean**: halts all the devices of a lab
 - **lrestart**: halts all the devices of a lab and start them again
 - **linfo**: provides information about a lab

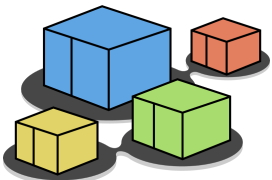


Kathará global commands

- Management commands
 - **check**: Check your system environment
 - **connect**: Connect to a running Kathará machine
 - **list**: Show all running Kathará machines of the current user
 - **settings**: Show and edit Kathará settings
 - **wipe**: Delete all Kathará machines and links, optionally also delete settings

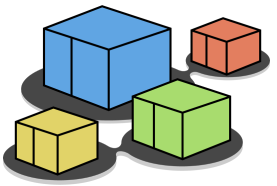


Testing Kathará

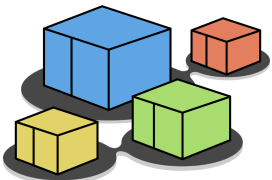


Testing Kathará

- To test if your setup works correctly run:
 - `kathara check`
 - This command will run automatic tests to your environment
 - `kathara vstart -n pc1 --eth 0:A`
 - This command will start a new device called **pc1** and connected to the virtual collision domain **A**
 - A terminal window will open allowing to run commands inside the device
 - `kathara vclean -n pc1`
 - This command will stop the previous started device

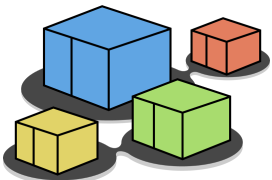


Preparing a Kathará lab



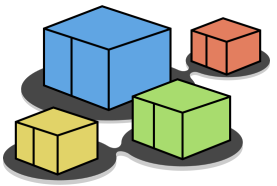
Kathará lab

- a **Kathará lab** is a set of preconfigured devices that can be started and halted together
- a basic Kathará lab is a directory tree containing:
 - a **lab.conf** file describing the network topology
 - a set of **subdirectories** that contain the configuration settings for each device
 - **<device_name>.startup** files that describe actions performed by devices when they are started



lab.conf

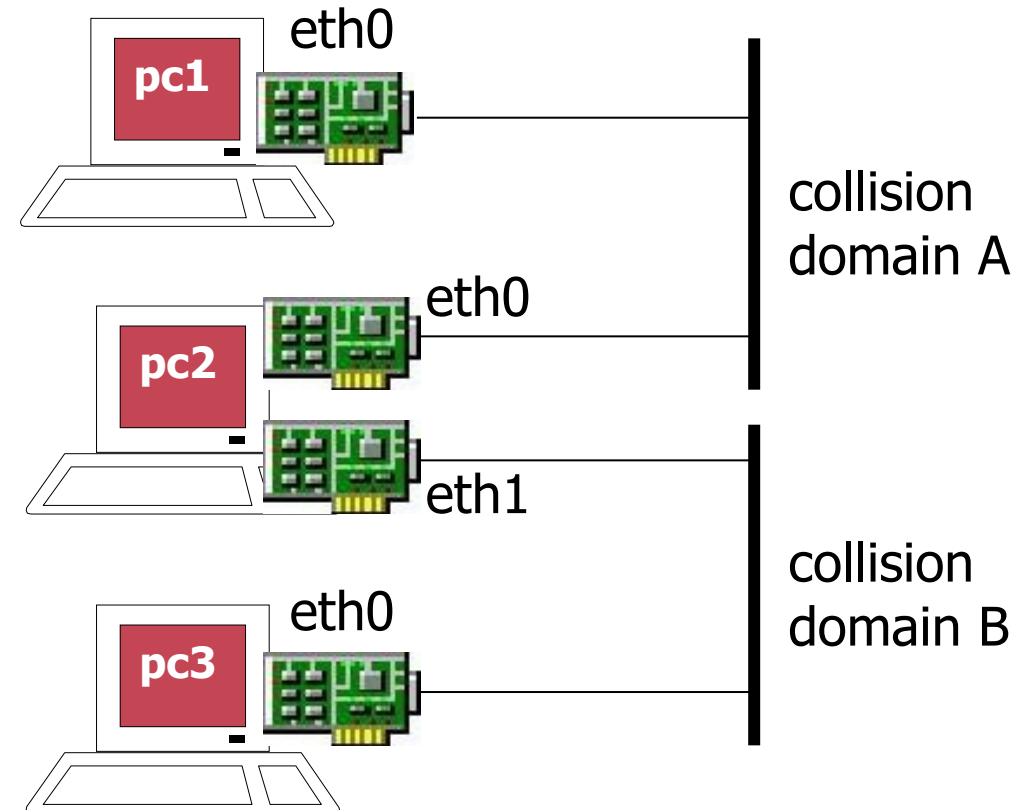
- describes:
 - The topology of the network that interconnects the lab's devices
 - The devices to be started
- contain a list of **machine[arg]=value** lines where:
 - **machine** is the name of the device (e.g. pc1)
 - if **arg** is a number, then **value** is the name of a collision domain to which **etharg** should be attached
 - if **arg** is not a number, then it must be an option and **value** the argument

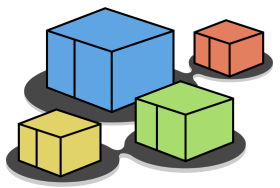


lab.conf

■ example

```
pc1[0]=A  
  
pc2[0]=A  
pc2[1]=B  
  
pc3[0]=B
```





Share files between the host and the devices

- There are two ways to share files between the host filesystem and the device filesystem:
 - Share files mirrored to the device
 - A change inside the device will reflect in the host filesystem
 - A change in the host filesystem will reflect in the device
 - Share files copied to the device
 - Two independent copies of the same files



Share files mirrored to the device

- There are two ways to share mirrored files:
 - the `/shared` directory inside a device directly points to the **shared** directory inside the lab
 - by default it is **ENABLED**, you can disable it in the settings
 - the `/hosthome` directory inside a device directly points to the **home directory** of the current user of the host
 - by default it is **DISABLED**, you can enable it in the settings



Share files copied to the device

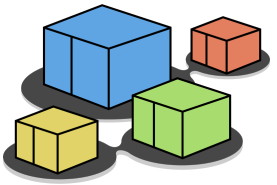
- Can be done through the subdirectories of a lab
- The contents of subdirectory **device** are copied into the root (/) of the device named **device** filesystem
 - for example, `pc1/foo/file.txt` is copied to `/foo/file.txt` inside the device `pc1`



.startup files

- Shell scripts that are executed inside a device right after its startup
- Typical usage of a `.startup` file is to configure network interfaces and/or start network services
 - For example:

```
ip address add 10.0.0.1/24 dev eth0  
systemctl start frr
```



Launching/halting a lab

- Open a terminal
- Enter the lab directory (**`cd lab_directory`**)
- Launch a Kathará l-command
 - Where l-command could be one of the following
 - **`kathara lstart`**, to start the lab
 - **`kathara lclean`**, to stop the lab
 - **`kathara lrestart`**, to restart the lab



More information

- Further information can be found:
 - On GitHub's Kathará wiki
 - On the official website <https://www.kathara.org>
 - Inside Kathará man pages
 - On Linux and MacOS accessible by the terminal `man kathara`
 - Available online at <https://www.kathara.org/man-pages/kathara.1.html>