netkit lab

single-host

<table>
<thead>
<tr>
<th>Version</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini</td>
</tr>
<tr>
<td>E-mail</td>
<td><a href="mailto:contact@netkit.org">contact@netkit.org</a></td>
</tr>
<tr>
<td>Web</td>
<td><a href="http://www.netkit.org/">http://www.netkit.org/</a></td>
</tr>
<tr>
<td>Description</td>
<td>how to set up and manage a single virtual machine</td>
</tr>
</tbody>
</table>
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

- netkit *little by little*: just a single virtual machine
- suggestion: before setting up a netkit lab, always make a clear diagram of the scenario that you are going to emulate!
- a simple scenario:

![Diagram showing a single host with IP addresses 10.0.0.1/24 and collision domain 0]
step 1 – creating a virtual machine

user@localhost:~$ vlist

<table>
<thead>
<tr>
<th>USER</th>
<th>VHOST</th>
<th>PID</th>
<th>SIZE</th>
<th>INTERFACES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>(you),</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>(all users).</td>
<td></td>
</tr>
</tbody>
</table>

Total virtual machines:
Total consumed memory:

user@localhost:~$ vstart pc1 --eth0=A

start a vm ...

...whose name is `pc1`

...that has a network interface on the collision domain called “A”
step 2 – logging on pc1

- A window containing pc1’s console pops up.
- Once the bootstrap of pc1 is terminated, a shell prompt is automatically displayed.

--- Netkit phase 2 init script terminated

pc1 login: root (automatic login)
Linux pc1 2.6.11.7 #1 Tue Sep 13 18:38:01 CEST 2005 i686 GNU/Linux
Welcome to Netkit

pc1:~#
step 3 – back to the host machine console

```
host machine

user@localhost:~$ vlist
USER    VHOST
user    pc1

Total virtual machines: 1 (you), 1 (all users).
Total consumed memory: 12380 KB (you), 12380 KB (all users).

```

```
user@localhost:~$ ls -l *.disk
-rw-r--r--  1 user  group  630358016 2006-02-02 16:07 pc1.disk
user@localhost:~$ 
```

```
netkit – [ lab: single-host ]
```

last update: Apr 2007
step 4 – configuring the network interface of pc1

```
interface of pc1

pc1:~# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up

automatically assigned mac address

interface status

pc1:~# ifconfig eth0

eth0      Link encap:Ethernet  HWaddr FE:FD:0A:00:00:01
         inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
         inet6 addr: fe80::fcfd:aff:fe00:0/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         RX packets:3 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 b)  TX bytes:238 (238.0 b)
         Interrupt:5
```

© Computer Networks
Research Group Roma Tre

netkit – [ lab: single-host ]
step 5 – checking the routing table

- the routing table has been automatically updated when the interface has been brought up:

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

- other labs show how to manually alter the routing table

netkit – [ lab: single-host ]
step 6 – shutting down the vm

- three possibilities
  - from inside
    - pc1:~# halt
  - from outside
    - user@localhost:~$ vhalt pc1
      Halting virtual machine "pc1" (PID 3559) owned by user [........]
step 6 – shutting down the vm

- from outside, brute force

```
user@localhost:~$ vcrash pc1

================ Crashing virtual machine "pc1" (PID 4830) =========
Virtual machine owner: user
Virtual machine mconsole socket: /home/user/.netkit/mconsole/pc1/mconsole
Crashing... done.
user@localhost:~$ 
```

- unless you chose to use vcrash, pc1’s filesystem is still stored in file pc1.disk, so it will be used again when pc1 is restarted
step 7 – a permanent configuration

- After halting **pc1**, if you want to restart it you also have to reconfigure its network interface **eth0**
- A permanent configuration can be obtained, e.g., by editing `/etc/network/interfaces` (inside the VM **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:**
  - You can use an editor like **vi** or **mcedit**
  - The permanent settings of a VM can be configured inside the same files that would be used in a real Linux box
- Removing the VM filesystem (**pc1.disk**) removes any permanent configuration as well.
step 8 – restarting network services

- at next boot **pc1** will be automatically configured by the os which will perform the suitable **ifconfig** and **route** commands based on the contents of **/etc/network/interfaces**

- the new configuration can also be fetched without rebooting by restarting network services:

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