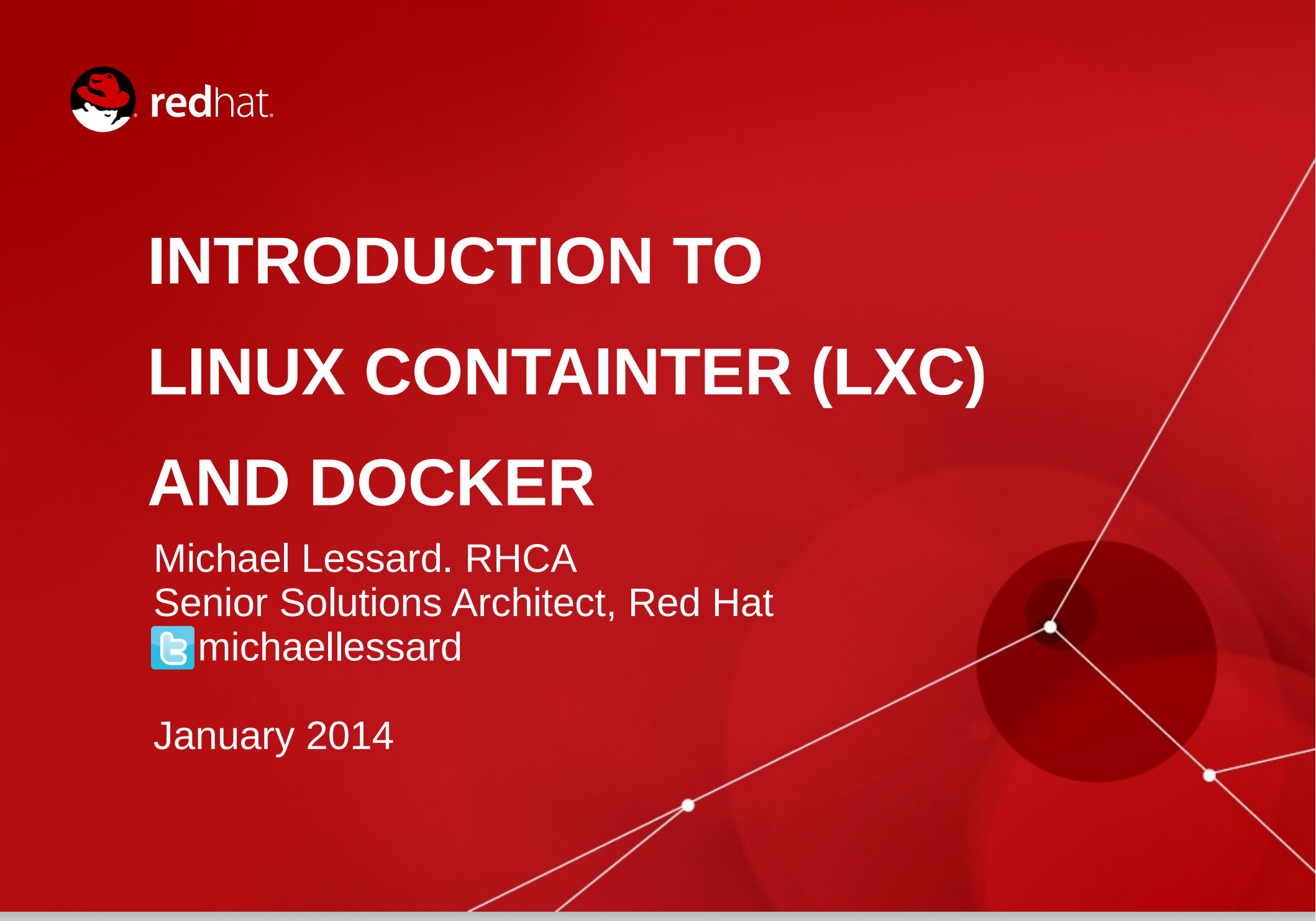


INTRODUCTION TO LINUX CONTAINTER (LXC) AND DOCKER

Michael Lessard. RHCA
Senior Solutions Architect, Red Hat
michaellessard

January 2014

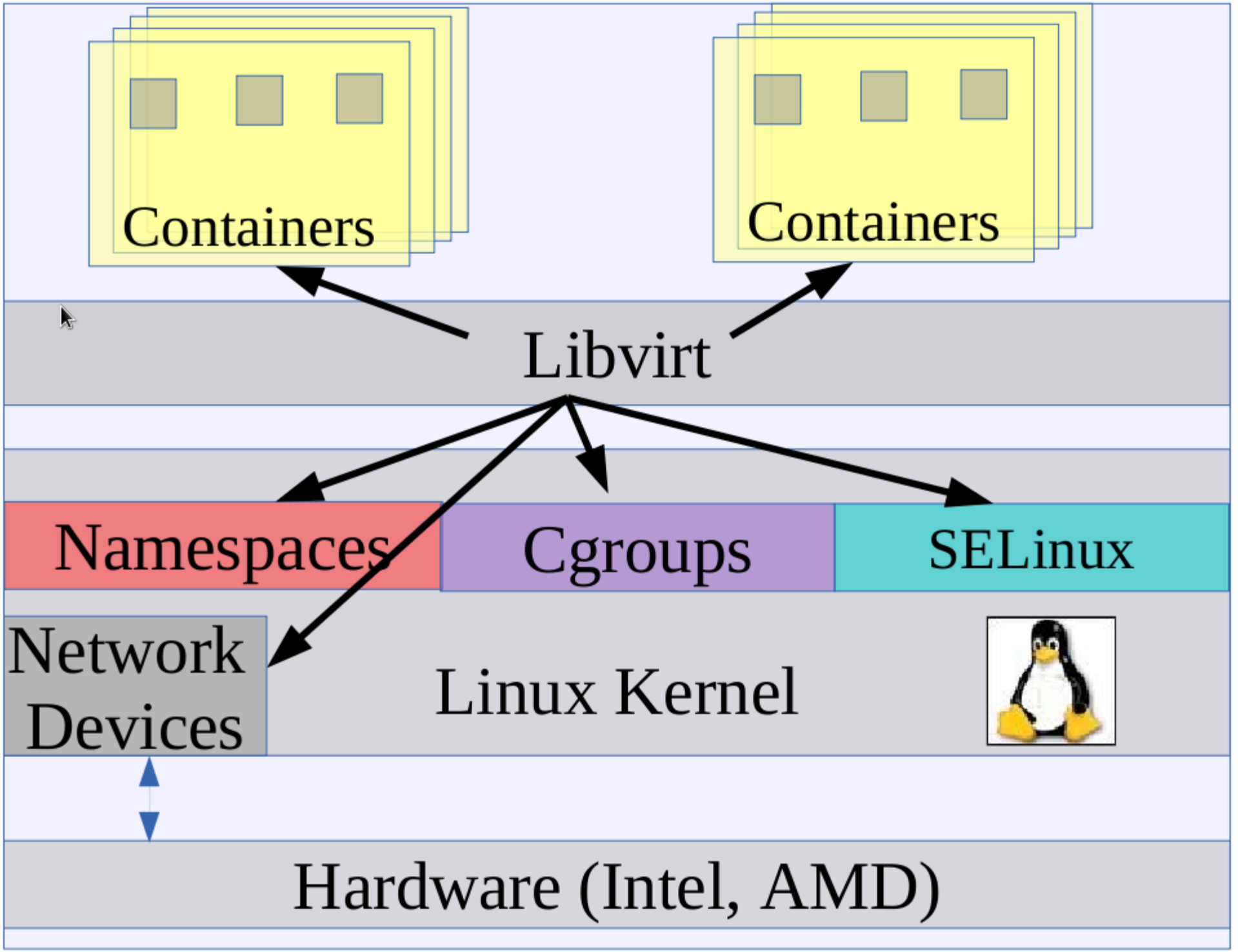
A bit of history – Virtualization and containers

- Chroot (version 7 Unix, 1979)
- FreeBSD Jails (FreeBSD 4, 2000)
- Linux vserver (Linux, Oct 2001)
- Para-virtualization Xen (Linux, 2003)
- Solaris zones (Solaris 10, 2004)
- OpenVZ (Linux, 2005)
- Full virtualization KVM (Linux, 2007)
- Linux Containers - LXC (Linux 2.6.29 2009)

In red – Virtualization on the os level (containers)

What is LXC ?

- An operating system-level virtualization
- Light weight virtualization
- Containers
- Relies on cgroup,selinux and namespace
- Included in the kernel
- Can be managed using libvirt-lxc (RHEL and Fedora) or lxc-tools (Fedora)
- Perceived near bear metal performance


Containers
Containers
Libvirt
Namespaces
Cgroups
SELinux
Network Devices
Linux Kernel
Hardware (Intel, AMD)

Uses cases

- Lightweight web servers
- Testing environment
- Application isolation
- Low latency app

Weaknesses

- Locked into running the host kernel
  - Unlike a fully virtualized machine, you are restricted to the kernel running on the host
- No Windows support


Demo

DEMO WITH LIBVIRT LXC (FEDORA)

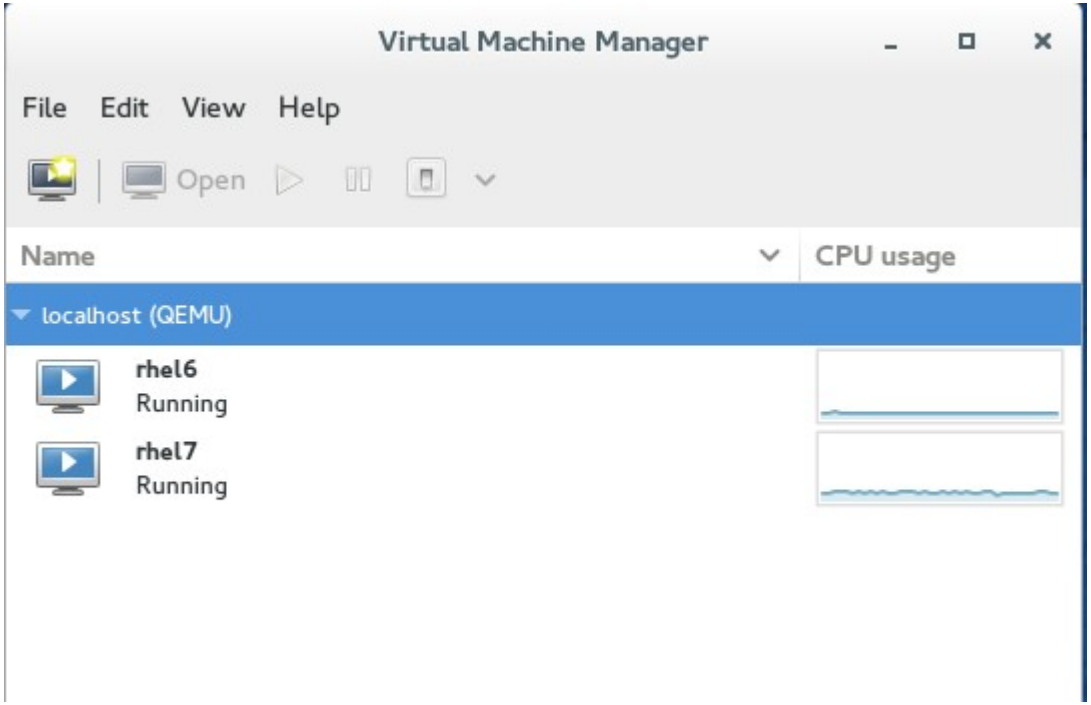
```
# yum install lxc libvirt-daemon-driver-lxc
# sytemctl restart libvirtd
```

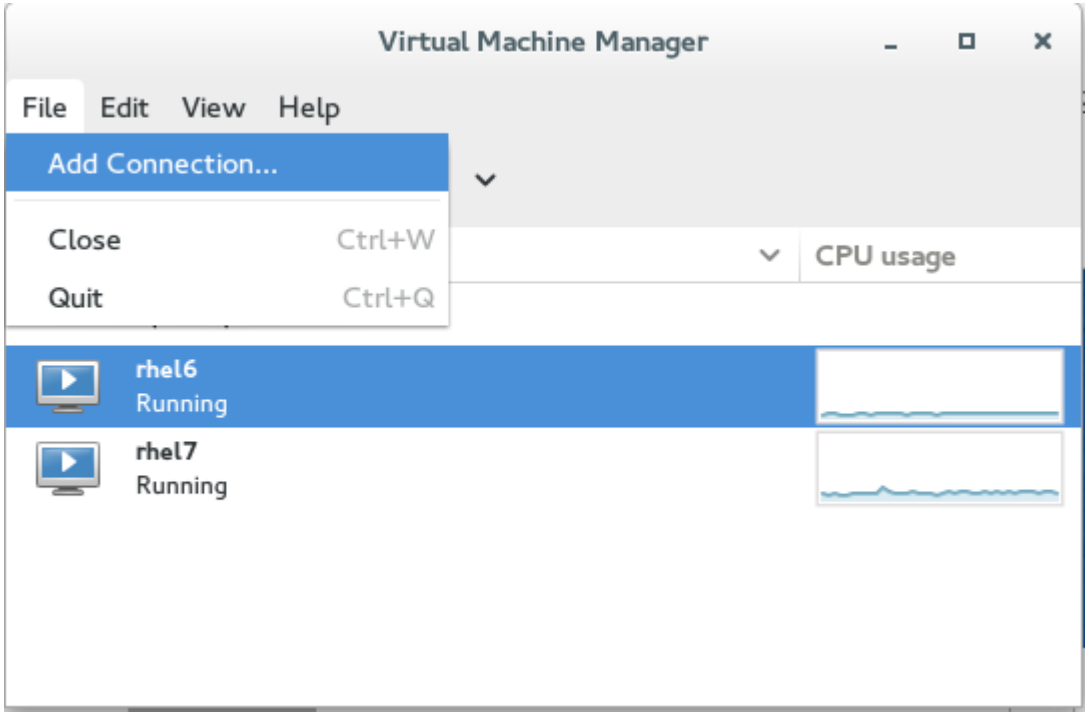
Demo lxc container1 and container2

Demo virt-sandbox

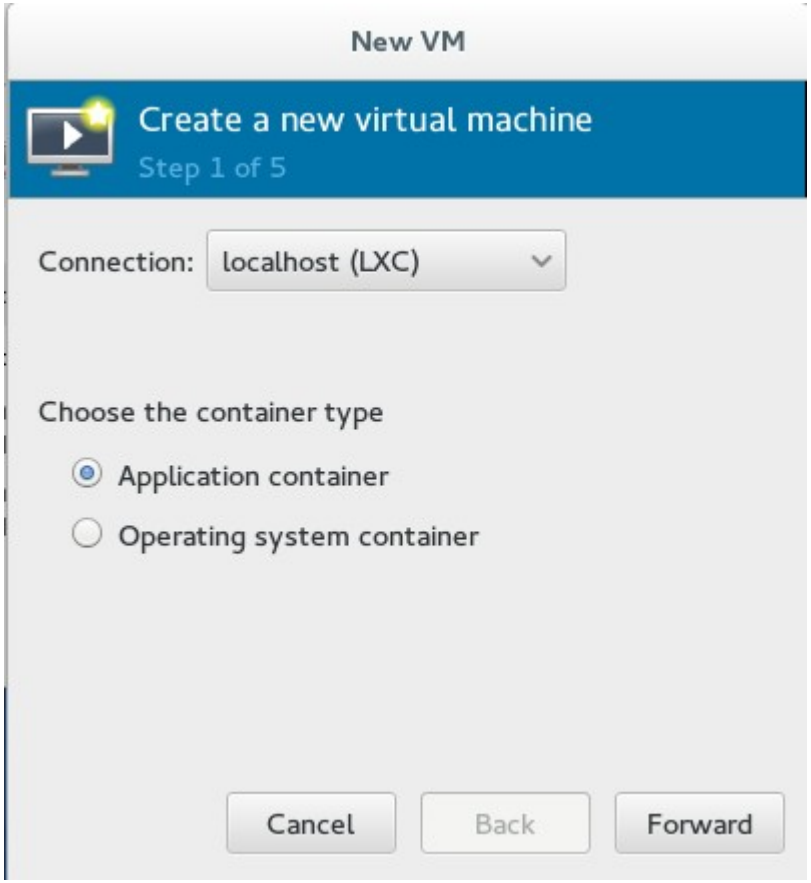
Demo Docker

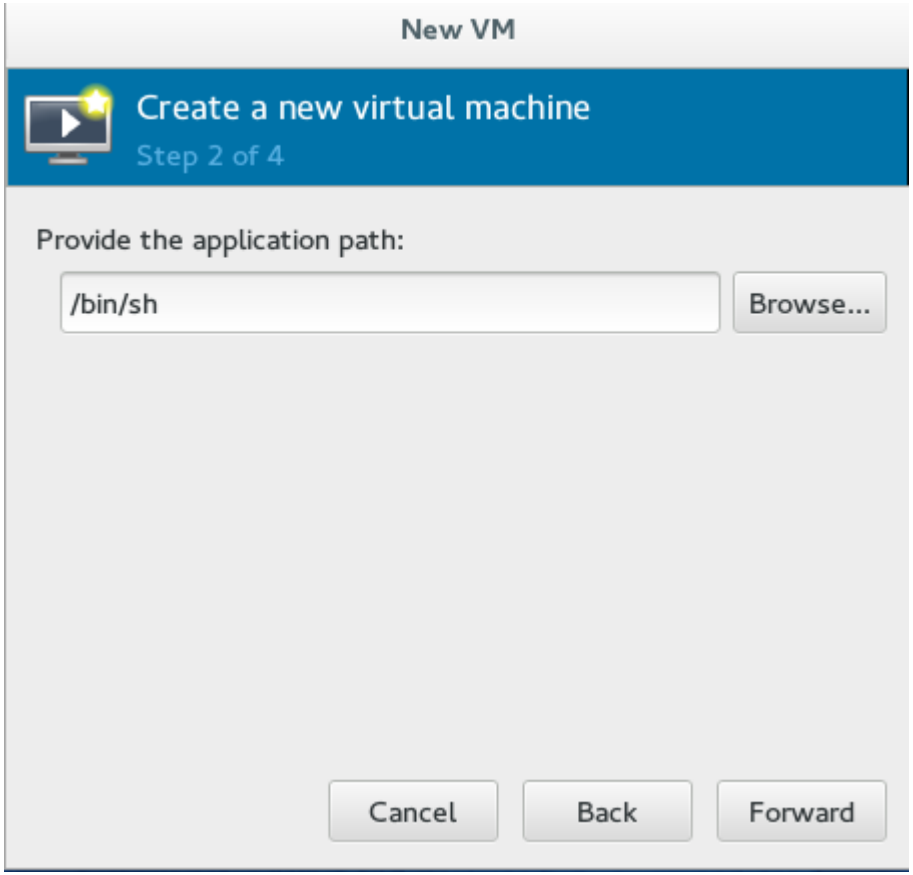
DEMO USING VIRT-MANAGER

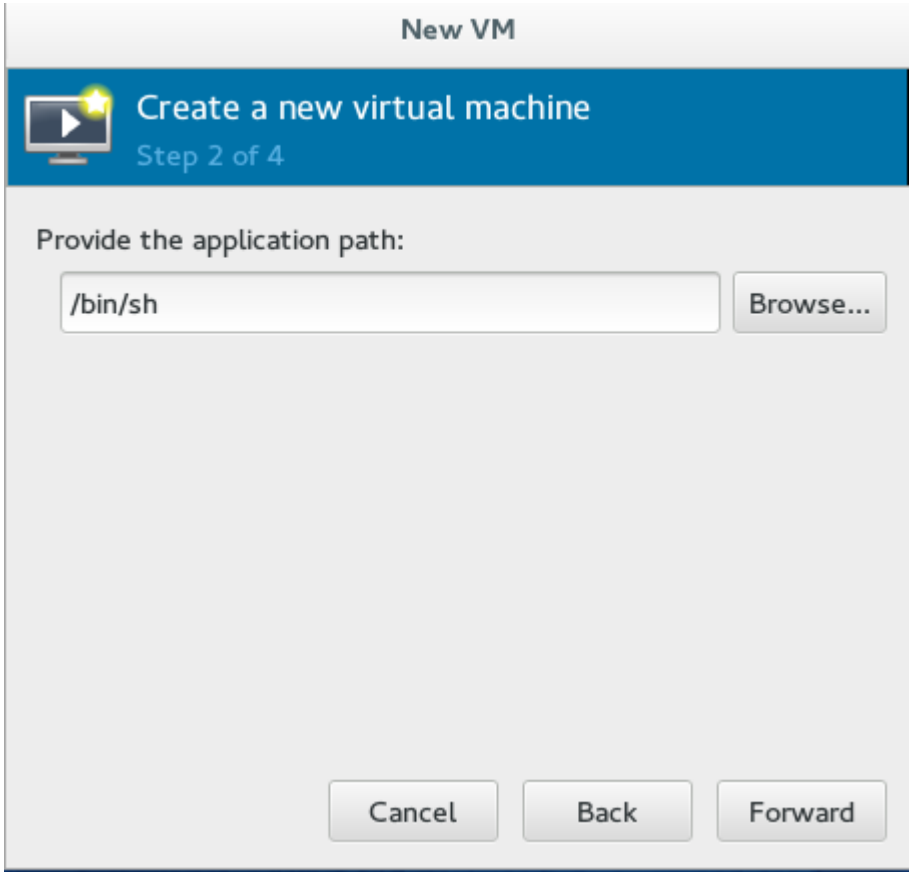


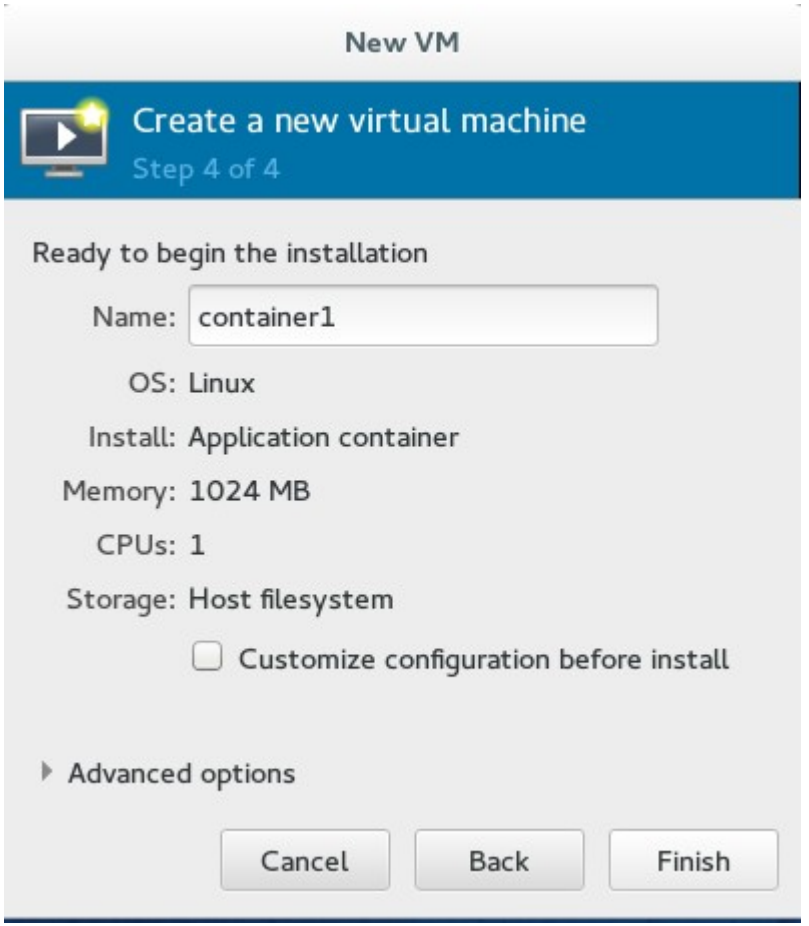

Virtual Machine Manager
File
Edit
View
Help
Add Connection...
Close
Ctrl+W
Quit
Ctrl+Q
CPU usage
rhel6
Running
rhel7
Running

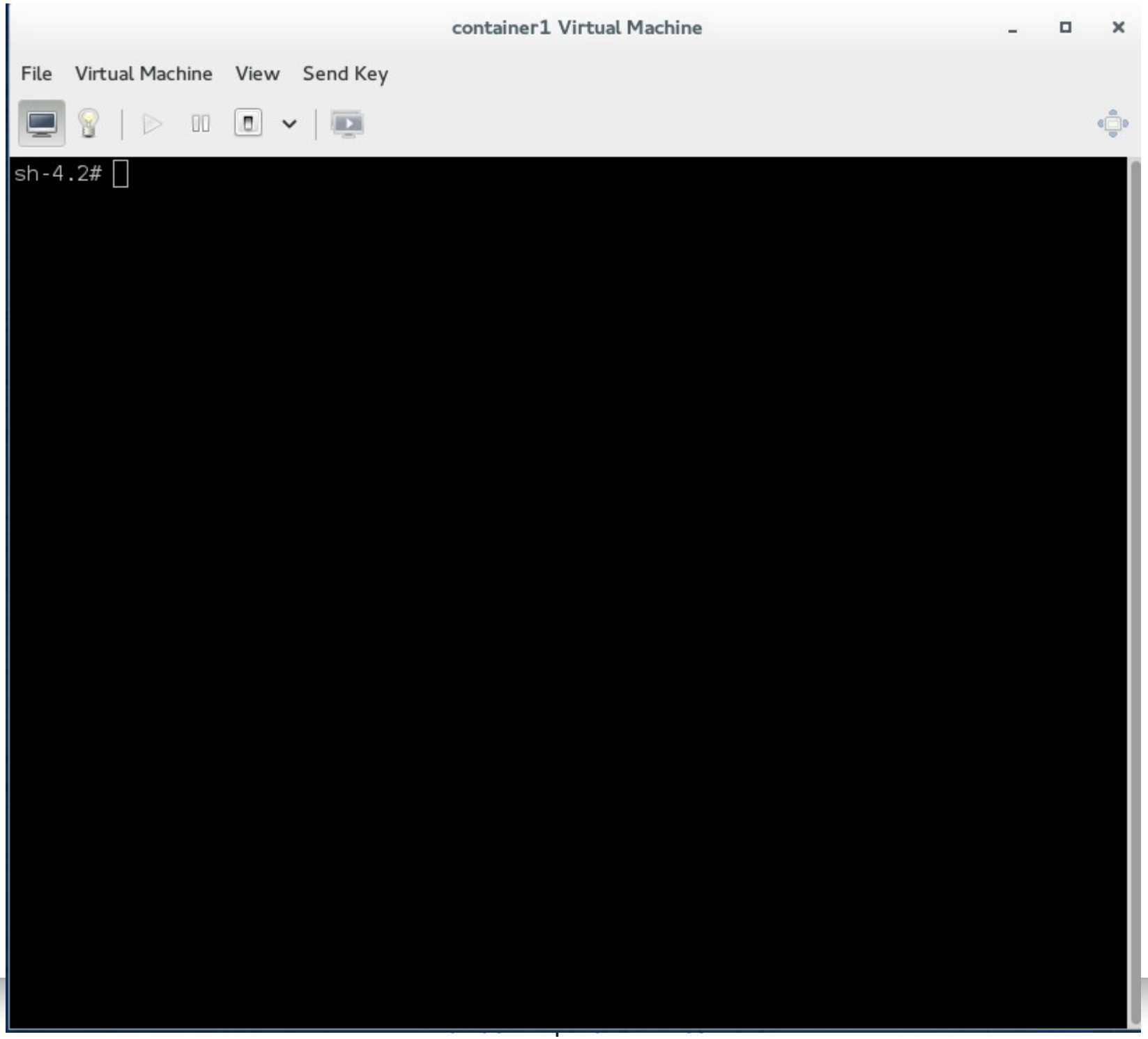

Add Connection
Hypervisor:
LXC (Linux Containers)
Connect to remote host
Method:
SSH
Username:
root
Hostname:
Autoconnect:
Generated URI: lxc:///
Cancel
Connect


New VM
Create a new virtual machine
Step 1 of 5
Connection:
localhost (LXC)
Choose the container type
Application container
Operating system container
Cancel
Back
Forward


New VM
Create a new virtual machine
Step 2 of 4
Provide the application path:
/bin/sh
Browse...
Cancel
Back
Forward


New VM
Create a new virtual machine
Step 2 of 4
Provide the application path:
/bin/sh
Browse...
Cancel
Back
Forward


New VM
Create a new virtual machine
Step 4 of 4
Ready to begin the installation
Name: container1
OS: Linux
Install: Application container
Memory: 1024 MB
CPUs: 1
Storage: Host filesystem
Customize configuration before install
Advanced options
Cancel
Back
Finish


container1 Virtual Machine
File
Virtual Machine
View
Send Key
sh-4.2#

LXC DEMO USING THE COMMAND LINE

```
# virsh uri
qemu:/// session
# export VIRSH_DEFAULT_CONNECT_URI=lxc:///
# virsh uri
lxc:///
```

VALIDATE LXC CAPABILITIES

virsh capabilities

```
<host>
  <uuid>8122b826-6452-cb11-af46-b502e54fad4c</uuid>
  <cpu>
    <arch>x86_64</arch>
  </cpu>
  <power_management>
    <suspend_mem/>
    <suspend_disk/>
  </power_management>
  <topology>
    <cells num='1'>
      <cell id='0'>
        <cpus num='4'>
          <cpu id='0' socket_id='0' core_id='0' siblings='0-1'/>
          <cpu id='1' socket_id='0' core_id='0' siblings='0-1'/>
          <cpu id='2' socket_id='0' core_id='1' siblings='2-3'/>
          <cpu id='3' socket_id='0' core_id='1' siblings='2-3'/>
        </cpus>
      </cell>
    </cells>
  </topology>
</host>
```

CONFIGURE A CONTAINER

vi lxc_example.xml

```
<domain type='lxc'>
  <name>lxc_example</name>
  <memory>500000</memory>
  <os>
        <type>exe</type>
        <init>/bin/sh</init>
  </os>
  <vcpu>1</vcpu>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
         <emulator>/usr/libexec/libvirt_lxc</emulator>
               <interface type='network'>
                     <source network='default'/>
               </interface>
        <console type='pty' />
  </devices>
</domain>
```

DEFINE AND START THE CONTAINER

```
# virsh define lxc_example.xml
# virsh start lxc_example
# virsh list
# virsh dominfo lxc_example
# virsh console lxc_example
```

DEMO WITH VIRTSANDBOX - HTTPD

```
# yum install libvirt-sandbox httpd
# systemctl restart libvirtd
# virt-sandbox-service create -C --network dhcp -u httpd.service httpd
# systemctl start  httpd_sandbox
# virt-sandbox-service connect httpd
# dhclient eth0
# ifconfig
```

Point browser on http:///(ip-address)

note : /var/lib/libvirt/filesystems


NEXT STEP

DOCKER

- Container-based tooling
- High level tool for LXC
- Portable deployment across machines
- Public shared containers
- Automatic build
- Tool ecosystem (nova, salt, chef, puppet, jenkins, openshift ...)
- And more ...

  http://docker.io

DOCKER DEMO (Fedora 20)

```
# yum install docker-io
# sytemctl start docker ; systemctl enable docker
# docker search ubuntu
# docker pull ubuntu
# sudo docker run ubuntu apt-cache search memcached
# docker search arch
# docker pull base/arch
# docker run base/arch pacman -Ss memcached
# docker images
# docker run -i -t ubuntu /bin/bash
# docker ps -a
# docker rmi base/arch
```


redhat