Virtual Server Management

with KVM for IBM z Systems

Viktor Mihajlovski <mihajlov@de.ibm.com>

Agenda

- KVM virtualization components
- Managing virtual servers
- Managing virtual resources
- Migrating running virtual servers
- Setting up virtual servers

KVM Virtualization Components

Host resource management
Virtual server life cycle

libvirt

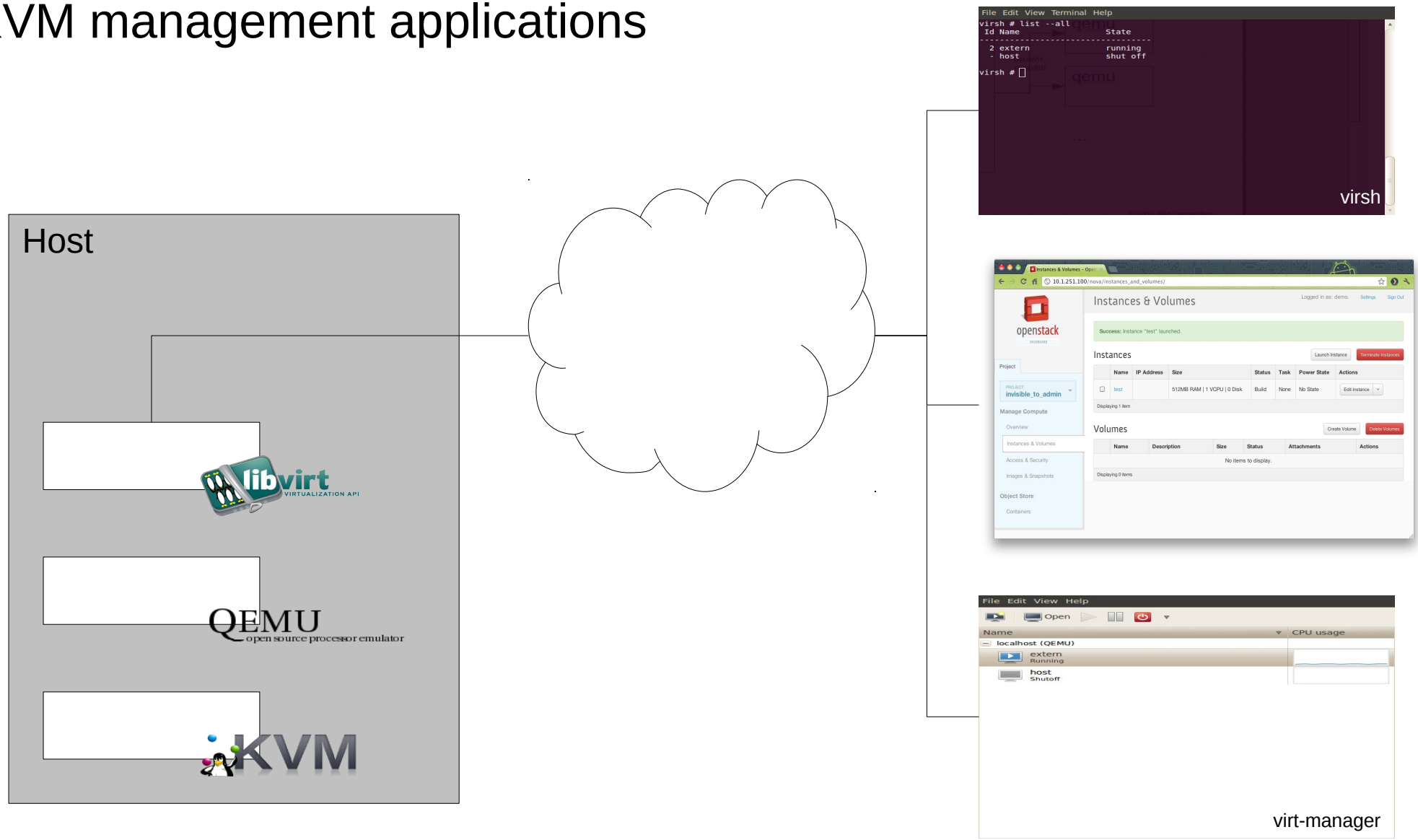
I/O virtualization
CPU and memory support

QEMU

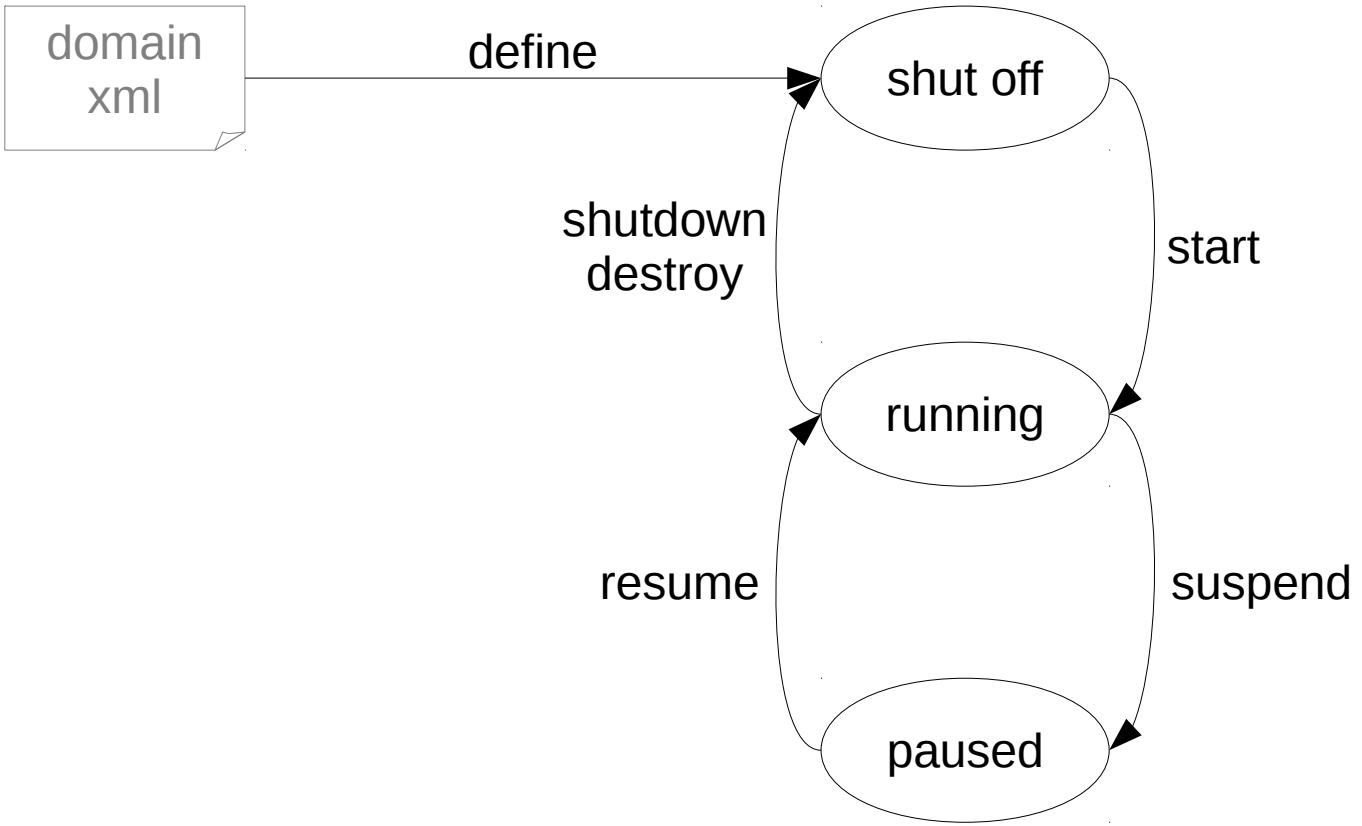
CPU and memory

kvm kernel module

KVM management applications

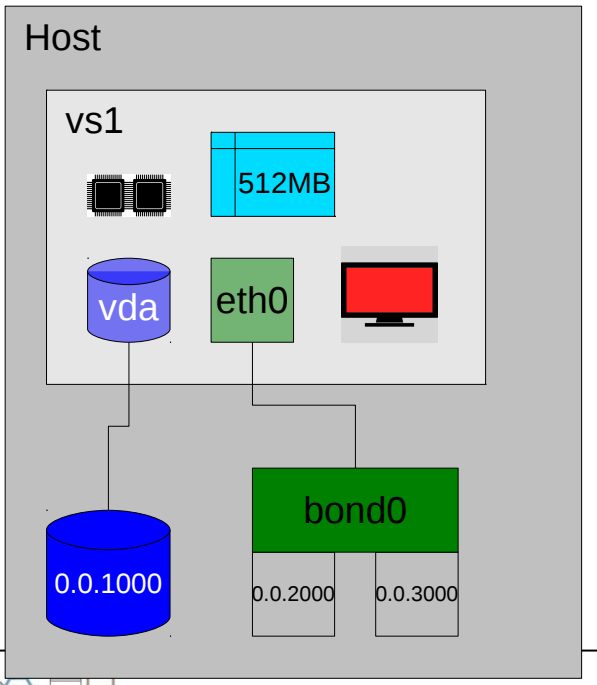


Virtual server states (simplified)



Defining a virtual server: the domain XML

```
<domain type='kvm'>
  <name>vs1</name>
  <memory>524288</memory>
  <vcpu>2</vcpu>
  <os>
    <type arch='s390x' machine='s390-ccw-virtio'>hvm</type>
  </os>
  <iothreads>1</iothreads>
  <devices>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' iothread='1' io='native'/>
      <source dev='/dev/disk/by-path/ccw-0.0.1000'/>
      <target dev='vda' bus='virtio'/>
    </disk>
    <interface type='direct'>
      <source dev='bond0' mode='bridge'/>
      <model type='virtio'/>
    </interface>
    <console type='pty'>
       <target type='sclp'/>
    </console>
  </devices>
</domain>
```



Defining a virtual server with virsh

Step 1: Make sure the libvirt daemon is running

```
# systemctl status libvirtd
libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled)
   Active: active (running) since Thu 2015-04-16 10:55:29 CEST; 2 months 3 days ago
     Docs: man:libvirtd(8)
           http://libvirt.org
 Main PID: 5615 (libvirtd)
   CGroup: /system.slice/libvirtd.service
           ├─5615 /usr/sbin/libvirtd
           ├─6750 /sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf  ...
           └─6751 /sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf  ...
```

Step 2: Define a virtual server

```
# virsh define vs1.xml
Domain vs1 defined from vs1.xml
```

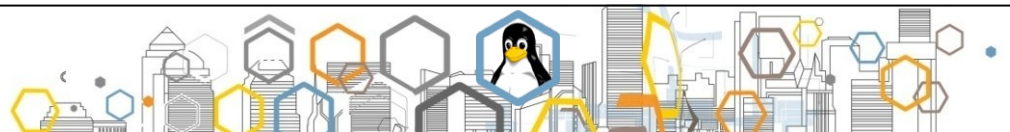
Starting and stopping a virtual server with virsh

Start a virtual server

```
# virsh start vs1
Domain vs1 started
```

Shut down a virtual server

```
# virsh shutdown vs1
Domain vs1 is being shut down
```



Monitoring a virtual server with virsh

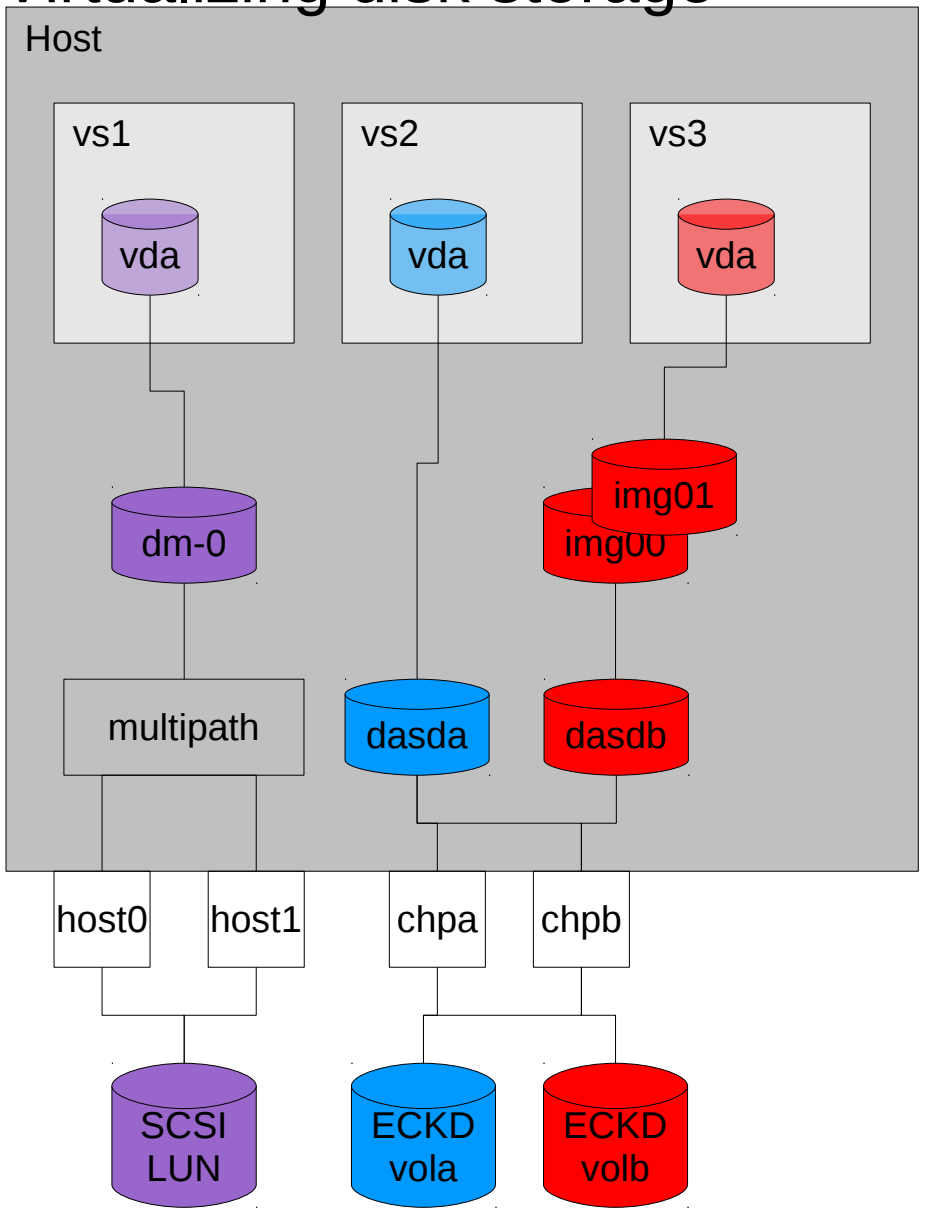
List running virtual servers

```
# virsh list
 Id    Name                           State
----------------------------------------------------
 11    vs1                            running
```

Display virtual server details

```
# virsh dominfo vs1
Id:             11
Name:           vs1
UUID:           ea6de929-a4df-487d-8054-d50b4af20708
OS Type:        hvm
State:          running
CPU(s):         2
CPU time:       266.6s
Max memory:     524288 KiB
Used memory:    524288 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: selinux
Security DOI:   0
Security label: system_u:system_r:svirt_t:s0:c24,c532 (enforcing)
```


Virtualizing disk storage

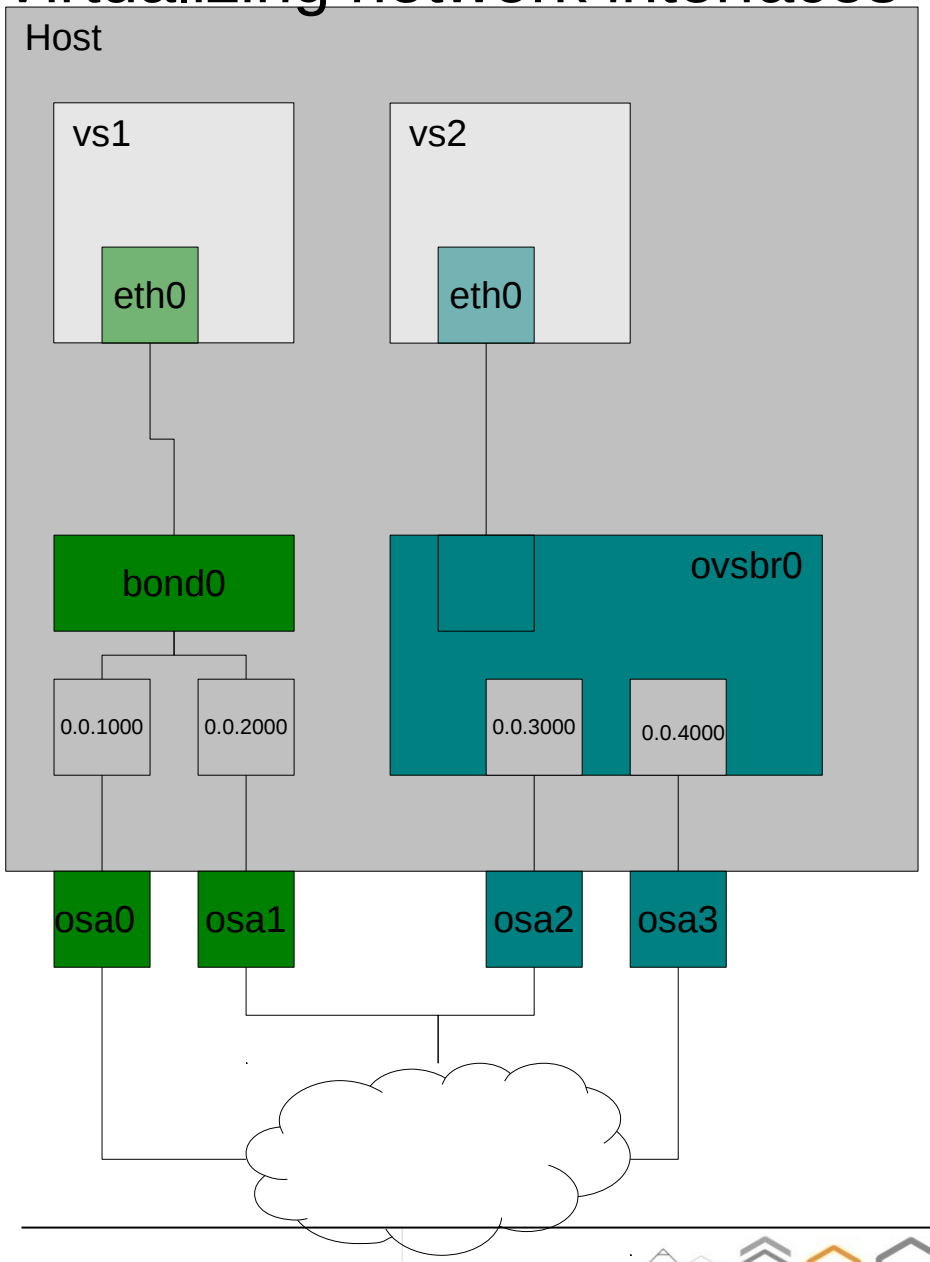



```
<disk type='file' device='disk'>
      <source file='/var/libvirt/images/img01'/>
      <driver name='qemu' type='raw' io='native'
          cache='none' iothread='1'/>
      <target dev='vda' bus='virtio'/>
</disk>
```

```
<disk type='block' device='disk'>
      <source dev='/dev/dasda'/>
      <driver name='qemu' type='raw' io='native'
          cache='none' iothread='1'/>
      <target dev='vda' bus='virtio'/>
</disk>
```

```
<disk type='block' device='disk'>
      <source dev='/dev/dm-0'/>
      <driver name='qemu' type='raw' io='native'
          cache='none' iothread='1'/>
      <target dev='vda' bus='virtio'/>
</disk>
```

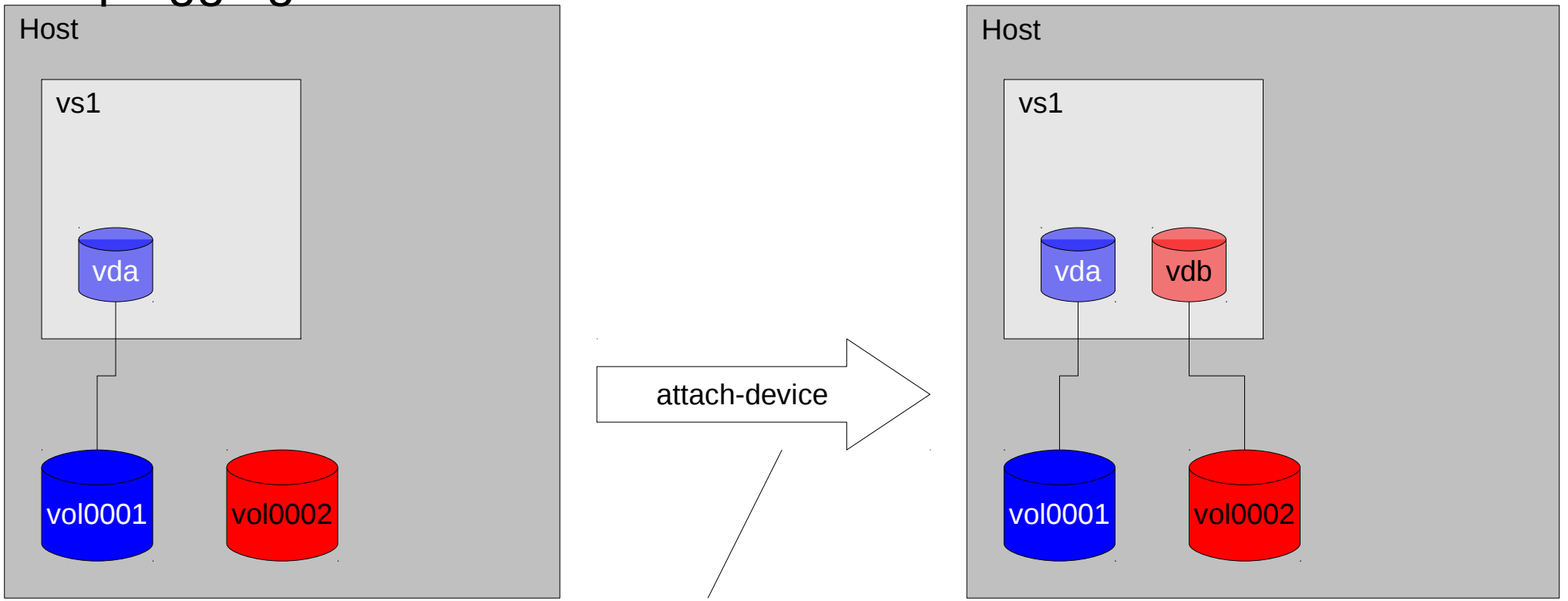
Virtualizing network interfaces




```
  <interface type='direct'>
        <source dev='bond0'/>
        <model type='virtio'/>
        <mac address='47:11:af:fe:08:15'/>
  </interface>
```

```
<interface type='bridge'>
      <source bridge='ovsbr0'/>
      <model type='virtio'/>
      <virtualport type='openvswitch'/>
      <mac address='be:ef:ca:fe:fe:ed'/>
</interface>
```

Hotplugging devices




```
<disk type='file' device='disk'>
       <source file='/var/libvirt/images/vol0002'/>
       <driver name='qemu' type='raw' io='native'
            cache='none' iothread='1'/>
       <target dev='vdb' bus='virtio'/>
</disk>
```

Attaching a second disk with virsh

Before attach

```
# virsh domblklist vs1
Target     Source
------------------------------------------------
vda        /dev/disk/by-path/ccw-0.0.1000
```

Attach

```
# cat vdb.xml
<disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' iothread='1' io='native'/>
      <source dev='/dev/disk/by-path/ccw-0.0.2000'/>
      <target dev='vdb' bus='virtio'/>
</disk>
# virsh attach-device –live vs1 vdb.xml
Device attached successfully
```

After attach

```
# virsh domblklist vs1
Target     Source
------------------------------------------------
vda        /dev/disk/by-path/ccw-0.0.1000
vdb        /dev/disk/by-path/ccw-0.0.2000
```

Virtual Server Trouble Shooting

- <on_crash>
  - restart: potential reboot loops
  - destroy: default, allows simple external automation
  - preserve: allows detailed analysis

Observing a crash

Before crash

```
# virsh domstate --reason vs1
running (booted)
```

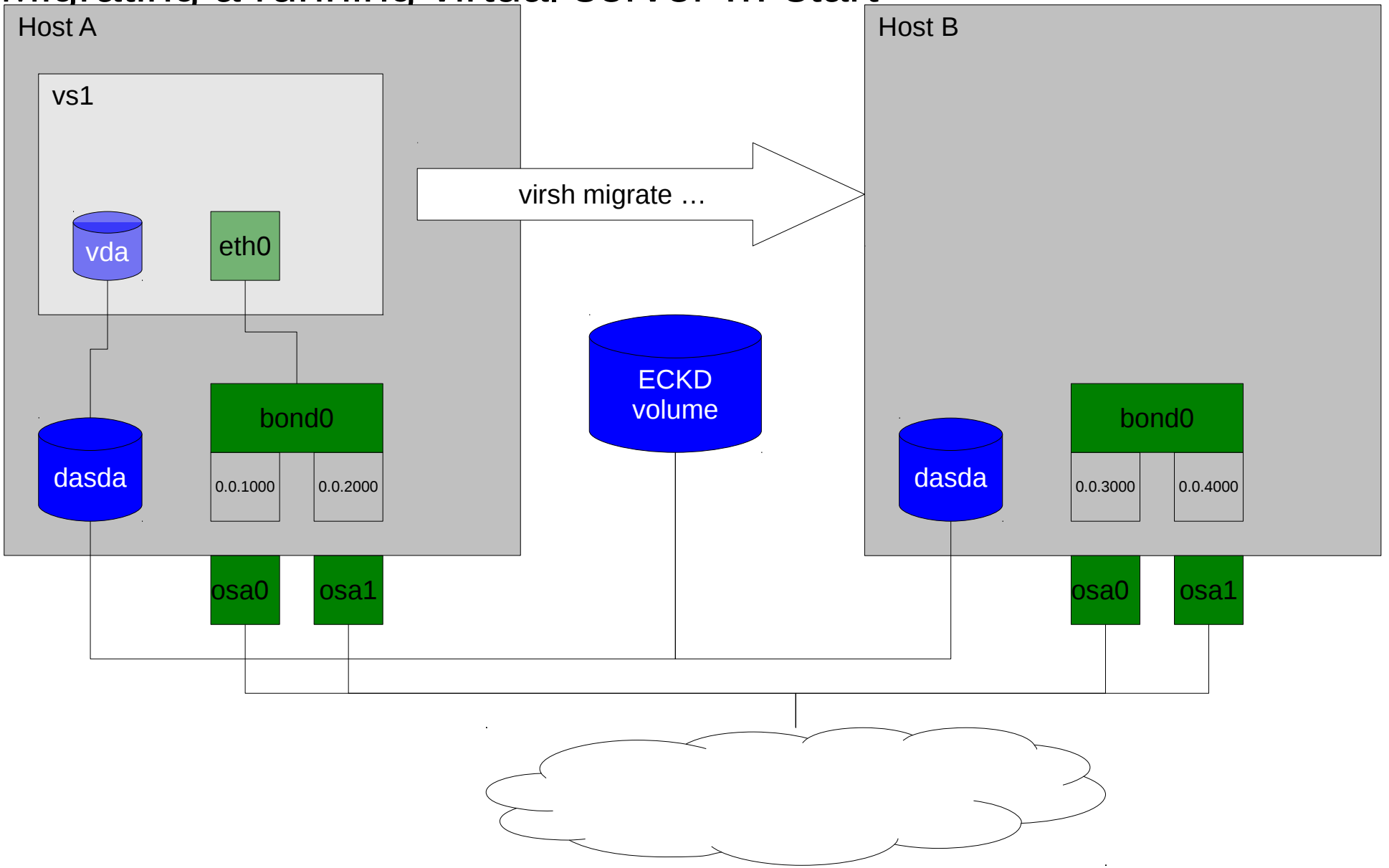
Provoke crash in virtual server

```
# virsh console vs1
~ # echo c > /proc/sysrq-trigger
[ 7056.393138] sysrq: SysRq : Trigger a crash
…
…
[ 7056.393302] Kernel panic - not syncing: Fatal exception: panic_on_oops
```

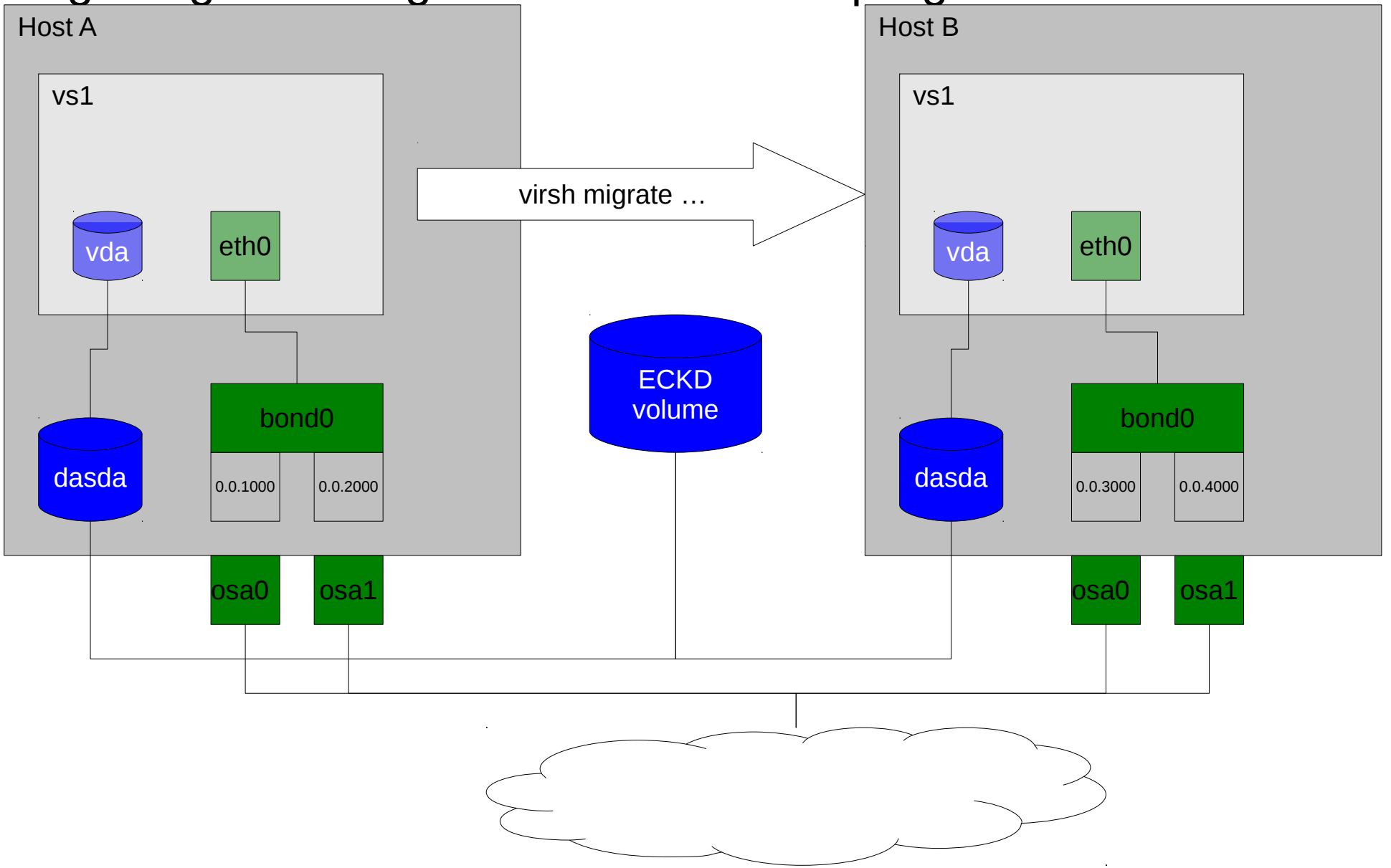
After crash

```
# virsh domstate --reason vs1
crashed (panicked)
```

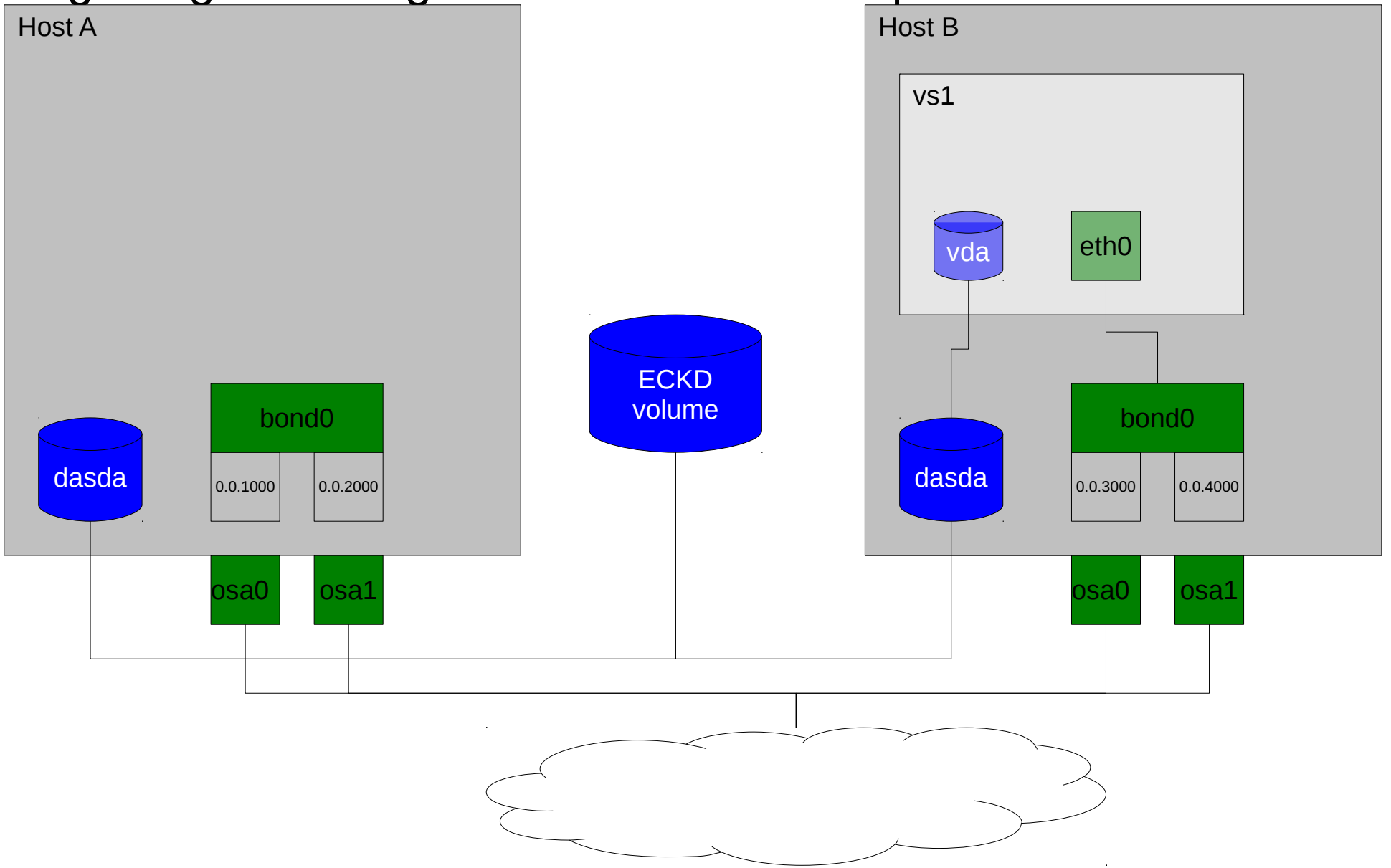
Migrating a running virtual server … start



Migrating a running virtual server … in progress



Migrating a running virtual server … completed



Converting an existing Linux installation

- Starting point: Linux installed on SCSI disk
- Perform a backup FIRST
- Modification needed:
  - Mount filesystems by id or label NOT by block device name
  - Same for boot configuration (root=...)

```
# fstab
. . .
/dev/mapper/mpath0  /    ext3 defaults  0 0
. . .
```

```
# fstab
. . .
LABEL=/   /    ext3 defaults  0 0
. . .
```

Converting an existing installation ...

- Modify boot configuration
  - Use LABEL=/ in zipl.conf or grub.conf
  - Re-run *zipl* if necessary
- Set up hypervisor resources for virtual servers
  - Ensure root disk is accessible from hypervisor
  - Provide network connectivity
- Define a virtual server
  - Write domain XML and run *virsh define ...*
- Start the virtual server
  - Run *virsh –console start …*
- Log in to the console and perform the necessary customizations
  - Network interface configuration
  - Additional file systems
  - ...

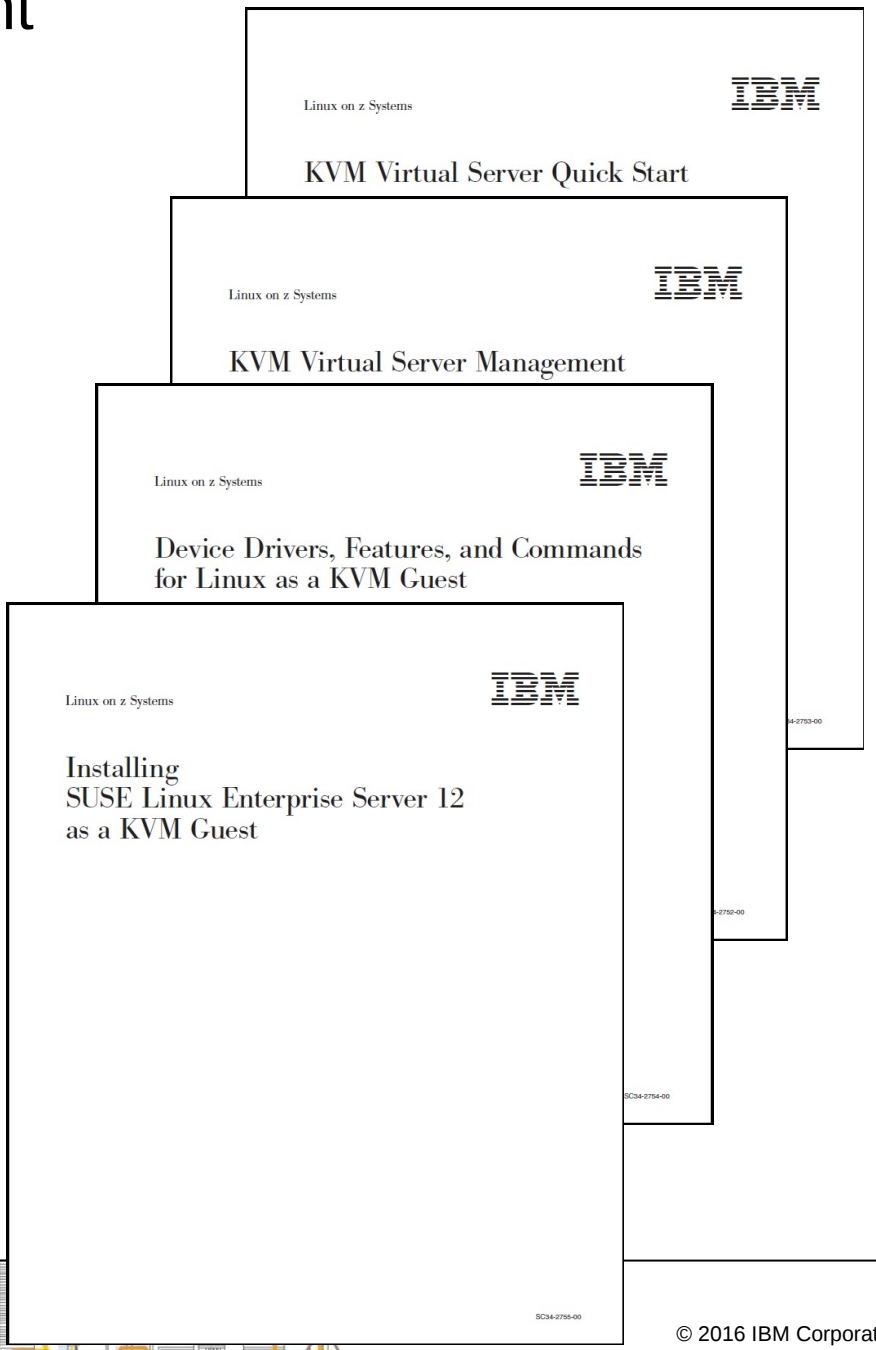
Installing a new guest

- Similar to z Systems LPAR installation
  - FTP Server with installation media
  - Boot from installaton kernel and RAM disk
- Steps
  - Copy kernel and initial RAMdisks to hypervisor (e.g. /var/lib/libvirt/images)
  - Define virtual server using direct kernel boot
    - For possible command line options, see the Linux distributor's documentation
  - After completing the installation, update the virtual server definition to boot from disk

```
<domain type='kvm'>
  <name>zguest</name>
  <memory>524288</memory>
  <vcpu>2</vcpu>
  <os>
    <type arch='s390x' machine='s390-ccw-virtio'>hvm</type>
    <kernel>/var/lib/libvirt/kernel</kernel>
    <initrd>/var/lib/libvirt/image</initrd>
    <cmdline>Install=ftp://user:password@server/directory/DVD1/ ...</cmdline>
  </os>
 ...
 ...
</domain>
```

Publications for KVM base component

- KVM Virtual Server Quick Start, SC34-2753
- KVM Virtual Server Management, SC34-2752
- Device Drivers, Features, and Commands for Linux as a KVM Guest, SC34-2754
- Installing SUSE Linux Enterprise Server 12 as a KVM Guest, SC34-2755



ibm.com/support/knowledgecenter

THANKS!!!

Questions?

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

The following are trademarks or registered trademarks of other companies.

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Backup

Terminology used in this presentation

- KVM virtual server, virtual server

  virtualized z Systems resources that comprise processor, memory, and I/O capabilities as provided and managed by KVM. A virtual server can include an operating system.

- KVM guest, guest

  an operating system of a virtual server.

- KVM host, host

  the Linux instance that runs the KVM virtual servers and manages their resources.

  - In libvirt context also
    - node (host)
    - domain (virtual server)