

Migrating with virsh

First, my setup.

We have a couple of semi-new servers for hypervisors, but we also wanted a couple of backup machines so we could migrate to them, do maintenance, then migrate back. So, a real mixture of cpu's.

All of our backend stores are on an iSCSI server which is **not** managed by libvirt. However, all of the hypervisors are running a fairly recent version of [Devuan Linux](#), which is a version of Debian without System-D, so the paths to the block devices are all the same. Devuan/Debian creates a soft link in /dev/disk/by-path/ with human decipherable names which don't change (so far).

Baseline CPU Definition

The different CPU's cause a problem, so I needed a baseline cpu definition for the virtuals that would work on all of the different hypervisors. Red Hat has a way of doing it, but their explanation stinks, so here is a recipe that works as of today (2022).

By default, if you do not define anything special, your xml file (the virtual definition) will define your cpu's to use whatever capabilities your host has. The code block, inside of your definition, looks like this.

```
--other stuff
<cpu mode='host-model' check='partial'>
  <model fallback='allow' />
</cpu>
-- other stuff
```

However, this will keep you from migrating if the target machine does not have the same, or a superset of, the capabilities your source has. To get around this, we will replace the above block with a cpu definition that will work on all of the machines.

1. On all hypervisor machines, run the following command. This creates a capabilities file (with your machines hostname as the filename)

```
1. virsh capabilities > `hostname`.capabilities.xml
```

2. Copy all files created to one of the machines
3. Combine the files, then run cpu-baseline on the result

```
cat *.xml > all.xml && virsh cpu-baseline all.xml > common.xml
```

4. Edit each of your virtual definitions, replacing the <cpu></cpu> block with the contents of common.xml.

When I ran this comparing two machines with an Intel Xeon X5650 and a Xeon E5-2430, the result was

```
<cpu mode='custom' match='exact' check='full'>
  <model fallback='forbid'>Nehalem</model>
  <vendor>Intel</vendor>
  <feature policy='require' name='vme' />
  <feature policy='disable' name='ds' />
  <feature policy='disable' name='acpi' />
  <feature policy='require' name='ss' />
  <feature policy='disable' name='ht' />
  <feature policy='disable' name='tm' />
  <feature policy='disable' name='pbe' />
  <feature policy='disable' name='dtes64' />
  <feature policy='disable' name='monitor' />
  <feature policy='disable' name='ds_cpl' />
  <feature policy='disable' name='vmx' />
  <feature policy='disable' name='smx' />
  <feature policy='disable' name='est' />
  <feature policy='disable' name='tm2' />
  <feature policy='disable' name='xtpr' />
  <feature policy='disable' name='pdc_m' />
  <feature policy='require' name='pcid' />
  <feature policy='disable' name='dca' />
  <feature policy='require' name='arat' />
  <feature policy='require' name='pdpelgb' />
  <feature policy='require' name='rdtscp' />
  <feature policy='require' name='x2apic' />
  <feature policy='require' name='hypervisor' />
</cpu>
```

I literally copied that block, did a *virsh edit* on each of my virtuals, found and deleted the existing `<cpu></cpu>` block, and pasted the above in.

Now, all my virtuals are set up to use a common set of cpu capabilities.

Get the correct version of netcat

On my Devuan machine, installing *netcat* gives me something that is not compatible with what *virsh migrate* needs. You need the BSD version of netcat, which you can install with the command:

```
apt -y install netcat-openbsd
```

Failing that, you will get an error about an incompatible netcat (nc) which does not have the -U flag.

Set up root access between hypervisors

From:

<https://wiki.linuxservertech.com/> - **Unix Server Tech Knowledge Base**

Permanent link:

<https://wiki.linuxservertech.com/unix/virtualization/virtlib/migrate?rev=1652941939>

Last update: **2022/05/19 06:32**

