

kvmctl

Linux KVM controller

September 24, 2009

kvmctl is how we manage our small "virtual datacenter".

It is experimental software, as the user interface is not stabilized yet.

1 The `kvmctl` environment

kvmctl makes the the following assumptions about its environment:

- Each virtual machine has its own parent directory (DRBD mountpoints in our case).
- In each directory resides the image and a `kvm.cfg` configuration file

2 `kvm.cfg` format

In `kvm.cfg`, the following options are mandatory:

- `IMAGE` (Mandatory) The absolute path to the image file.
- `MAC` (Mandatory) A unique MAC address for the VM.
- `NUMID` (Mandatory) A unique ID which is used for calculating VNC and monitor ports.
- `MEMORY` (Mandatory) Amount of memory that will be assigned to the VM.
- `MOREOPTS` (Optional) Additional options that will be passed to KVM.

```
IMAGE=/drbd/foo/foo.vmdk
MAC=00:16:3e:12:34:56
MEMORY=512
NUMID=4
MOREOPTS="-cdrom /path/to/cd.iso -boot d"
```

Listing 1: `kvm.cfg` example

3 `kvm.cfg` syntax

Currently, there are two possible invocations for `kvmctl`:

- `kvmctl list` - Prints a list of all VMs that are currently under control of `kvmctl`.
- `kvmctl <image_dir> <command>` - Applies *command* to the image contained in *image_dir*.

The syntax for the "command" invocation is subject to change in the near future.

4 `kvmctl` commands

4.1 `start`

`kvmctl <image_dir> start` boots the image contained in *image_dir*.

The started VM is backgrounded. VNC is available at (5900+NUMID) and the KVM monitor in raw TCP mode is available at (1000+NUMID).

4.2 `stop`

`kvmctl <image_dir> stop` sends a poweroff signal to the VM running from the image in *image_dir*, usually resulting in a clean shutdown.

4.3 `kill`

FIXME: Not implemented.

`kvmctl <image_dir> kill` kills the VM running from the image in *image_dir*.

4.4 `status`

`kvmctl <image_dir> status` prints a brief summary of the status of the VM running from the image in *image_dir*.

4.5 `vnc`

`kvmctl <image_dir> vnc` launches a VNC viewer and connects it to the VM running from the image in *image_dir*.

4.6 `monitor`

`kvmctl <image_dir> vnc` connects to the KVM/Qemu monitor of the VM running from the image in *image_dir*.

5 Bugs and Limitations

- Support for live migration is not planned.
- The syntax for the "command" invocation is subject to change in the near future.

A Links

- Python script for generating random MAC addresses
<http://tinyurl.com/macgen-py>
- Qemu-Dokumentation
<http://www.qemu.org/qemu-doc.html>
- Qemu-KVM-Wiki
<http://qemu-buch.de/de/index.php/Hauptseite>
- KVM Howto
<http://www.linux-kvm.org/page/HOWTO>
- KVM FAQ
<http://www.linux-kvm.org/page/FAQ>

B FYI: Interface configuration

Here's how we configure our interfaces on CentOS 5:

```
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
BRIDGE=br0
HWADDR=00:...
```

Listing 2: ifcfg-eth0 - The ethernet interface

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
ONBOOT=yes
TYPE=Bridge
BROADCAST=192.168....
IPADDR=192.168....
NETMASK=255.255.255.0
NETWORK=192.168....
```

Listing 3: ifcfg-br0 - The bridge

Scripts for pulling up *tap* devices are in the *tapscripts* directory. These should go to */etc*.

C Copyright and license

Copyright (c) 2009, Martin Schmitt <mas at scsy dot de>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RE-

SULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.