kantan(8)

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Name

kantan - simple test suite for autotesting software using Grml and KVM

2 Synopsis

For the server (main) instance:

```
kantan server <disk.img> <grml.iso> </mnt/point/of/iso/> [kvm_arguments]
```

or for client(s):

```
kantan client <disk.img> <name> [kvm_arguments]
```

For further details please see section Section 3 and Section 6.

3 Options

Options for *kantan server*:

```
<disk.img> <grml.iso> </mnt/point/of/iso/> [kvm_arguments]
```

where *disk.img* is an existing image file (created e.g. via *qemu-img create disk.img 3G*), grml.iso is an a Grml ISO (http://grml.org/-), /mnt/point/of/iso/ refers to the mount point where the specified Grml ISO is mounted to (e.g. *mount -o loop grml-medium_sid_latest.iso /mnt/test*) and *kvm_arguments* are optional and can be used to add additional arguments to the KVM commandline, like *vga=791*.

Tip

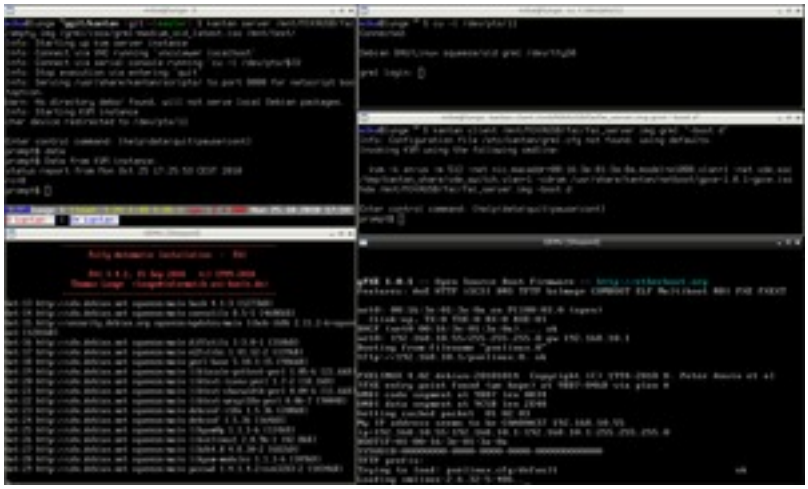
The </mnt/point/of/iso/> doesn't necessarily need to be a real mountpoint. It's fine if you just copy linux26 and initrd.gz from /boot/* of the Grml ISO to </path/to/foobar/boot/> and specify </path/to/foobar/> as mountpoint directory then.

Options for *kantan client*:

```
<disk.img> <name> [kvm_arguments]
```

where *disk.img* is an existing image file (created e.g. via *qemu-img create disk.img 3G*), *name* assigns the virtual instance a name (for reading /etc/kantan/name.cfg iff the file exists). The *kvm_arguments* are optional and can be used to add additional arguments to the KVM commandline, like *-boot d* for booting from the ISO.

4 Description



Kantan is a set of scripts to automatically test software using KVM (http://www.linux-kvm.org/) and the Debian based Linux live system Grml (http://grml.org/). You don't have to set up neither tons of software packages nor a complex environment but just follow some simple setup instructions.

It was originally developed to provide a simple way for autotesting specific versions of FAI (http://fai-project.org/) but is expected to provide a simple-to-use test suite for autotesting any Linux based software.

The concept of Kantan is to boot one virtual machine (the main instance, *kantan server* . . .) which is supposed to be able to serve as installation server for further virtual machines if needed. This concept for example allows to automatically set up a FAI server which then deploys further started virtual machines (the clients, *kantan client* . . .).

Kantan is meant to be a prototype implementation to show how automated testing could be realised. In the long run the author would like to get a suite to automatically test:

- complete Debian installations
- testing Debian packages within current state of Debian suites
- software configurations that are required to run on more than one host (installation procedures with dependencies between different systems during installation, cluster + HA setups,. . .)
- live systems with their flexible boot options, specific features and services
- kernel versions
- software packages that provide unit tests
- low-level tools like partitioning, LVM, mdadm,. . .
- filesystems (crash, repair, mount,. . .)

Caution
This software is WIP, though released in an early stage to gain feedback, testers and developers. It's a prototype implementation mainly in shellscript but is expected to be rewritten in a real scripting language like python.

5 Requirements

What you need to be able to use Kantan:

- a Debian (based) system (any other Linux system might do it as well, but the scripts and docs are optimised for Debian (based) systems for now)
- a system running i386 or amd64 architecture (to smoothly use KVM)
- at least 2GB of RAM are recommended
- at least 4GB of harddisk space for >=2 virtual disk images
- a Grml ISO (see instructions below)

6 Setup and Usage Instructions

Install the kantan Debian package, which for now is available from http://people.debian.org/~mika/kantan/

Grab a Grml ISO and mount it somewhere:

```
% wget download.grml.org/grml64-medium_2010.04.iso
% wget download.grml.org/grml64-medium_2010.04.iso.md5
% md5sum -c grml64-medium_2010.04.iso.md5
% sudo mount -o loop grml64-medium_2010.04.iso /mnt/test
```

If necessary adjust /etc/kantan/server.cfg according to your needs.

Create image files for use as /srv within FAI server and as harddisk for the FAI client (qemu-img is available from qemu-utils):

```
% qemu-img create fai-server.img 3G
% qemu-img create fai-client.img 3G
```

If you want to provide any local Debian packages to the FAI server instance (like for example a specific version of FAI you'd like to test) just create the directory *debs* in the current working directory (being the one where you'll invoke kantan later on), like for example:

```
% mkdir debs
% dget -u -d http://fai-project.org/download/squeeze/fai_3.4.4_amd64.changes
% mv fai*.deb debs/
```

Finally execute the kantan script as $USER for the server instance, providing the path to the generated image file, the Grml-ISO and the mountpoint where the Grml ISO is mounted on:

```
% kantan server fai-server.img grml64-medium_2010.04.iso /mnt/test
```

Finally start the FAI client instance (the one that should be installed by the kantan server VM), specifying *grml* as client name (so /etc/kantan/grml.cfg would be read if it exists) and "-boot d" as boot option for KVM so it uses the PXE ISO for booting (just drop "-boot d" then when installation of client has finished):

```
% kantan client fai-client.img grml "-boot d"
```

That's it! :) Further usage scenarious and tests will follow.

7 FAQ

7.1 Why Kantan?

The author of Kantan thinks that lack of proper Q/A is one of the most annoying issues in the open source world. So let's do something against that.

7.2 What does Kantan mean?

The author of Kantan is a friend of the Kanban concept and created the word Kantan based on "the Kanban of testing". Amusingly according to http://www.cjvlang.com/Writing/writjpn/signs/kantan.html "Kantan is a Chinese-style compound (on-reading) meaning *simple*." which represents the idea of Kantan: provide a *simple* method for testing software.

7.3 Why use a VM and not just a chroot?

Chroots provide a nice way for testing stuff like package installations. But they are limited to some restricted actions. Chroots don't properly support testing kernel versions, bootoptions, partitioning tools, LVM, mdadm,... in a reasonable environment.

7.4 How can I use Kantan with physical machines?

By default Kantan uses vde_switch for network configuration. This provides a working network setup between the server and the client(s) machines without having to configure anything. But if you want to use external, physical machines this does not work any longer. Instead just set up tap devices so you can install clients on real[tm] hardware.

Assume the following setup: the Kantan server is the server where you're executing "kantan server ...". The test client is the machine where you want to boot the system which would be corresponding to "kantan client ..." (but instead of executing "kantan client ..." as virtual guest you're running it on physical machine). NIC eth0 is providing internet access (WAN). NIC eth1 is the network link between the server and the client machine.

```
+-----------------+
| Kantan    ,-----|----- eth0 ----- [ Internet/WAN ]
| Server    |     |
|         [tap1]  |
|           |     |
|           `-----|----- eth1 ----- [ Test client ]
+-----------------+
```

Then the following configuration should do the trick for you:

```
# example config for /etc/network/interfaces
iface vnet inet static
      post-up tunctl -u grml -g grml -t tap1 ; brctl addif vnet tap1 ; ip link set up dev
      post-up brctl addif vnet eth1
      pre-down ip link set down dev tap1 ; tunctl -d tap1
      bridge_ports none
      address 192.168.10.2
      netmask 255.255.255.0
```

Configure /etc/kantan/server.cfg:

```
VLAN1_DEVICE='tap1'
```

Then execute:

```
ifup vnet
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

To forward DNS requests from the clients to the physical net use dnsmasq:

```
/etc/init.d/dnsmasq start || apt-get install dnsmasq
```

8 TODOs

8.1 Known todos

- provide test features and tests from grml-unittests
- better logging and data sharing support to collect data and status report and control data flow between different virtual machines
- make sure vnc and serial console display the same things during execution within Grml
- netscript.sh:
  - support svn/git/... config space retrieval
  - improve arch and suite support through base.tgz
  - support canceling the script and getting a debugshell

8.2 Checkout

- use kvm's monitor support for sharing/controlling data?
- investigate and combine/merge features/ideas/approaches from:
  - http://kvm.et.redhat.com/page/KVM-Autotest
  - http://lizards.opensuse.org/2010/05/25/automated-opensuse-testing/
  - http://git.grml.org/?p=grml-unittests.git;a=summary
  - http://www.mozilla.org/projects/testopia/
  - https://wiki.ubuntu.com/AutomatedTesting
  - https://wiki.ubuntu.com/AutomatedTestingDeployment
  - https://wiki.ubuntu.com/Testing/ISO/Procedures
  - https://wiki.edubuntu.org/VirtFeatureVerification
  - http://wiki.virtualsquare.org/wiki/index.php/VDE_Basic_Networking#Dump_or_Monitor_switch_traffic
  - http://developer.amd.com/zones/opensource/AMDTapper/Pages/default.aspx
- tools that might help in automated testing:
  - GNU LDTP
  - Selenium
  - ShUnit
  - Sikuli
  - Tuitest
  - Xautomation
  - WebTest
  - http://en.wikipedia.org/wiki/List_of_GUI_testing_tools

9 Bugs

Probably. We just need a test suite for Kantan for testing.

10 Author

Michael Prokop <mika@grml.org>