MyWebUniversity
python
Rocky Linux
Manual Pages
FPDF Library
PDF generator
FPDF
Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'virt-what.1'

$ man virt-what.1

VIRT-WHAT(1)   Virtualization Support   VIRT-WHAT(1)

NAME

virt-what - detect if we are running in a virtual machine

SUMMARY

virt-what [options]

DESCRIPTION

"virt-what" is a shell script which can be used to detect if the

program is running in a virtual machine.

The program prints out a list of "facts" about the virtual machine,

derived from heuristics. One fact is printed per line.

If nothing is printed and the script exits with code 0 (no error), then

it can mean either that the program is running on bare-metal or the

program is running inside a type of virtual machine which we don't know

about or cannot detect.

FACTS

alibaba_cloud

alibaba_cloud-ebm

This is a cloud computing service based on Alibaba Cloud.

    Status: contributed by Weisson.

aws Amazon Web Services.

    Note that virt-what will print this fact for baremetal AWS

    instances, which you might not consider to be true virtualization.

    In this case other facts (eg. "kvm" or "xen") would not be present.

    Status: contributed by Qi Guo, Vitaly Kuznetsov, confirmed by RWMJ.

bhyve

    This is a bhyve (FreeBSD hypervisor) guest.

    Status: contributed by Leonardo Brondani Schenkel.

docker

    This is a Docker container.

    Status: confirmed by Charles Nguyen

google_cloud

    This is running on Google Cloud Platform / Google Compute Engine.

    Status: confirmed by RWMJ

hyperv

    This is Microsoft Hyper-V hypervisor.

    Status: confirmed by RWMJ

ibm_power-kvm

    This is an IBM POWER KVM guest.

    Status: contributed by Adrian Likins.

ibm_power-lpar_shared

ibm_power-lpar_dedicated

    This is an IBM POWER LPAR (hardware partition) in either shared or

    dedicated mode.

    Status: contributed by Adrian Likins.

ibm_systemz

    This is an IBM SystemZ (or other S/390) hardware partitioning

    system. Additional facts listed below may also be printed.

ibm_systemz-direct

    This is Linux running directly on a IBM SystemZ hardware

    partitioning system.

    This is expected to be a highly unusual configuration - if you see this result you should treat it with suspicion.

Status: not confirmed

ibm_systemz-lpar

This is Linux running directly on an LPAR on an IBM SystemZ

hardware partitioning system.

Status: confirmed by Thomas Huth

ibm_systemz-zvm

This is a z/VM guest running in an LPAR on an IBM SystemZ hardware

partitioning system.

Status: confirmed by RWMJ using a Fedora guest running in z/VM

ibm_systemz-kvm

This is a KVM guest running on an IBM System Z hardware system.

Status: contributed by Thomas Huth

illumos-lx

The guest is running on Illumos with a Linux syscall emulation

layer.

Status: contributed by Steve Mokris

ldoms

The guest appears to be running on an Linux SPARC system with

Oracle VM Server for SPARC (Logical Domains) support.

Status: contributed by Darren Kenny

ldoms-control

The is the Oracle VM Server for SPARC (Logical Domains) control

domain.

Status: contributed by Darren Kenny

ldoms-guest

The is the Oracle VM Server for SPARC (Logical Domains) guest

domain.

Status: contributed by Darren Kenny

ldoms-io

The is the Oracle VM Server for SPARC (Logical Domains) I/O domain.

Status: contributed by Darren Kenny

ldoms-root

The is the Oracle VM Server for SPARC (Logical Domains) Root
domain.
Status: contributed by Darren Kenny

linux_vserver
This is printed for backwards compatibility with older virt-what
which could not distinguish between a Linux VServer container guest
and host.

linux_vserver-guest
This process is running in a Linux VServer container.
Status: contributed by BarXX Metin

linux_vserver-host
This process is running as the Linux VServer host (VxID 0).
Status: contributed by BarXX Metin and Elan Ruusamaee

lxc This process is running in a Linux LXC container.
Status: contributed by Marc Fournier

kvm This guest is running on the KVM hypervisor using hardware
acceleration.
Note that if the hypervisor is using software acceleration you
should not see this, but should see the "qemu" fact instead.
Status: confirmed by RWMJ.

lkvm
This guest is running on the KVM hypervisor using hardware
acceleration, and the userspace component of the hypervisor is lkvm
(a.k.a kvmtool).
Status: contributed by Andrew Jones

nutanix_ahv
The guest is running inside Nutanix Acropolis Hypervisor (AHV).
Status: confirmed by RWMJ.

oci The guest is running in an OCI container.
Status: contributed by Alessandro Valentini, confirmed by RWMJ

openvz
The guest appears to be running inside an OpenVZ or Virtuozzo
container.

Status: contributed by Evgeniy Sokolov

ovirt

The guest is running on an oVirt node. (See also "rhev" below).

Status: contributed by RWMJ, not confirmed

parallels

The guest is running inside Parallels Virtual Platform (Parallels Desktop, Parallels Server).

Status: contributed by Justin Clift

podman

This is a Podman container. (See also "oci" above.)

Status: contributed by Jordan Webb

powervm_lx86

The guest is running inside IBM PowerVM Lx86 Linux/x86 emulator.

Status: data originally supplied by Jeffrey Scheel, confirmed by Yufang Zhang and RWMJ

qemu

This is QEMU hypervisor using software emulation.

Note that for KVM (hardware accelerated) guests you should not see this.

Status: confirmed by RWMJ.

rhev

The guest is running on a Red Hat Enterprise Virtualization (RHEV) node.

Status: confirmed by RWMJ

redhat

The guest is running on the Red Hat hypervisor.

Status: confirmed by RWMJ

uml This is a User-Mode Linux (UML) guest.

Status: contributed by Laurent Leonard

virt

Some sort of virtualization appears to be present, but we are not sure what it is. In some very rare corner cases where we know that virtualization is hard to detect, we will try a timing attack to see if certain machine instructions are running much more slowly than they should be, which would indicate virtualization. In this case, the generic fact "virt" is printed.

virtage

This is Hitachi Virtualization Manager (HVM) Virtage hardware partitioning system.

Status: data supplied by Bhavna Sarathy, not confirmed

virtualbox

This is a VirtualBox guest.

Status: contributed by Laurent Leonard

virtualpc

The guest appears to be running on Microsoft VirtualPC.

Status: not confirmed

vmm This is a vmm (OpenBSD hypervisor) guest.

Status: contributed by Jasper Lievisse Adriaanse.

vmware

The guest appears to be running on VMware hypervisor.

Status: confirmed by RWMJ

xen The guest appears to be running on Xen hypervisor.

Status: confirmed by RWMJ

xen-dom0

This is the Xen dom0 (privileged domain).

Status: confirmed by RWMJ

xen-domU

This is a Xen domU (paravirtualized guest domain).

Status: confirmed by RWMJ

xen-hvm

This is a Xen guest fully virtualized (HVM).

Status: confirmed by RWMJ

EXIT STATUS

Programs that use or wrap "virt-what" should check that the exit status is 0 before they attempt to parse the output of the command.

A non-zero exit status indicates some error, for example, an unrecognized command line argument. If the exit status is non-zero then the output "facts" (if any were printed) cannot be guaranteed and should be ignored.

The exit status does not have anything to do with whether the program is running on baremetal or under virtualization, nor with whether "virt-what" managed detection "correctly" (which is basically unknowable given the large variety of virtualization systems out there and that some systems deliberately emulate others).

RUNNING VIRT-WHAT FROM OTHER PROGRAMS

"virt-what" is designed so that you can easily run it from other programs or wrap it up in a library.

Your program should check the exit status (see the section above).

Some programming languages (notably Python: issue 1652) erroneously mask the "SIGPIPE" signal and do not restore it when executing subprocesses. "virt-what" is a shell script and some shell commands do not work correctly when you do this. You may see warnings from "virt-what" similar to this:

```
 echo: write error: Broken pipe
```

The solution is to set the "SIGPIPE" signal handler back to "SIG_DFL" before running "virt-what".

IMPORTANT NOTE

Most of the time, using this program is the wrong thing to do. Instead you should detect the specific features you actually want to use. (As an example, if you wanted to issue Xen hypervisor commands you would look for the "/proc/xen/privcmd" file).

However people keep asking for this, so we provide it. There are a few legitimate uses:

Bug reporting tool

  If you think that virtualization could affect how your program runs, then you might use "virt-what" to report this in a bug reporting tool.

Status display and monitoring tools

  You might include this information in status and monitoring programs.

System tuning (sometimes)

You might use this program to tune an operating system so it runs better as a virtual machine of a particular hypervisor. However if installing paravirtualized drivers, it's better to check for the specific features your drivers need (eg. for the presence of PCI devices).

SEE ALSO

<http://people.redhat.com/~rjones/virt-what/>,
<http://www.vmware.com/>,
<http://www.microsoft.com/windows/products/winfamily/virtualpc>,
<http://xensource.com/>, <http://bellard.org/qemu/>,
<http://kvm.qumranet.com/>, <http://openvz.org/>

AUTHORS

Richard W.M. Jones <rjones @ redhat . com>

COPYRIGHT

REPORTING BUGS

Bugs can be viewed on the Red Hat Bugzilla page:
<https://bugzilla.redhat.com/>.

If you find a bug in virt-what, please follow these steps to report it:

1. Check for existing bug reports

   Go to <https://bugzilla.redhat.com/> and search for similar bugs.

   Someone may already have reported the same bug, and they may even

   have fixed it.

2. Capture debug and error messages

   Run

   ```
   virt-what > virt-what.log 2>&1
   ```

   and keep virt-what.log. It may contain error messages which you

   should submit with your bug report.

3. Get version of virt-what.

   Run

   ```
   virt-what --version
   ```

4. Submit a bug report.

   Go to <https://bugzilla.redhat.com/> and enter a new bug. Please

   describe the problem in as much detail as possible.

   Remember to include the version numbers (step 3) and the debug

   messages file (step 2) and as much other detail as possible.

5. Assign the bug to rjones @ redhat.com

   Assign or reassign the bug to rjones @ redhat.com (without the

   spaces). You can also send me an email with the bug number if you

   want a faster response.