NAME

bhyve_config - bhyve configuration variables

DESCRIPTION

bhyve(8) uses a hierarchical tree of configuration variables to describe global and per-device settings. Internal nodes in this tree do not have a value, only leaf nodes have values. This manual describes the configuration variables understood by bhyve(8). If additional variables are defined, bhyve(8) will ignore them and will not emit errors for unknown variables. However, these additional variables can be referenced by other variables as described below.

VARIABLE VALUES

Configuration variable values are stored as strings. A configuration variable value may refer to one or more other configuration values by name. Instances of the pattern '%(*var*)' are replaced by the value of the configuration variable *var*. To avoid unwanted expansion, '%' characters can be escaped by a leading '%'. For example, if a configuration variable *disk* uses the value */dev/zvol/bhyve/%(name)*, then the final value of the *disk* variable will be set to the path of a ZFS volume whose name matches the name of the virtual machine on the pool *bhyve*.

Some configuration variables may be interpreted as a boolean value. For those variables the following case-insensitive values may be used to indicate true:

- true
- on
- yes
- 1

The following values may be used to indicate false:

- false
- off
- no
- 0

Some configuration variables may be interperted as an integer. For those variables, any syntax supported by strtol(3) may be used.

GLOBAL SETTINGS

Architecture Neutral Settings

Name	Format	Default	Description
name	string		The name of the VM.

cpus	integer	1	The total number of virtual CPUs.
cores	integer	1	The number of virtual cores in each virtual socket.
threads	integer	1	The number of virtual CPUs in each virtual core.
sockets	integer	1	The number of virtual sockets.
memory.guest_in_core	bool	false	Include guest memory in core file.
memory.size	string	256M	Guest physical memory size in bytes. The value must be formatted as described in expand_number(3).
memory.wired	bool	false	Wire guest memory.
acpi_tables	bool	false	Generate ACPI tables.
acpi_tables_in_memory	bool	true	bhyve(8) always exposes ACPI tables by FwCfg. For backward compatibility bhyve copies them into the guest memory as well. This can cause problems if the guest uses the in-memory version, since certain advanced features, such as TPM emulation, are exposed only via FwCfg. Therefore, it is recommended to set this flag to false when running Windows guests.
destroy_on_poweroff	bool	false	Destroy the VM on guest-initiated power-off.
gdb.address	string	localhost	Hostname, IP address, or IPv6 address for the debug server.
gdb.port	integer	0	TCP port number for the debug server. If this is set to a non-zero value, a debug server will listen for connections on this port.
gdb.wait	bool	false	If the debug server is enabled, wait for a debugger to connect before starting the guest.
keyboard.layout	string		Specify the keyboard layout name with the file name in */usr/share/bhyve/kbdlayout*. This value only works when loaded with UEFI mode for VNC, and used a VNC client that don’t support QEMU Extended Key Event Message (e.g. TightVNC).
tpm.path	string		Path to the host TPM device. This is typically /dev/tpm0.
tpm.type	string		Type of the TPM device passed to the guest. Currently, only "passthru" is supported.
tpm.version	string	2.0	Version of the TPM device according to the TCG specification. Currently, only version 2.0 is supported.
rtc.use_localtime	bool	true	The real time clock uses the local time of the host. If this is set to false, the real time clock uses UTC.
uuid	string		The universally unique identifier (UUID) to use in the guest’s System Management BIOS System Information structure. If an explicit value is not set, a valid UUID is generated from the host’s hostname and the VM name.
virtio_msix	bool	true	Use MSI-X interrupts for PCI VirtIO devices. If set to

			false, MSI interrupts are used instead.
config.dump	bool	false	If this value is set to true after bhyve(8) has finished parsing command line options, then bhyve(8) will write all of its configuration variables to stdout and exit. No VM will be started.
bios.vendor	string	BHYVE	This value is used for the guest's System Management BIOS System Information structure.
bios.version	string	14.0	This value is used for the guest's System Management BIOS System Information structure.
bios.release_date	string	10/17/2021	This value is used for the guest's System Management BIOS System Information structure.
system.family_name	string	Virtual Machine	Family the computer belongs to. This value is used for the guest's System Management BIOS System Information structure.
system.manufacturer	string	FreeBSD	This value is used for the guest's System Management BIOS System Information structure.
system.product_name	string	BHYVE	This value is used for the guest's System Management BIOS System Information structure.
system.serial_number	string	None	This value is used for the guest's System Management BIOS System Information structure.
system.sku	string	None	Stock keeping unit of the computer. It's also called product ID or purchase order number. This value is used for the guest's System Management BIOS System Information structure.
system.version	string	1.0	This value is used for the guest's System Management BIOS System Information structure.
board.manufacturer	string	FreeBSD	This value is used for the guest's System Management BIOS System Information structure.
board.product_name	string	BHYVE	This value is used for the guest's System Management BIOS System Information structure.
board.version	string	1.0	This value is used for the guest's System Management BIOS System Information structure.
board.serial_number	string	None	This value is used for the guest's System Management BIOS System Information structure.
board.asset_tag	string	None	This value is used for the guest's System Management BIOS System Information structure.
board.location	string	None	Describes the board's location within the chassis. This value is used for the guest's System Management BIOS

			System Information structure.
chassis.manufacturer	string	FreeBSD	This value is used for the guest's System Management BIOS System Information structure.
chassis.version	string	1.0	This value is used for the guest's System Management BIOS System Information structure.
chassis.serial_number	string	None	This value is used for the guest's System Management BIOS System Information structure.
chassis.asset_tag	string	None	This value is used for the guest's System Management BIOS System Information structure.
chassis.sku	string	None	Stock keeping unit of the chassis. It's also called product ID or purchase order number. This value is used for the guest's System Management BIOS System Information structure.

x86-Specific Settings

Name	Format	Default	Description
x86.mptable	bool	true	Generate an MPTable.
x86.x2apic	bool	false	Configure guest's local APICs in x2APIC mode.
x86.strictio	bool	false	Exit if a guest accesses an I/O port that is not emulated. By default, writes are ignored and reads return all bits set.
x86.strictmsr	bool	true	Inject a general protection fault if a guest accesses a Model Specific Register (MSR) that is not emulated. If this is false, writes are ignored and reads return zero.
x86.vmexit_on_hlt	bool	false	Force a VM exit when a guest CPU executes the HLT instruction. This allows idle guest CPUs to yield the host CPU.
x86.vmexit_on_pause	bool	false	Force a VM exit when a guest CPU executes the PAUSE instruction.

DEVICE SETTINGS

Device settings are stored under a device node. The device node's name is set by the parent bus of the device.

PCI Device Settings

PCI devices are described by a device node named "pci.*bus*.*slot*.*function*" where each of *bus*, *slot*, and *function* are formatted as decimal values with no padding. All PCI device nodes must contain a configuration variable named "device" which specifies the device model to use. The following PCI device models are supported:

hostbridge

Provide a simple PCI-Host bridge device. This is usually configured at pci0:0:0 and is required by most guest operating systems.

ahci
AHCI storage controller.

e1000
Intel e82545 network interface.

fbuf
VGA framebuffer device attached to VNC server.

lpc LPC PCI-ISA bridge with COM1-COM4 16550 serial ports, a boot ROM, an optional fwcfg type, and an optional debug/test device. This device must be configured on bus 0.

hda High Definition audio controller.

nvme
NVM Express (NVMe) controller.

passthru
PCI pass-through device.

uart
PCI 16550 serial device.

virtio-9p
VirtIO 9p (VirtFS) interface.

virtio-blk
VirtIO block storage interface.

virtio-console
VirtIO console interface.

virtio-input
VirtIO input interface.

virtio-net
VirtIO network interface.

virtio-rnd
: VirtIO RNG interface.

virtio-scsi
: VirtIO SCSI interface.

xhci
: Extensible Host Controller Interface (XHCI) USB controller.

USB Device Settings

USB controller devices contain zero or more child USB devices attached to slots. Each USB device stores its settings in a node named "slot.*N*" under the controller's device node. *N* is the number of the slot to which the USB device is attached. Note that USB slot numbers begin at 1. All USB device nodes must contain a configuration variable named "device" which specifies the device model to use. The following USB device models are supported:

tablet
: A USB tablet device which provides precise cursor synchronization when using VNC.

Block Device Settings

Block devices use the following settings to configure their backing store. These settings are stored in the configuration node of the respective device.

Name	Format	Default	Description
path	string		The path of the file or disk device to use as the backing store.
nocache	bool	false	Disable caching on the backing file by opening the backing file with O_DIRECT.
nodelete	bool	false	Disable emulation of guest trim requests via DIOCGDELETE requests.
sync	bool	false	Write changes to the backing file with synchronous writes.
direct	bool	false	An alias for *sync*.
ro	bool	false	Disable writes to the backing file.
sectorsize	*logical*[/*physical*]		Specify the logical and physical sector size of the emulated disk. If the physical size is not specified, it is equal to the logical size.

Network Backend Settings

Network devices use the following settings to configure their backend. The backend is responsible for passing packets between the device model and a desired destination. Configuring a backend requires setting the *backend* variable. The type of a backend can either be set explicitly via the *type* variable or it

can be inferred from the value of *backend*.

The following types of backends are supported:

tap Use the tap(4) interface named in *backend* as the backend.

netgraph Use a netgraph(4) socket hook as the backend. This backend uses the following additional variables:

Name	Format	Default	Description
path	string		The name of the netgraph(4) destination node.
peerhook	string		The name of the destination hook.
socket	string		The name of the created ng_socket(4) node.
hook	string	vmlink	The name of the source hook on the created ng_socket(4) node.

netmap Use netmap(4) either on a network interface or a port on a vale(4) bridge as the backend. The value of *backend* is passed to nm_open to connect to a netmap port.

If *type* is not specified explicitly, then it is inferred from *backend* based on the following patterns:

Pattern	Type
tap*N*	tap
vmnet*N*	tap
netgraph	netgraph
netmap:*interface*	netmap
vale*bridge*:*port*	netmap

UART Device Settings

Name	Format	Default	Description
path	path		Backend device for the serial port. Either the pathname of a character device or "stdio" to use standard input and output of the bhyve(8) process.

Host Bridge Settings

Name	Format	Default	Description
*pcireg.**	integer		Values of PCI register.

Name	Default	
vendor	integer	0x1275
device	integer	0x1275

AHCI Controller Settings

AHCI controller devices contain zero or more ports each of which provides a storage device. Each port stores its settings in a node named "port.*N*" under the controller's device node. The *N* values are formatted as successive decimal values starting with 0. In addition to the block device settings described above, each port supports the following settings:

Name	Format	Default	Description
type	string		The type of storage device to emulate. Must be set to either "cd" or "hd".
nmrr	integer	0	Nominal Media Rotation Rate, also known as RPM. A value 1 of indicates a device with no rate such as a Solid State Disk.
ser	string	generated	Serial number of up to twenty characters. A default serial number is generated using a hash of the backing store's pathname.
rev	string	001	Revision number of up to eight characters.
model	string		Model number of up to forty characters. Separate default model strings are used for "cd" and "hd" device types.

e1000 Settings

In addition to the network backend settings, Intel e82545 network interfaces support the following variables:

Name	Format	Default	Description
mac	MAC address	generated	MAC address. If an explicit address is not provided, a MAC address is generated from a hash of the device's PCI address.

Frame Buffer Settings

Name	Format	Default	Description
wait	bool	false	Wait for a remote connection before starting the VM.
rfb	[*IP*:]*port*	127.0.0.1:5900	TCP address to listen on for remote connections. The IP address must be given as a numeric address. IPv6 addresses must be enclosed in square brackets and support scoped identifiers as described in getaddrinfo(3). A bare port number may be given in which case the IPv4 localhost address is used.
vga	string	io	VGA configuration. More details are provided in bhyve(8).
w	integer	1024	Frame buffer width in pixels.
h	integer	768	Frame buffer height in pixels.
password	string		Password to use for VNC authentication. This type of authentication is known to be cryptographically weak and is not intended for use on untrusted networks.

High Definition Audio Settings

Name	Format	Default	Description
play	path		Host playback device, typically */dev/dsp0*.
rec	path		Host recording device, typically */dev/dsp0*.

LPC Device Settings

The LPC bridge stores its configuration under a top-level *lpc* node rather than under the PCI LPC device's node. The following nodes are available under *lpc*:

Name	Format	Default	Description
bootrom	path		Path to a boot ROM. The contents of this file are copied into the guest's memory ending just before the 4GB physical address. If a boot ROM is present, a firmware interface device is also enabled for use by the boot ROM.
bootvars	path		Path to boot VARS. The contents of this file are copied beneath the boot ROM. Firmware can write to it to save variables. All variables will be persistent even on reboots of the guest.
com1	node		Settings for the COM1 serial port device.
com2	node		Settings for the COM2 serial port device.
com3	node		Settings for the COM3 serial port device.
com4	node		Settings for the COM4 serial port device.
fwcfg	string	bhyve	The fwcfg type to be used. Supported values are "bhyve" for fwctl and "qemu" for fwcfg.
pc-testdev	bool	false	Enable the PC debug/test device.
*pcireg.**	integer		Values of PCI register. It also accepts the value *host* to use the pci id of the host system. This value is required for the Intel GOP driver to work properly.

Name	Default
vendor	0x8086
device	0x7000
revid	0
subvendor	0
subdevice	0

NVMe Controller Settings

Each NVMe controller supports a single storage device. The device can be backed either by a memory disk described by the *ram* variable, or a block device using the block device settings described above. In addition, each controller supports the following settings:

Name	Format	Default	Description
maxq	integer	16	Maximum number of I/O submission and completion queue pairs.
qsz	integer	2058	Number of elements in each I/O queue.
ioslots	integer	8	Maximum number of concurrent I/O requests.
sectsz	integer		Sector size. Can be one of 512, 4096, or 8192. Devices backed by a memory disk use 4096 as the default. Devices backed by a block device use the block device's sector size as the default.
ser	string		Serial number of up to twenty characters. A default serial number is generated using a hash of the device's PCI address.
eui64	integer		IEEE Extended Unique Identifier. If an EUI is not provided, a default is generated using a checksum of the device's PCI address.
dsm	string	auto	Whether or not to advertise DataSet Management support. One of "auto", "enable", or "disable". The "auto" setting only advertises support if the backing store supports resource freeing, for example via TRIM.
ram	integer		If set, allocate a memory disk as the backing store. The value of this variable is the size of the memory disk in megabytes.

PCI Passthrough Settings

The ppt(4) device driver must be attached to the PCI device being passed through. The device to pass through can be identified either by name or its host PCI bus location.

Name	Format	Default	Description
bus	integer		Host PCI bus address of device to pass through.
slot	integer		Host PCI slot address of device to pass through.
func	integer		Host PCI function address of device to pass through.
pptdev	string		Name of a ppt(4) device to pass through.
rom	path		ROM file of the device which will be executed by OVMF to init the device.

VirtIO 9p Settings

Each VirtIO 9p device exposes a single filesystem from a host path.

Name	Format	Default	Description
sharename	string		The share name exposed to the guest.
path	path		The path of a directory on the host to export to the guest.
ro	bool	false	If true, the guest filesystem is read-only.

VirtIO Block Device Settings

In addition to the block device settings described above, each VirtIO block device supports the following settings:

Name	Format	Default	Description
ser	string	generated	Serial number of up to twenty characters. A default serial number is generated using a hash of the backing store's pathname.

VirtIO Console Device Settings

Each VirtIO Console device contains one or more console ports. Each port stores its settings in a node named "port.*N*" under the controller's device node. The *N* values are formatted as successive decimal values starting with 0. Each port supports the following settings:

Name	Format	Default	Description
name	string		The name of the port exposed to the guest.
path	path		The path of a UNIX domain socket providing the host connection for the port.

VirtIO Input Interface Settings

Each VirtIO Input device contains one input event device. All input events of the input event device are send to the guest by VirtIO Input interface. VirtIO Input Interfaces support the following variables:

Name	Format	Default	Description
path	path		The path of the input event device exposed to the guest

VirtIO Network Interface Settings

In addition to the network backend settings, VirtIO network interfaces support the following variables:

Name	Format	Default	Description
mac	MAC address	generated	MAC address. If an explicit address is not provided, a MAC address is generated from a hash of the device's PCI address.
mtu	integer	1500	The largest supported MTU advertised to the guest.

VirtIO SCSI Settings

Name	Format	Default	Description
dev	path		The path of a CAM target layer (CTL) device to export: */dev/cam/ctl*[*pp.vp*].
iid	integer	0	Initiator ID to use when sending requests to the CTL port.

SEE ALSO

expand_number(3), getaddrinfo(3), strtol(3), netgraph(4), netmap(4), ng_socket(4), tap(4), vale(4), vmnet(4), bhyve(8)