Virtuozzo

Virtuozzo Hybrid Server 7

Command Line Reference

September 7, 2023

Contents

CHAPTER 1

Introduction

Virtuozzo Hybrid Server 7 is a virtualization solution that allows you to run multiple virtual machines and containers on a single physical server.

This chapter provides general information about Virtuozzo Hybrid Server and this guide.

1.1 About Virtuozzo Hybrid Server

Virtuozzo Hybrid Server is a virtualization solution that allows you to simultaneously run multiple Virtuozzo virtual machines and containers on a single physical server. With Virtuozzo Hybrid Server, you can efficiently share your server's hardware resources among virtual machines and containers.

Virtuozzo Hybrid Server is installed directly on the server hardware and does not need any operating system to function. Once it is installed, Virtuozzo Hybrid Server allows you to create virtual machines and containers and manage them using the Virtuozzo Hybrid Server command-line interface (CLI). The command-line interface comprises a set of Virtuozzo Hybrid Server command-line utilities that you can use to manage virtual machines and containers, both locally and remotely.

1.2 About This Guide

This guide is a reference of Virtuozzo Hybrid Server configuration files and command-line utilities. It familiarizes you with the way to configure Virtuozzo Hybrid Server to meet your requirements and to perform various tasks by using the corresponding command-line utilities.

The primary audience for this guide is anyone who is looking for an explanation of a particular configuration option, needs help for a particular command, or is seeking for a command to perform a certain task.

CHAPTER 2

Managing Virtuozzo Hybrid Server

This chapter provides instructions on configuration files, scripts, and command-line utilities that can be used to configure the settings related to the Virtuozzo Hybrid Server software and the hardware node.

2.1 System Configuration Files

The table below lists the configuration files available in Virtuozzo Hybrid Server 7. Most files are located in the `/etc` directory on a hardware node. If a configuration file is stored in a place other than the hardware node, its exact location is specified.

Name	Description
`/etc/vz/vz.conf`	Global configuration file. This file keeps system-wide settings, such as the default location of templates and global network settings.
`/etc/vz/conf/<CT_name>.conf`	Private configuration file of a container with the name `<CT_name>`. This file keeps container-specific settings: resource management parameters, the location of its private area, IP address, and so on.
`/etc/vz/conf/ve-<name>. conf-sample`	Sample files containing a number of default container configurations. Some pre-created samples file are shipped with Virtuozzo Hybrid Server (e.g., `basic` and `confixx`), but you can also create your own samples to meet your demands.

Table 2.1.1 -- continued from previous page

Name	Description
`/usr/libexec/libvzctl/dists/` `<distribution_name>.conf`	Linux distribution configuration files. These files define what scripts should be run when you perform specific operations with containers (e.g., when you set a new IP address for a container). The scripts differ from Virtuozzo Hybrid Server action scripts and depend on the Linux version a particular container is running.
`/etc/vz/pfcache.conf`	Configuration file used by the `pfcache` utility to manage memory and IOPS deduplication.
`/etc/vz/conf/networks_classes`	Configuration file defining the network classes for traffic shaping and bandwidth management.
`/etc/sysctl.d/*`	Configuration files with kernel parameters required for VMs and containers to work.
`/etc/vztt/vztt.conf`	Configuration file used by the `vzpkg` utility to manage OS and application EZ templates.
`/etc/vz/tools-update.conf`	Configuration file used by the `vz-guest-tools-updater` script to manage automatic Virtuozzo Hybrid Server tools updating.
`/etc/shaman/shaman.conf`	Local configuration file used by the `shaman` and `shaman-monitor` utilities to manage and monitor high availability clusters.
`/vstorage/<cluster_name>/.drs/` `config`	Global configuration file used by the `pdrs` daemon to configure the DRS resource relocation mode for high availability clusters.

2.1.1 Global System Configuration File

Virtuozzo Hybrid Server keeps its system wide configuration parameters in the `/etc/vz/vz.conf` configuration file. This file is in shell format. Keep in mind that Virtuozzo Hybrid Server scripts source this file - thus, shell commands in this file will cause system to execute them under root account. Parameters in this file are presented in the form `PARAMETER="value"`. Logically all the parameters belong to the following groups: global parameters, logging, disk quotas, template, network traffic, and container default parameters. Below is the description of all the parameters defined in this version of Virtuozzo Hybrid Server.

Name	Description	Default Value
VIRTUOZZO	Can be `yes` or `no`. Virtuozzo Hybrid Server System V startup script checks this parameter. If set to `no`, then Virtuozzo Hybrid Server modules are not loaded. You might set it to "no" if you want to perform system maintenance and do not want to bring up all containers on the server.	yes
HTTP_PROXY	Specifies either the hostname or the IP address of the HTTP proxy server. After setting this parameter and in case you use an HTTP proxy server for handling all HTTP requests, the Virtuozzo Hybrid Server utilities communicating with the outer world through HTTP will use this server for managing all your HTTP messages.	n/a
ACTIONLOGDIR	This is the directory where `prlctl` keeps a log of its actions in the format suitable for Virtuozzo Hybrid Server statistics daemon `hwcoll`.	/vz/actionlog
LOCKDIR	Actions on a container should be serialized, since two simultaneous operations on the same container may break its consistency. Virtuozzo Hybrid Server keeps lock files in this directory in order to serialize access to one container.	/vz/lock
VEFSTYPE	File system to use when caching OS templates: • ext4, • simfs.	ext4
IPV6	Defines whether the IPv6 support is enabled on the hardware node.	yes
GOLDEN_IMAGE	Enables (`yes`) or disables (`no`) embedding application templates into OS EZ template cache prior to creating containers based on this cache.	yes
PFCACHE	Path to the memory and IOPS deduplication cache with common container files.	/vz/pfcache

Table 2.1.1.1 -- continued from previous page

Name	Description	Default Value
PFCACHE_IMAGE	Path to the private area of the memory and IOPS deduplication cache.	`/vz/pfcache.hdd`
PFCACHE_IMAGE_SIZE	Image size (in 1KB blocks) of the memory and IOPS deduplication cache.	`10485760`
PFCACHE_INCLUDES	Directories for which memory and IOPS deduplication is enabled by default.	`bin lib lib64 opt sbin usr`
VZ_TOOLS_BCID	Enables limits for the backup, restore, and migration operations.	
VZ_TOOLS_IOLIMIT	Sets the disk I/O limit for the backup and migration operations, in bytes per second. Not set by default.	
ON_VE_FSERROR	The action to be performed on a container if a file system error occurs. Currenly, the only available action is forced container stop. To enable, set the parameter to `stop`.	n/a
SERVER_UUID	(Virtuozzo Hybrid Server 7.0.6 and newer) Unique server identifier that is randomly generated and set on the `vz` service start. It is used to identify host ownership of containers on shared storage and provide access to their disks. **Note:** If you want to deploy multiple servers from a backup, make sure this parameter is not in the source server configuration file. Otherwise, restored servers will have the same `SERVER_UUID` and their containers host ownership identification will fail.	n/a
DEF_OSTEMPLATE	An OS template for newly created containers.	`.centos-7` for previous versions and `.almalinux-8` for new VHS versions.

Logging parameters:

Name	Description	Default Value
`LOGGING`	This parameter defines whether `prlctl` should log its actions.	`yes`
`LOGFILE`	File where `libvzctl` logs the actions of programs linked to this library.	`/var/log/vzctl.log`
`LOG_LEVEL`	Logging verbosity, from 0 to 10 (higher is more verbose).	`0`

Disk quota parameters:

Name	Description	Default Value
`DISK_QUOTA`	Enables or disables disk quotas for containers. If set to `no` then disk space accounting will be disabled.	`yes`

Network traffic parameters:

Name	Description	Default Value
`TRAFFIC_SHAPING`	Traffic shaping allows you to limit the bandwidth consumed by containers for outgoing traffic. If it is set to "yes", then limitations will be turned on. If you want to use this feature, `TRAFFIC_ACCOUNTING` should be set to `yes` as well.	`no`
`BANDWIDTH`	This is the list of network interfaces on which we want to shape the traffic and their speed in the form of "dev:rate". The rate is measured in Kbps. If you want to shape traffic on more than one interface, set this parameter to `dev1:rate1 dev2:rate2`. For example, for two 100 Mbps Ethernet cards, set it to `enp0s5:100000 enp0s6:100000`.	`enp0s5:100000`

Table 2.1.1.4 -- continued from previous page

Name	Description	Default Value
TOTALRATE	This parameter sets the size of the bandwidth pool for all containers. It is the upper limit for the bandwidth available to all your containers and is specified in the form of "dev:class:rate". The rate is measured in Kbps. Containers can consume bandwidth up to this limit in addition to the limit specified by the RATE parameter. Default value corresponds to 4 Mbps limit for the Class 1 containers.	*:1:4000
RATE	This parameter is the default bandwidth guaranteed to a container for outgoing traffic if the container configuration file does not explicitly specify a different value. This value is in the same format as TOTALRATE ("dev:class:rate"). The rate is measured in Kbps. Note that 8 Kbps, offered by the default configuration, is the guarantee and the container cannot consume less than this value and more than the sum of this value and TOTALRATE.	*:1:8
RATEMPU	This optional parameter (where MPU stands for "minimum packet unit") limits the packet rate by making packets smaller than MPU in size consume HTB tokens. With it, small packets can be accounted as larger ones and limited by TOTALRATE and RATE parameters. Approximately, the maximum packets per second rate can be calculated as TOTALRATE / RATEMPU. The parameter is specified in the form "dev:class:MPU per packet". If the part "MPU per packet" is omitted, the default value of 1000 bytes is used.	*:1:1000

Template parameters:

Name	Description	Default Value
TEMPLATE	This is the directory where to find templates. It is not recommended to redefine this option since all Virtuozzo Hybrid Server templates use the default directory.	`/vz/template`

Container default parameters:

Name	Description	Default Value
VE_ROOT	The mount point for container's `root`. Must contain the literal string `$VEID` that will be substituted with the actual container UUID.	`/vz/root/$VEID`
VE_PRIVATE	The directory where all the files and directories specific to the container are stored. Must contain the literal string `$VEID` that will be substituted with the actual container UUID.	`/vz/private/$VEID`
CONFIGFILE	The default configuration file sample to be used for the container creation; it may be overridden with the `--config` option of the `prlctl create` command.	`basic`
DEF_OSTEMPLATE	The default OS template to be used for the container creation.	`centos-7` for previous versions. `almalinux-8` for new VHS versions.
VE_ENVIRONMENT	Additional environment variables to be passed to the container `init` process. Should be provided as any number of `<name>=<value>` pairs separated by spaces.	

2.1.2 Container Configuration File

Each container has its own configuration file, which is stored in the `/etc/vz/conf` directory and has a name like `<CT_name>.conf`. This file has the same format as the global configuration file. The settings specified in this file can be subdivided into the following categories:

- miscellaneous,

- resource management parameters,

- networking.

2.1.2.1 Miscellaneous Parameters

The table below list the miscellaneous parameters you can set in the configuration file of a container:

Name	Description
VERSION	Specifies the Virtuozzo Hybrid Server version the configuration file applies to. `2` relates to Virtuozzo Hybrid Server version 4 and newer.
ONBOOT	Specifies whether the container should be started automatically on system startup. Virtuozzo Hybrid Server automatically starts all containers that have this parameter set to "yes" upon startup.
ALLOWREBOOT	Specifies whether the container may be restarted with the `reboot` command run from inside. If omitted or set to `yes`, restarting is allowed.
OSTEMPLATE	The name of the OS template that was used for creating the container. You do not have to change this parameter; `prlctl` will set it for you upon calling the `prlctl create` command (or using the defaults from the global configuration file). The . symbol before the OS template name, if specified, indicates that this is an EZ OS template.
TEMPLATES	In a configuration file of an existing container, this parameter lists application templates installed with the `prlctl create` or `vzpkg install` commands. In this case you should not modify it, because it is used by template management utilities to track installation history. This parameter is omitted if no templates have been installed to the container.
VE_ROOT	Overrides the `VE_ROOT` parameter from the global configuration file.
VE_PRIVATE	Overrides the `VE_PRIVATE` parameter from the global configuration file.
VE_ENVIRONMENT	Overrides the `VE_ENVIRONMENT` parameter from the global configuration file.

Table 2.1.2.1.1 -- continued from previous page

Name	Description
TECHNOLOGIES	Determines a set of technologies which should be provided by the Virtuozzo Hybrid Server kernel for container operation. Currently, this parameter can contain the information about the following technologies: • The system architecture of the container (`x86`, `x86_64`, or `i64`). • Whether the container is based on the OS template supporting the Native POSIX Thread Library (NPTL). In this case, the `nptl` entry is specified as the value of this parameter. • Whether the OS EZ template the container is based on requires the `sysfs` file system support (e.g., the OS EZ template for SUSE Linux Enterprise 10).
DISABLED	If set to `yes`, disables the container making it impossible to start the container once it was stopped. You can start the disabled container after setting the value of this parameter to `no`.
DESCRIPTION	Sets the description for the container. **Note:** You are allowed to use only symbols in the *A-z* and *0-9* ranges in your descriptions.
NAME	Container name that can be used to refer to said container in commands. Names must be alphanumeric and may contain the characters `\`, `-`, `_`. Names with white spaces must be enclosed in quotation marks.
ORIGIN_SAMPLE	The configuration sample the container was based on when created.
CONFIG_CUSTOMIZED	Indicates whether any of the container configuration parameters have been modified as regards its original configuration sample. If this parameter is omitted, its value is considered as `no`.
UUID	The container unique identifier. This identifier is used by certain Virtuozzo Hybrid Server utilities during their execution.

2.1.2.2 Resource Management Parameters

All resource management parameters can be subdivided into the CPU, disk, system, and VSwap categories for your convenience. Any parameter can be set with the `prlctl set` command and the corresponding option name (in the lower case, e.g., `--cpuunits` for `CPUUNITS`, etc.). See *Managing Containers* on page 41 for more details. The **Typical value** column, if present, specifies a range of reasonable parameter values for different applications, from light to huge heavy loaded containers. If the barrier and limit fields are in use, ranges for both thresholds are given.

CPU parameters:

Parameter	Description	Typical value
CPUUNITS	CPU weight. This is a positive integer number that defines how much CPU time the container can get as compared to the other virtual machines and containers running on the server. The larger the number, the more CPU time the container can receive. Possible values range from 8 to 500000. If this parameter is not set, the default value of 1000 is used.	250...1000
CPULIMIT, CPULIMIT_MHZ	CPU limit, in per cent (CPULIMIT) or megahertz (CPULIMIT_MHZ), the container is not allowed to exceed. The parameter is not set for newly created containers; so they can consume all free CPU power of the server. When setting this parameter in per cent, keep in mind that one CPU core makes up 100%. So if the server has 4 CPU cores, the total CPU power will equal 400%.	
CPUS	Number of CPU cores defining the CPU limit for a container. The limit is calculated by multiplying the power of one CPU core by the number of the specified CPU cores. This option also defines the number of CPUs shown to users from inside a container. This parameter is not set for newly created containers; so they can consume all free CPU power of the server.	
CPUMASK	The CPU affinity mask defining which CPUs on the Node can be used to handle the processes running in the container. The CPU mask can be specified as both separate CPU index numbers (1,2,3) and CPU ranges (2-4,5-7).	

Table 2.1.2.2.1 -- continued from previous page

Parameter	Description	Typical value
NODEMASK	The NUMA node mask defining a NUMA node to bind the container to. Once you set the mask, the processes running in the container will be executed only on the CPUs that belong to the specified NUMA node.	

Disk parameters:

Parameter	Description	Typical value
DISKSPACE	Total size of disk space that can be consumed by the container, in 1 KB blocks.	
QUOTAUGIDLIMIT	This parameter enables (if set to a value other than `0`) or disables (if set to `0`) per-user and per-group quotas for further management with the standard Linux `quota` utility. Enabling per-user and per-group quotas for a container requires restarting the container.	`0...N`
IOPRIO	The container priority for disk I/O operations. The higher the priority, the more time the container has for writing to and reading from the disk. The default container priority is 4.	`0-7`
IOPSLIMIT	The maximum number of disk input and output operations per second a container is allowed to perform. By default, any newly created container does not have the IOPS limit set and can perform so many disk I/O operations per second as necessary.	

Table 2.1.2.2.2 -- continued from previous page

Parameter	Description	Typical value
IOLIMIT	The bandwidth a container is allowed to use for its disk input and output (I/O) operations. By default, the limit is set in megabytes per second. However, you can use the following suffixes to use other measurement units: • G - sets the limit in gigabytes per second. • K - sets the limit in kilobytes per second. • B - sets the limit in bytes per second. In the current version of Virtuozzo Hybrid Server, the maximum I/O bandwidth limit you can set for a container is 2 GB per second. The default I/O bandwidth limit for all newly created containers is set to 0, which means that no limits are applied to any containers.	

System parameters:

Parameter	Description	Typical value
NUMPROC	Number of processes and threads allowed. Upon hitting this limit, container will not be able to start a new process or thread.	40...400
AVNUMPROC	Number of processes expected to run in the container on average. This is informational parameter used to ensure configuration correctness.	0...NUMPROC
VMGUARPAGES	Memory allocation guarantee, in pages. Applications are guaranteed to be able to allocate memory while the amount of memory accounted as privvmpages does not exceed the configured barrier of the vmguarpages parameter. Above the barrier, memory allocation is not guaranteed and may fail in case of overall memory shortage.	1725...107520
LOCKEDPAGES	Memory not allowed to be swapped out (locked with the mlock() system call), in pages (one page is 4 KB).	4...4096

Table 2.1.2.2.3 -- continued from previous page

Parameter	Description	Typical value
SHMPAGES	Total size of shared memory (including IPC, shared anonymous mappings and `tmpfs` objects), allocated by processes of a particular container, in pages.	512...16384
PRIVVMPAGES	Size of private (or potentially private) memory, allocated by an application. Memory that is always shared among different applications is not included in this resource parameter.	
NUMFILE	Number of files opened by all container processes.	512...8192
NUMFLOCK	Number of file locks created by all container processes.	50...200-60...220
NUMPTY	Number of pseudo-terminals. For example, the `ssh` session, `screen`, the `xterm` application consumes pseudo-terminal resources.	4...64
NUMSIGINFO	Number of `siginfo` structures (essentially this parameter limits the size of signal delivery queue).	256...512
PHYSPAGES	Total size of RAM used by processes. This parameter is used for accounting purposes only. It shows the usage of RAM by the container. For memory pages used by several different containers (mappings of shared libraries, for example), only a fraction of a page is charged to each container. The sum of the `physpages` for all containers corresponds to the total number of pages used in the system by all accounted users.	Not limited
NUMIPTENT	The number of IP packet filtering entries.	12...128

VSwap parameters:

Parameter	Description	Typical value
PHYSPAGES	Amount of RAM that can be used by the processes of a container, in 4KB pages.	
SWAP	Amount of swap space that can be used by the container for swapping out memory once the RAM is exceeded, in 4KB pages.	

Table 2.1.2.2.4 -- continued from previous page

Parameter	Description	Typical value
VM_OVERCOMMIT	Memory overcommit factor that defines the memory allocation limit for a container. The limit is calculated as `(PHYSPAGES + SWAP) * factor`.	`Not limited`

2.1.2.3 Networking Parameters

Network-related parameters allow you to set bandwidth management parameters, hostname and IP addresses that a container can use, and other parameters.

Name	Description
HOSTNAME	If this parameter is specified, then `prlctl` will set the hostname to its value upon the next container start. This parameter can be omitted. In this case, the container administrator should configure the hostname manually.
IP_ADDRESS	This is the list of IP addresses, which can be used on container network interfaces. This list is an argument of the container start call and it is impossible to assign IP address from inside the container if the address is not on the list. Any IP address assigned from within the container will be visible only within the container.
NAMESERVER	The IP address of the DNS server the container is supposed to use. More than one server can be specified in the space-separated format.
SEARCHDOMAIN	DNS search domains for the container. More than one domain can be specified.
NETDEV	The names of physical network adapters that have been moved from the server to the given container.
NETFILTER	Indicates which `iptables` modules are allowed for the container. If some of the allowed modules are not loaded on the destination Hardware Node after migration or restoration from backup, they will be automatically loaded on the migrated or restored container start. The following modes are available: • `disabled`: none. • `stateless`: (default) all modules except `conntrack` and NAT-related. • `stateful`: all modules except NAT-related. • `full`: all modules.

Table 2.1.2.3.1 -- continued from previous page

Name	Description
NETIF	Specifies a number of parameters for the virtual network adapters existing inside the container. These parameters include: • `ifname`: the name of the `veth` virtual Ethernet interface inside the container. • `mac`: the MAC address assigned to the `veth` virtual Ethernet interface inside the container. • `host_mac`: the MAC address assigned to the `veth` virtual Ethernet interface on the server. • `network`: the name of the virtual network where the `veth` virtual network adapter is included. • `ip`: the IP address(es) assigned to the `veth` virtual network adapter.
RATE	If traffic shaping is turned on, then this parameter specifies bandwidth guarantee, in Kbps, for the container. The parameter should be set in the form of `dev:class:rate`.
RATEBOUND	If set to `yes`, the bandwidth guarantee is also the limit for the container, and the container cannot borrow the bandwidth from the `TOTALRATE` bandwidth pool.

2.1.3 Linux Distribution Configuration Files

Some Virtuozzo Hybrid Server tools (e.g., `prlctl`) need to run special scripts inside a container to perform certain operations on it. However, carrying out one and the same operation inside containers running different Linux versions may require execution of different actions. This may be caused by the fact that different Linux distributions store files in different locations, use different commands to complete one and the same task, and so on. To distinguish between containers running different Linux versions and to determine what scripts should be executed while performing the relevant container-related operations, Virtuozzo Hybrid Server uses special distribution configuration files located in the `/usr/libexec/libvzctl/dists` directory on the server.

There are a number of distribution configuration files shipped with Virtuozzo Hybrid Server by default (`centos.conf`, `fedora-core.conf`, `gentoo.conf`, etc.). To view all configuration files available on your Virtuozzo Hybrid Server, you can go to the `/usr/libexec/libvzctl/dists` directory and issue the `ls` command. The distribution configuration files will be displayed in the form of `<Linux_distribution>-<version>.conf` where `<Linux_distribution>` and `<version>` denote the name of the Linux distribution and its version, respectively (e.g., `centos-7.conf`).

Any distribution configuration file consists of a number of entries in the form of `<parameter_name>=<script_name>` where `<parameter_name>` denotes the name of the parameter defining the operation when the script in the right part of the entry is to be executed and `<script_name>` is the name of the script to be run on performing the operation defined by the parameter in the left part of the entry. In the current version of Virtuozzo Hybrid Server, the following parameters are used to define what scripts should be executed for the corresponding Linux version a container is running:

- `ADD_IP`: the script specified as the value of this parameter has the default name of `<distribution_name>-add_ip.sh` and is used to configure the network settings during the container startup and the IP address(es) assignment. The script is launched inside the container on executing the following commands:

```
# prlctl start <CT_name>
# prlctl set <CT_name> --ipadd <IP_address>
# prlctl set <CT_name> --ipadd <IP_address> --ipdel all
```

- `DEL_IP`: the script specified as the value of this parameter has the default name of `<distribution_name>-del_ip.sh` and is used to delete an existing IP address from the container. The script is launched inside the container on executing the following commands:

```
# prlctl set <CT_name> --ipdel <IP_address>
# prlctl set <CT_name> --ipdel all
```

- `SET_HOSTNAME`: the script specified as the value of this parameter has the default name of `<distribution_name>-set_hostname.sh` and is used to configure the hostname of the container. The script is launched inside the container on executing the following command:

```
# prlctl set <CT_name> --hostname <name>
```

- `SET_DNS`: the script specified as the value of this parameter has the default name of `<distribution_name>-set_dns.sh` and is used to configure DNS parameters in the `/etc/resolv.conf` file. The script is launched inside the container on executing the following command:

```
# prlctl set <CT_name> --searchdomain <domain> --nameserver <IP_address>
```

- `SET_USERPASS`: the script specified as the value of this parameter has the default name of `<distribution_name>-set_userpass.sh` and is used to add a new user or change the current password. The script is launched inside the container on executing the following command:

```
# prlctl set <CT_name> --userpasswd <user>:<passwd>
```

- `SET_UGID_QUOTA`: the script specified as the value of this parameter has the default name of `<distribution_name>-set_ugid_quota.sh` and is used to set up per-user/group quota. The script is

launched inside the container on executing the following command:

```
# prlctl set <CT_name> --quotaugidlimit <num>
```

- `POST_CREATE`: the script specified as the value of this parameter has the default name of `<distribution_name>-postcreate.sh` and is used to perform certain tasks (e.g., to modify the `crontab` files) after the container creation. This script is launched on the server on executing the following command:

```
# prlctl create <CT_name>
```

The scripts specified in distribution configuration files are located in the `/usr/libexec/libvzctl/dists/scripts` directory on the server and executed on performing the aforementioned operations on the containers. After an operation has been initiated, the `prlctl` utility turns to the corresponding container configuration file, looks for the value of the `DISTRIBUTION` variable or, if the latter is not present, of the `OSTEMPLATE` variable in this file, and defines on their basis what Linux version the given container is running. After that, `prlctl` reads the corresponding configuration file for the determined Linux version from the `/usr/libexec/libvzctl/dists/` directory and executes the scripts specified in this file.

Note: If no distribution is specified as the value of the `DISTRIBUTION` and `OSTEMPLATE` variables in the container configuration file or no configuration file for the given Linux version was found in the `/usr/libexec/libvzctl/dists` directory, the `default` file from this directory is used.

2.1.4 Memory and IOPS Deduplication Configuration File

Contained in the `/etc/vz/pfcache.conf` file, memory and IOPS deduplication parameters allow you to tailor cache behavior and performance to your needs.

Name	Description	Default Value
COUNT	The minimum number of file copies required for the file to become cacheable. Copies may exist in the same container or different containers.	2
MINSIZE	Minimal cacheable file size, bytes. Files smaller than this value will not be cached.	0

Table 2.1.4.1 -- continued from previous page

Name	Description	Default Value
MAXSIZE	Maximal cacheable file size, bytes. Files larger than this value will not be cached.	2147483648
TIMEOUT	Time between caching attempts, seconds.	5
PFCACHE_IOLIMIT	Memory and IOPS deduplication cache IO bandwidth limit, bps. Unlimited by default.	
PFCACHE_IOPSLIMIT	Memory and IOPS deduplication cache IOPS limit. Unlimited by default.	
LOGLEVEL	Logging verbosity. Messages are logged in the system log file `/var/log/messages`.	1
PAGEMIN	The total number of memory pages used in containers: • `0` - Cache even files without memory pages. • `1` - Cache only files in use. • `N` - Cache only when the total number of memory pages in containers reaches N.	1
PURGEAHEAD	Extra cache space to free up in addition to the requested space. In per cent of the requested space. Used with the `pfcache purge --size` command.	20%

2.1.5 Network Classes Definition File

In Virtuozzo Hybrid Server, both traffic accounting and bandwidth management are based on network classes. The network classes' definition file (`/etc/vz/conf/networks_classes`) describes network classes that Virtuozzo Hybrid Server recognizes. Currently, there can be up to 15 classes defined.

The lines in this file have the following format:

```
<class_ID> <IP_address>/<prefix_length>
```

where `<class_ID>` defines the network class identifier, `<IP_address>` defines the starting IP address, and `<prefix_length>` defines the subnet mask. In pair `<IP_address>` and `<prefix_length>` define the range of IP addresses for this class. There may be several lines for each class. Classes should be defined after Class 1 and represent exceptions from the "matching-everything" rule of Class 1. Class 0 has a special meaning and defines the IP ranges for which no accounting is done (this server container addresses).

The definition of class 1 is required; any class except class 1 can be omitted. However, it is recommended to define class 0 correctly as it will improve performance. For example:

```
# HW node VPS's networks
0 10.10.10.0/24
0 10.10.15.0/24
# all IP("local" traffic)
1 0.0.0.0/0
# class 2 - "foreign" traffic
#2 10.0.0.0/8
#2 11.0.0.0/8
# inside "foreign" network there
# is a hole with "local" traffic
#1 10.10.16.0/24
```

2.1.6 Offline Management Configuration Files

The offline management configuration files located in the `/etc/vzredirect.d` directory define various modes of container offline management by container administrators. One configuration file describes one offline management mode. In the current Virtuozzo Hybrid Server version, two files are accessible: `vzpp.conf` and `vzpp-plesk.conf`. The first file defines the container offline management by means of Power Panel, and the second one by means of the same Power Panel with an integrated Plesk control panel.

There are two parameters in each of the files.

```
<parameter_name>=<parameter_value>
```

Name	Description	Example
PORT	This port must be entered in the address line of an Internet browser after the container IP address when managing the container by means of Power Panel or the Plesk control panel.	PORT=8443
DST_VEID	The UUID of the container where the requests coming to the specified port will be redirected.	DST_VEID=1

2.1.7 vztt Configuration File

This file (`/etc/vztt/vztt.conf`) is the configuration file used by the `vzpkg` utility when managing OS and application EZ templates.

```
<parameter_name>=<parameter_value>
```

Name	Description
VZTT_PROXY	The IP address or hostname of the caching proxy server to be used by the vzpkg tool for managing OS and application EZ templates.
HTTP_PROXY	The IP address or hostname of the HTPP proxy server address, if you use this server.
HTTP_PROXY_USER	The user name used by the HTTP proxy server for your authentication.
HTTP_PROXY_PASSWORD	The password of the user specified in the HTPP_PROXY_USER parameter and used for your authentication by the HTTP proxy server.
METADATA_EXPIRE	Defines the period of time, in seconds, in the course of which the downloaded software packages in the vzpkg cache are regarded as *not obsolete*. During this time, the vzpkg utility searches for the EZ template packages in the local cache only (without checking the remote repositories set for EZ templates). By default, this period is set to 86400 seconds (24 hours).
EXCLUDE	List of comma-separated packages that are not to be installed or updated during the vzpkg execution. The package names should correspond to the name of real packages in the repository and can contain file globs (e.g., * and ?).

2.1.8 pcompact.conf

The /etc/vz/pcompact.conf file is used by the pcompact utility to compact virtual disks in containers.

```
<parameter_name>=<parameter_value>
```

Name	Description
THRESHOLD=<number>	Compact the virtual disk if unused space on it exceeds **THRESHOLD** percent of the ploop size.
DELTA=<number>	Reduce disk space to be compacted by **DELTA** percent of the ploop size.
DEFRAG=<yes|no>	Perform or skip file system defragmentation.

2.1.9 tools-update.conf

The file /etc/vz/tools-update.conf is used by the vz-guest-tools-updater script to manage automatic Virtuozzo Hybrid Server tools updating.

```
"<parameter_name>": <parameter_value>
```

Name	Description
`"MaxVMs": <number>`	Sets the maximum number of virtual machines whose Virtuozzo Hybrid Server tools can be updated simultaneously.
`"InstallTools": true|false`	Enables or disables automatic installation of Virtuozzo guest tools in virtual machines.

2.1.10 shaman Configuration File

`/etc/shaman/shaman.conf` is the local configuration file for the `shaman-monitor` daemon and the `shaman` command-line tool. It affects settings of `shaman-monitor` running on the particular node. At the same time, a number of cluster-wide configuration parameters used by the `shaman-monitor` daemon are defined in the global configuration file `shaman-config`. You can obtain these parameters by running the `shaman get-config` command and modify them with `shaman set-config`.

```
<parameter_name>=<parameter_value>
```

Local parameters:

Name	Description
`CLUSTER_NAME`	Set the name of the cluster for `shaman` to operate on.
`LOG_LEVEL`	Set the verbosity level for printed messages: • `0`: print error messages only; • `1`: print warning messages in addition to error messages; • `2` (default): print information messages in addition to errors and warnings; • `4`: print debug diagnostic messages in addition to messages from levels 0-2.

Global parameters:

Name	Description
LOCK_TIMEOUT	Set the timeout for `shaman-monitor` operations (e.g., electing a new master or deciding that a slave node is down). This parameter helps avoid situations when `shaman-monitor` performs a cluster-related operation if someone pulls out the network cable for just a couple of seconds (not longer than the half of the `LOCK_TIMEOUT` value). The default value is 60 seconds. The specified value is added to the value of other timeout-related parameters (see below).
LEASE_CHECK_TIMEOUT_FOR_MASTER	Set the timeout for electing a new master node when the original master node or the `shaman-monitor` daemon fails, or high availability support gets disabled. The default value is 10 seconds.
LEASE_CHECK_TIMEOUT_FOR_SLAVE	Set the timeout after which the master node will consider a slave node as broken if this node and the `shaman-monitor` daemon get down, or high availability support gets disabled. The default value is 10 seconds.
LEASE_LOST_ACTION	Defines the action to perform if `shaman-monitor` loses connection to cluster. This may happen when a node goes online after having been disconnected from network for more than `LOCK_TIMEOUT` seconds. In this case, the watchdog timer has not expired yet, but the cluster is already unavailable, because the master node has prohibited access to the cluster until the node is rebooted. Available values are `crash`, `halt`, `reboot`, and `none` (do nothing).
CLUSTER_MOUNTPOINT_DEAD_ACTION	Defines the action to perform when `shaman-monitor` detects that the cluster mount point is no longer functioning properly for some reason. The supported actions are `crash`, `halt`, `reboot`, `none`.

Table 2.1.10.2 -- continued from previous page

Name	Description
RELOCATION_SKIP_THRESHOLD	Sets the threshold for the number of simultaneously crashed nodes.
	If the number of simultaneously crashed nodes becomes greater than or equal to the threshold, the master stops relocating resources from the crashed nodes. When the number of simultaneously crashed nodes drops below the threshold, the master automatically resumes relocating resources from the crashed nodes. The threshold can be useful when multiple nodes are being rebooted at the same time. Without it, the master would start relocating resources from all the rebooting nodes.
	The threshold is set to 3 by default and must be 2 or greater. For clusters with only 3 nodes, the threshold is automatically set to 2.
POOL_CHECK_TIMEOUT	Set the interval for `shaman-monitor` to check for the resources scheduled for relocation.
	The default value is 30 seconds.

Continued on next page

Table 2.1.10.2 -- continued from previous page

Name	Description
`RESOURCE_RELOCATION_MODE`	Defines a sequence of algorithms (modes) used for resource relocation on hardware node failure. At least one mode must be specified. Multiple modes must be separated with commas. On hardware node failure, relocation using the first specified mode is attempted. If unsuccessful, the next specified mode is attempted and so on. If relocation using the last specified mode is unsuccessful, the resources are left on the failed hardware node. The following resource relocation modes are supported: • `round-robin`: Each resource from the failed hardware node is relocated to another node, which is chosen using the round-robin algorithm. In general, resources are relocated to different hardware nodes. • `spare`: All resources from the failed hardware node are relocated to a 'spare' node. A spare node is a hardware node, which is registered in the cluster and has no resources stored on it. • `drs`: All resources from the failed hardware node are relocated using an external DRS daemon. The default sequence is `drs, round-robin`.
`WATCHDOG_TIMEOUT`	Set the interval for the watchdog timer. The watchdog timer is responsible for performing the action defined in `WATCHDOG_ACTION` (see below) if `shaman-monitor` crashes or hangs up. The `shaman-monitor` daemon activates the watchdog timer on its start-up and periodically resets it to the specified value. If something goes wrong with `shaman-monitor` so that it fails to reset the timer, the watchdog timer counts down until it reaches zero and performs the defined action. Setting the interval to zero disables the watchdog timer. Minimal watchdog timer interval that could be set is 10 seconds. The default value is 120 seconds.

Continued on next page

Table 2.1.10.2 -- continued from previous page

Name	Description
WATCHDOG_ACTION	Defines a sequence of actions to perform after the watchdog timer expires (happens when `shaman-monitor` crashes or hangs up). When the watchdog timer expires, the first specified action is attempted. If unsuccessful, the next specified action is attempted and so on. If the last specified action is unsuccessful, then the action specified in the `/sys/kernel/watchdog_action` file is performed. At least one action must be specified. Multiple actions must be separated with commas. Available actions are listed in the `/sys/kernel/watchdog_available_actions` file. The default sequence is `netfilter, reboot`.

2.1.11 pdrs Configuration File

The file `/vstorage/<cluster_name>/.drs/config` is used by the `pdrs` daemon to configure the DRS resource relocation mode for high availability clusters.

```
<parameter_name>=<parameter_value>
```

Name	Description
SNMP_PORT	Specifies a UDP listen port for SNMP traps. The default port is 33333.
SNMP_TRAP_PERIOD	Sets a period between SNMP traps. The default value is 60.
SNMP_TRAP_NUMBER	Sets the number of SNMP traps. The default value is 20.
DISP_CONN_TIMEOUT	Sets a dispatcher connection timeout. The default value is 600.
MASTER_LEASE_TIMEOUT	Sets a master node connection timeout, after which said node will lose its master status in the cluster. The default value is 60.

Table 2.1.11.1 -- continued from previous page

Name	Description
MEMPOL_CT_RAM_OVERCOMMIT	Sets an overcommit coefficient used to find a suitable node according to its available RAM capacity while relocating containers from failed nodes to healthy ones. The amount of RAM required for a container on a healthy node can be calculated by dividing the container RAM size by its overcommit coefficient. The value is specified as a decimal fraction. The default value is 2.0.
MEMPOL_VM_RAM_OVERCOMMIT	Sets an overcommit coefficient used to find a suitable node according to its available RAM capacity while relocating virtual machines from failed nodes to healthy ones. The amount of RAM required for a virtual machine on a healthy node can be calculated by dividing the VM RAM size by its overcommit coefficient. The value is specified as a decimal fraction. The default value is 1.0.

2.2 System Utilities

This section provides information on utilities that can be used to manage Virtuozzo Hybrid Server parameters.

2.2.1 prlsrvctl

The prlsrvctl command-line utility is used to perform management tasks on the hardware node and Virtuozzo Hybrid Server. The tasks include getting the Virtuozzo Hybrid Server information, modifying its preferences, installing a license, obtaining statistics and problem reports, and some others.

```
prlsrvctl [<command> [<options>]
          [-l, --login [<user>[:<passwd>]@]<server>[:<port>]]
```

Name	Description
`<command>`	The command to execute.
`<options>`	Command options. See individual commands for available options.
`-l, --login`	Connect to the remote hardware node and execute a command there. If this parameter is omitted, the command will be executed on the local server.
`<user>`	The name of the user used to log in to the remote server.
`<passwd>`	The user password. If the password is omitted, you will be prompted to enter it.
`<server>:<port>`	The remote server IP address or hostname and port number, If port number is omitted, the default port will be used.

Note: To display help, enter `prlsrvctl` on the command-line without any options.

2.2.1.1 prlsrvctl backup

The command is used to back up all virtual environments on the node.

```
prlsrvctl backup [-f,--full] [-i,--incremental]
                 [-s,--storage <user>[:<password>]@<server>[:<port>]]
                 [--description <desc>] [-u,--uncompressed]
```

Name	Description
`-f,--full`	Creates a full backup of each virtual environment on the node. A full backup contains all the virtual environment data.
`-i,--incremental`	Creates an incremental backup of each virtual environment on the node. An incremental backup contains only the files that were changed since the previous full or incremental backup. This is the default backup type.
`-s,--storage` `<user>[:<password>]@<server>[:<port>]`	The host to store backup images at.
`--description <desc>`	Adds a description to each virtual environment backup.
`-u,--uncompressed`	Does not compress backup images.

28

2.2.1.2 prlsrvctl info

Displays the hardware node and Virtuozzo Hybrid Server configuration information.

```
prlsrvctl info
```

The information returned by the `info` command includes the following:

- Server ID and hostname.

- Virtuozzo Hybrid Server version number.

- Default directory for storing virtual machine files.

- Virtuozzo Hybrid Server memory limits.

- Virtuozzo Hybrid Server minimum allowable security level.

- Default directory for storing virtual machine backups.

- Virtuozzo Hybrid Server license information.

- Server hardware configuration information.

- Other miscellaneous info.

2.2.1.3 prlsrvctl net

The `prlsrvctl net` command is used to create and configure virtual networks.

Subcommands

Name	Description
net add	Creates a new virtual network
net set	Configures the parameters of an existing virtual network.
net del	Removes an existing virtual network.
net list	List the available virtual networks.

net add

Creates a new virtual network.

```
prlsrvctl net add <vnetwork_ID> [-i, --ifname <if>] [-m, --mac <mac_address>]
                  [-t, --type <bridged|host-only>] [-d, --description <desc>]
```

```
               [--ip <IP_address>[/<mask>]] [--dhcp-server <on|off>]
               [--dhcp-ip <IP_address>] [--ip-scope-start <IP_address>]
               [--ip-scope-end <IP_address>] [--ip6 <IP_address>[/</emphasis><mask>_]]
               [--dhcp6-server <on|off>] [--dhcp-ip6 <IP_address>]
               [--ip6-scope-start <IP_address>] [--ip6-scope-end <IP_address>]
```

Name	Description
`<vnetwork_ID>`	A user-defined name that will identify the new virtual network.
`-i, --ifname <if>`	The name of a physical network adapter on the hardware node to which this virtual network should be bound.
`-m, --mac <mac_address>`	The MAC address of a virtual network adapter on the hardware node to which this virtual network should be bound.
`-t, --type <bridged|host-only>`	The type of the virtual network to create. Possible values are: • `bridged`. A virtual machine and container connected to this type of virtual network appears as an independent computer on the network. • `host_only` (default). A virtual machine and container connected to this type of virtual network can access only the hardware node and the virtual machines and containers connected to the same virtual network.
`-d, --description <desc>`	A user-defined description of the virtual network. Descriptions with white spaces must be enclosed in quotation marks.
`--ip <IP_address>[/<mask>]` `--ip6 <IP_address>[/<mask>]`	Set an IPv4/IPv6 address and subnet mask for the Virtuozzo Hybrid Server virtual adapter.
`--dhcp-server <on|off>` `--dhcp6-server <on|off>`	Enable or disable the Virtuozzo Hybrid Server virtual DHCPv4/DHCPv6 server.
`--dhcp-ip <IP_address>` `--dhcp-ip6 <IP_address>`	Set an IPv4/IPv6 address for the Virtuozzo Hybrid Server virtual DHCPv4/DHCPv6 server.

`--ip-scope-start <IP_address>`
`--ip-scope-end <IP_address>`
`--ip6-scope-start <IP_address>`
`--ip6-scope-end <IP_address>`
	Set the starting and ending IPv4/IPv6 addresses for the DHCPv4/DHCPv6 pool. The virtual machines and containers connected to the network you are creating will automatically receive their IPv4/IPv6 addresses from the respective DHCPv4/DHCPv6 pool.

net set

Configures the settings of an existing virtual network.

```
prlsrvctl net set <vnetwork_ID> [-i, --ifname <if>] [-m, --mac <mac_address>]
                  [-t, --type <bridged|host-only>] [-d, --description <desc>]
                  [--ip <IP_address>[/<mask>]] [--dhcp-server <on|off>]
                  [--dhcp-ip <IP_address>] [--ip-scope-start <IP_address>]
                  [--ip-scope-end <IP_address>] [--ip6 <IP_address>[/</emphasis><mask>_]]
                  [--dhcp6-server <on|off>] [--dhcp-ip6 <IP_address>]
                  [--ip6-scope-start <IP_address>] [--ip6-scope-end <IP_address>]
```

Name	Description
`<vnetwork_ID>`	The name of the virtual network to modify.
`-i, --ifname <if>`	The name of a physical network adapter on the hardware node to which this virtual network should be bound.
`-m, --mac <mac_address>`	The MAC address of a virtual network adapter on the hardware node to which this virtual network should be bound.
`-t, --type <bridged|host-only>`	The type of the virtual network to modify. Possible values are: • `bridged`. A virtual machine and container connected to this type of virtual network appears as an independent computer on the network. • `host_only` (default). A virtual machine and container connected to this type of virtual network can access only the hardware node and the virtual machines and containers connected to the same virtual network.
`-d, --description <desc>`	A user-defined description of the virtual network. Descriptions with white spaces must be enclosed in quotation marks.
`--ip <IP_address>[/<mask>]` `--ip6 <IP_address>[/<mask>]`	Set an IPv4/IPv6 address and subnet mask for the Virtuozzo Hybrid Server virtual adapter.
`--dhcp-server <on|off>` `--dhcp6-server <on|off>`	Enable or disable the Virtuozzo Hybrid Server virtual DHCPv4/DHCPv6 server.
`--dhcp-ip <IP_address>` `--dhcp-ip6 <IP_address>`	Set an IPv4/IPv6 address for the Virtuozzo Hybrid Server virtual DHCPv4/DHCPv6 server.

`--ip-scope-start <IP_address>`
`--ip-scope-end <IP_address>`
`--ip6-scope-start <IP_address>`
`--ip6-scope-end <IP_address>`
	Set the starting and ending IPv4/IPv6 addresses for the DHCPv4/DHCPv6 pool. The virtual machines and containers connected to the network you are creating will automatically receive their IPv4/IPv6 addresses from the respective DHCPv4/DHCPv6 pool.

net del

Deletes an existing virtual network.

```
prlsrvctl net del <vnetwork_ID>
```

Name	Description
<vnetwork_ID>	The name of the virtual network to delete.

net list

Lists the existing virtual networks.

```
prlsrvctl net list
```

2.2.1.4 prlsrvctl problem-report

Generates and displays problem reports.

```
prlsrvctl problem-report
```

The command collects technical data about Virtuozzo Hybrid Server and the hardware node and displays the report on screen (the output can also be piped to a file). The report can then be directed to the Virtuozzo technical support team for analysis.

2.2.1.5 prlsrvctl set

Configures Virtuozzo Hybrid Server preferences.

```
prlsrvctl set [-s, --min-security-level <low|normal|high>]
              [--mng-settings <allow|deny>]
              [--device <device> --assignment <host|vm>_]
              [--backup-storage [<user>[:<passwd>]@]<server>[:<port>]]
              [--backup-tmpdir <path>] [--backup-path <path>]
              [--backup-mode <push|push-with-reversed-delta>]
              [--idle-connection-timeout <timeout>]
              [--verbose-log <on|off>] [--cluster-mode <on|off>]
              [--cpu-features-mask <{+|-}feature1,feature2=value[,...]>]
              [--vm-cpulimit-type <full|guest>]
              [--vcmmd-policy <density|performance>]
              [--vnc-ssl-certificate <path> --vnc-ssl-key <path>] [--vnc-clipboard <on|off>]
              [--vnc-default-address <ip>]
```

Name	Description
`-s, --min-security-level <low|normal|high>`	The lowest allowable security level that can be used to connect to the hardware node. The following options are available: • `low`, plain TCP/IP (no encryption). • `normal`, **most important data is sent and received using SSL** over TCP/IP (user credentials during login, guest OS clipboard, etc.) Other data is sent and received using plain TCP/IP with no encryption. • `high`, all of the data is sent and received using SSL.
`--mng-settings <allow|deny>`	Grants or denies permission to new users to modify Virtuozzo Hybrid Server preferences. By default, only administrators of the host OS can modify Virtuozzo Hybrid Server preferences. When a new user profile is created (this happens when a user logs in to the hardware node for the first time), he/she will be granted or denied this privilege based on the default setting. This parameter allows you to set that default setting. Please note that this parameter only affects new users (the users that will be created in the future). The profiles of the existing users will not be modified.
`--device <device> --assignment <host|vm>`	Allows to set the assignment mode for the specified VTd device. The following options are available: • `host`, assign the device to the hardware node. • `vm`, assign the device to virtual machines.
`--backup-storage <server>[:<port>]`	The default backup server where to store virtual machine backups. Requires RSA authentication to be set up. See Setting Up RSA Authentication Between Nodes.
`--backup-path <path>`	The name and path of the default directory on the backup server where to store virtual machine backups.
`--verbose-log <on|off>`	Turns the verbose output for the command on or off.
`--cluster-mode <on|off>`	Turns the cluster mode on or off.

Table 2.2.1.5.1 -- continued from previous page

Name	Description
`--idle-connection-timeout <timeout>`	Sets a timeout interval in seconds after which, if no data has been received from the storage server or backup client, the process of backup/restore is terminated.
`--backup-tmpdir <path>`	Sets the directory for temporary snapshots created during backup. If this directory is set without setting the backup mode as well, and if the value is not empty, the backup mode automatically switches to `push-with-reversed-delta` with a corresponding warning.
`--backup-mode <push|push-with-reversed-delta>`	Sets the backup mode. If both the `backup-mode` and `backup-tmpdir` options are set, the latter has higher priority. If only `backup-tmpdir` is set, the backup mode switches to `push-with-reversed-delta`.
`--cpu-features-mask <{+|-}feature1,feature2=value[,...]>`	Changes CPU features mask on the host. To mask/unmask features, use the `+feature/-feature` syntax respectively. Omitting the sign is equvalent to unmasking. Features that require specific value can be set using the `feature=value` syntax. To view a full list of host CPU features which are supported, unmaskable and already masked, run the `prlsrvctl info --full` command. Note the following: • Only Intel Ivy Bridge and newer CPUs are supported. AMD processors are not supported. • All virtual machines and containers on the host must be stopped. • You can change CPU features mask only for physical servers.

Continued on next page

Table 2.2.1.5.1 -- continued from previous page

Name	Description
`--vm-cpulimit-type <full|guest>`	Specifies the type of virtual machine threads to be affected by the CPU limit: • `full` **(default) - both hardware emulation and guest OS threads** are limited. • `guest` - only guest OS threads are limited. With the `guest` option, the guest OS is guaranteed to have all the resources implied by the VM configuration. At the same time, the VM's hardware emulation threads spend additional resources of the host. For example, for a VM with two 2.8 GHz vCPUs, switching to guest means that VM's guest applications will have all the resources of two 2.8 GHz vCPUs at their disposal. Note the following: • Some types of guest applications, like voice-over-IP software, significantly increase expenses on hardware emulation threads. • After changing this parameter, restart running virtual machines for the changes to take effect.
`--vcmmd-policy <density|performance>`	Switches the automatic memory management policy on the host: • `performance` **(default), used for nodes without memory** overcommit. • `density`, recommended for nodes with memory overcommit. **Note:** Before setting a policy, make sure there are no running virtual machines or containers on the host.

Continued on next page

Table 2.2.1.5.1 -- continued from previous page

Name	Description
`--vnc-ssl-certificate <path>` `--vnc-ssl-key <path>`	Names and paths of SSL certificate file and key used to encrypt VNC connections on the node. To disable VNC encryption, specify empty arguments (e.g., ").
`--vnc-clipboard <on|off>`	You can enable or disable the option of copying to the clipboard.
`--vnc-default-address <ip>`	Specify a default VNC address for all created VMs after setting the option. If used, `vnc-default-address` shows up in the VE configuration as an additional VNC address marked as a global address.

2.2.1.6 prlsrvctl usb

The `prlsrvctl usb` command is used to permanently assign a USB device to a specific virtual machine. This functionality works only with virtual machines (not containers).

Subcommands

Name	Description
`usb list`	Lists USB devices connected to the server together with the information about their virtual machine assignments for the current user.
`usb set`	Permanently assigns a USB device to the specified virtual machine.
`usb del`	Removes a previously created USB device assignment.

usb list

Lists the USB devices connected to the physical server.

```
prlsrvctl usb list
```

Returns a list of USB devices in tabular format with the following columns:

- **Name** - the USB device name.

- **ID** - a string that uniquely identifies the USB devices on the physical server. The ID never changes even if the device is disconnected from the server and then reconnected again. Please note that if a device ID is listed in quotes, they are a part of the ID and must be included in other calls that use it as an input

parameter.

- **VM UUID** - a universally unique ID of the virtual machine to which this USB device is permanently assigned. If a USB device is not assigned to any virtual machine, this column will be empty.

usb set

Permanently assigns a USB device to the specified virtual machine. A permanently assigned USB device will be connected to the virtual machine automatically every time you start this virtual machine and connect the device to the hardware node. This functionality works only with virtual machines (not containers).

```
prlsrvctl usb set <usb_dev_ID> <VM_UUID>
```

Name	Description
<usb_dev_ID>	The USB device ID. To obtain the list of USB devices connected to the server use the `usb list` command.
<VM_UUID>	The UUID of the virtual machine to which to assign the USB device.

usb del

Deletes a USB device assignment previously created with the `usb set` command.

```
prlsrvctl usb del <usb_dev_ID>
```

Name	Description
<usb_dev_ID>	The USB device ID. To see the current USB device assignments for the current user use the `usb list` command.

2.2.1.7 prlsrvctl user list

Displays the list of Virtuozzo Hybrid Server users. Only those users are displayed who has created at least one virtual machine and container.

```
prlsrvctl user list [-o, --output <name|mng_settings|def_vm_home>]
```

Name	Description
`-o, --output <name|mng_settings|def_vm_home>`	Fields to include in the output. The following fields are available: • `name`, user name. • `mng_settings`, **indicates whether the user is allowed to modify** Virtuozzo Hybrid Server preferences. • `def_vm_home`, user's default virtual machine folder. The fields must be specified in lowercase.

2.2.1.8 prlsrvctl user set

Configures the profile of the user currently logged in to the Virtuozzo Hybrid Server server.

```
prlsrvctl user set [--def-vm-home <path>]
```

Name	Description
`--def-vm-home <path>`	The default virtual machine and container directory name and path.

2.2.1.9 prlsrvctl cttemplate

The `prlsrvctl cttemplate` command is used to manage OS and application EZ templates for containers on the Virtuozzo Hybrid Server server.

list

Lists all the OS and application templates installed on the server.

```
prlsrvctl cttemplate list
```

copy

Copies the specified OS or application template from the local server to the destination server. To copy an application EZ template, additionally specify the `os_template_name` parameter.

```
prlsrvctl cttemplate copy <dst_node> <name> [<os_template_name>] [-f, --force]
```

Name	Description
`<name>`	OS or application template name.
`<os_template_name>`	OS template name. Required for copying application templates.
`<dst_node>`	Destination server specified in the format [<user>[:<password>]@]<server>[:<port>].
`-f, --force`	Specifies whether all validation checks should be skipped.

remove

Removes the specified OS or application template from the server.

```
prlsrvctl cttemplate remove <name> [<os_template_name>]
```

Name	Description
`<name>`	OS or application template name.
`<os_template_name>`	OS template name. Required for deleting application templates.

2.3 System Updates

Virtuozzo Hybrid Server provides various ways to update its components:

- Quick and easy updates of utilities, libraries, kernel and EZ templates with the `yum` utility standard for RPM-compatible Linux operating systems. For more information on `yum`, see Keeping Your System Up To Date and the `yum` manual page.

- The `vzpkg` utility allows you to update OS EZ templates, their caches on the hardware node and software packages inside containers based on application EZ templates. For more information on `vzpkg`, see *EZ Template Management Utilities* on page 69.

Note: To update software in virtual machines, you can use native Linux and Windows updaters.

- ReadyKernel cumulative patches that allow a rebootless alternative to updating the kernel the usual way.

2.3.1 readykernel

`readykernel` is the command-line utility for configuring and displaying status of the Virtuozzo ReadyKernel service and managing ReadyKernel updates.

```
# readykernel <command> [options]
```

Command	Description
`info`	Shows the current ReadyKernel status.
`check-update`	Checks for a newer ReadyKernel patch.
`update`	Downloads, installs and loads the latest ReadyKernel patch for the current kernel.
`autoupdate [enable <hour>|disable]`	Enables or disables daily automatic downloading, installation, and loading of the latest ReadyKernel patches. If enabled, the service will check for updates daily at the specified hour (set in 24-hour format, server time) by means of the `cron.d` script.
`load`	Loads the latest installed ReadyKernel patch for the current kernel.
`autoload <enable|disable>`	Enables or disables automatic loading of the latest installed ReadyKernel patches at boot.
`load-replace`	Unloads all the kernel patches (ReadyKernel and other), then loads the latest installed ReadyKernel patch for the current kernel.
`patch-info`	Shows information about the loaded ReadyKernel patch.
`unload`	Unloads the currently loaded ReadyKernel patch.
`report`	Creates a report for the technical support team.
`help`	Shows help on command usage.

CHAPTER 3

Managing Containers

Virtuozzo containers can be managed using the `prlctl` command-line utility. The utility is installed on the hardware node during the product installation.

3.1 Command-Line Utilities

The table below contains the full list of Virtuozzo Hybrid Server command-line utilities and command you can use for managing containers.

General Utilities

Name	Description
`prlctl`	Utility to manage containers.

Template Management Utilities

Name	Description
`vzpkg`	Utility to manage OS and application EZ templates either inside your containers or on the server itself.

Supplementary Utilities

Name	Description
`vzps, vztop`	Utilities working as the standard `ps` and `htop` utilities, with container-related functionality added.

Table 3.1.3 -- continued from previous page

Name	Description
vzpid	Utility that prints container UUID the process belongs to.
vzsplit	Utility to generate container configuration file sample, "splitting" the server into equal parts.
pfcache	Memory and IOPS deduplication management utility.
pcompact	Utility to compact containers by removing unused blocks from their virtual disks.

3.2 Container Utilities

Virtuozzo containers can be managed using the `prlctl` command-line utility. The utility is installed on the hardware node during the product installation.

3.2.1 prlctl

`prlctl` is the primary tool for container management. To use it, you have to log in to the server as the root user. The following sections describe `prlctl` subcommands.

```
prlctl <command> <CT_name> [<options>] [-v, --verbose <number>] [--timeout <sec>]
       [-l, --login [<user>[:_<passwd>_]@]<server>] [-p, --read-passwd <file>]]
```

Name	Description
<command>	The name of the command to execute.
<CT_name>	The name of the container to perform the operation on. To obtain the list of the available containers, use the `prlctl list` command.
<options>	Command options. See individual commands for available options.
-v, --verbose <number>	Enables verbose output. The greater the <number>, the higher the verbosity.
--timeout <sec>	Use a custom operation timeout specified in seconds. The default timeout for all operations is unlimited.
-l, --login [<user>[:<passwd>]@]<server>	Connect to a remote <server> with the specified credentials. If this flag is omitted, the `prlctl` command is assumed to be run locally.

Continued on next page

Table 3.2.1.1 -- continued from previous page

Name	Description
`-p, --read-passwd <file>`	Use the password from the file `<file>` to log in to a remote hardware node, other credentials being specified with the `--login` option. The `--read-passwd` option can be specified multiple times in order to form a password stack for operations requiring multiple passwords. Each password must be supplied in a separate file.

To display help, enter `prlctl` without any options.

3.2.2 prlctl backup, backup-list, backup-delete, restore

Creates, lists, deletes, or restores container backups.

```
prlctl backup <CT_name|CT_UUID> [-f, --full] [-i, --incremental]
            [-s, --storage [<user>[:<passwd>]@]<server>[:<port>]]
            [--description <desc>] [--no-compression] [--no-tunnel]
prlctl backup-list [<CT_name|CT_UUID>] [-f, --full] [--localvms] [--vmtype ct]
                [-s, --storage [<user>[:<passwd>]@]<server>[:<port>]]
prlctl backup-delete {<CT_name|CT_UUID> | -t, --tag <backup_ID>} [--keep-chain]
prlctl restore {<CT_UUID> | -t, --tag <backup_ID>} [-n, --name <new_name>]
            [--dst=<path>] [-s, --storage [<user>[:<passwd>]@]<server>[:<port>]]
            [--no-tunnel]
```

Name	Description
`<CT_name|CT_UUID>`	Container name or UUID. • Use with `prlctl backup` to create a backup of the specified container. • Use with `prlctl backup-list` to list backups of the specified container. • Use with `prlctl backup-delete` to delete all backups of the specified container. • Use `<CT_UUID>` with `prlctl restore` to restore the most recent backup of the specified container.

Continued on next page

Table 3.2.2.1 -- continued from previous page

Name	Description
`-s, --storage` `[<user>[:<passwd>]@]<server>[:<port>]`	Specifies a remote backup server address, port, and credentials. If this option is omitted, the backup will be saved on the default backup server that can be configured using the `prlsrvctl set` command.
`--description <desc>`	Backup description. Descriptions with white spaces must be enclosed in quotation marks.
`-f, --full`	- Use with `prlctl backup` to create a full backup of the container. A full backup contains all container data. - Use with `prlctl backup-list` to display full backup information.
`-i, --incremental`	Create an incremental backup of the container. An incremental backup contains only the files changed since the previous full or incremental backup. This is the default backup type.
`--localvms`	List local backups only.
`--vmtype ct`	List container backups on the server.
`-t, --tag <backup_ID>`	The ID of the backup to restore or delete.
`-n, --name <new_name>`	A new name to assign to the restored container. If this option is omitted, the container will be restored with the original name.
`--dst=<path>`	Restore the container to the specified directory on the server. If this option is omitted, the container will be restored to `/vz/private/<CT_UUID>`.
`--no-compression`	Do not compress the created backup image.
`--no-tunnel`	Disables connection tunneling for backup. Connection tunneling provides secure data transmission. To use this option, configure the firewall of the destination server to allow incoming connections on any port on the corresponding network interface.

Continued on next page

Table 3.2.2.1 -- continued from previous page

Name	Description
`--keep-chain`	Preserve the rest of the backup chain when deleting specific backups.

3.2.3 prlctl clone

Creates an exact copy of the specified container.

```
prlctl clone <CT_name> --name <new_name> [--template] [--dst=<path>]
```

Name	Description
`<CT_name>`	Name of the container to clone.
`--name <new_name>`	Name to be assigned to the new container.
`--template`	Create a container template instead of a clone. Template cannot be started.
`--dst=<path>`	Path to the directory where the `<CT_UUID>` directory with cloned container private area will be stored. If this parameter is omitted, the clone is created in the default directory `/vz/private`.

3.2.4 prlctl console

Creates a command prompt channel to a container. Allows to log in to and execute commands in running containers as well as attach to stopped containers to get information on their startup from bootstrap programs (such as `init`) for troubleshooting purposes. Logging in to containers requires a virtual terminal (e.g., mingetty) to be installed in the container.

Note: To exit the console, press **Esc** and then **.** (period).

```
prlctl console <CT_name>
```

Name	Description
`<CT_name>`	Container name.

3.2.5 prlctl create

This command is used to create new containers.

```
prlctl create <CT_name> --vmtype ct [<options>]
```

With this command, you can create regular containers. A unique container name is required for this command.

Name	Description
`<CT_name>`	An arbitrary name to assign to the new container.
`--vmtype ct`	Tells the `prlctl create` command to make a container. If the option is omitted, a virtual machine is created instead.
`--ostemplate <name>`	OS EZ template to use for creating the container. If omitted, this value is taken from the `DEF_OSTEMPLATE` parameter in the global Virtuozzo Hybrid Server configuration file.
`--config <name>`	Container sample configuration file to use for creating the container. Sample configuration files are located in `/etc/vz/conf` and have names in the format `ve-<name>.conf-sample`. The sample configuration files usually have a number of resource control limits for the container and some application templates to be added to the container immediately upon its creation. If you skip this option and the default configuration file name is not specified in the global Virtuozzo Hybrid Server configuration file, you will have to set resource control parameters for the container using the `prlctl set` command.
`--uuid <uuid>`	A custom UUID to assign to the container.

3.2.6 prlctl delete

Deletes a container from the server.

```
prlctl delete <CT_name>
```

Name	Description
`<CT_name>`	Container name.

When executed, `prlctl delete` physically removes all the files located in the container private area (specified as the `VE_PRIVATE` variable in the container configuration file) and renames the container configuration file in

/etc/vz/conf from <CT_name>.conf to <CT_name>.conf.destroyed. It also renames container action scripts, if any, in a similar manner.

Note: A container must be stopped before its private area can be unmounted.

3.2.7 prlctl exec, enter

Allow running arbitrary commands in a container.

```
prlctl exec <CT_name> [--without-shell] <command>
prlctl enter <CT_name>
```

where <command> is a string to be executed in the container. If <command> is specified as -, then the commands for execution will be read from the standard input until the end of file or exit is encountered.

Name	Description
<CT_name>	Container name.
--without-shell	Run commands directly without bash or cmd shell.

When using prlctl exec, remember that the shell parses the command-line and, if your command has shell metacharacters in it, you should escape or quote them.

The prlctl enter command is similar to prlctl exec /bin/bash. The difference between the two is that prlctl enter makes the shell interpreter believe that it is connected to a terminal. As such, you receive a shell prompt and are able to execute multiple commands as if you were logged in to the container.

3.2.8 prlctl migrate

Migrates a container from one server to another.

```
prlctl migrate <CT_name> <destination_server>[/<CT_name>]
            [--dst=<path>] [--clone|--remove-src] [--no-compression] [--ssh <options>]
```

Name	Description
<CT_name>	The source container name.

Table 3.2.8.1 -- continued from previous page

Name	Description
`<source_server>`	The source server information. Use the following format to specify this info: `[<user>[:<password>]@]<server_IP_address_or_hostname>[:<port>]`.
`<destination_server>`	The destination server information. If omitted, the migration will be performed locally. Use the following format to specify this info: `[<user>[:<password>]@]<server_IP_address_or_hostname>[:<port>]`.
`--dst=<path>`	Path to the directory on the destination server where the `<CT_UUID>` directory with container private area will be stored.
`--clone`	Clone the original container to the destination server and leave it intact on the source server. The clone will have a different UUID, MAC address, and offline management disabled. If this option is omitted, the original container will be removed from the source server after migration. Cannot be used together with `--remove-src`.
`--remove-src`	Remove the original container from the source server. Cannot be used together with `--clone`.
`--no-compression`	Disable data compression during migration.
`--ssh`	Additional options to pass to `ssh` to connect to the destination server. All standard `ssh` options are supported. **Note:** Do not specify the destination server hostname or IP address as an `ssh` option.

3.2.9 prlctl mount, umount

The `prlctl mount` command mounts the container private area to the container root directory (`/vz/root/<CT_name>` on the server) without starting it. Normally, you do not have to use this command as the `prlctl start` command mounts the container private area automatically.

The `prlctl umount` command unmounts the container private area. Usually, there is no need in using this command either because `prlctl stop` unmounts the container private area automatically.

Note: These commands can trigger the execution of action scripts (see *Action Scripts* on page 68).

```
prlctl mount  <CT_name>
prlctl umount <CT_name>
```

Name	Description
<CT_name>	Container name.

3.2.10 prlctl move

Moves the directory with container private area to a new location on the same server.

```
prlctl move <CT_name> --dst=<path>
```

Name	Description
<CT_name>	Container name.
--dst=<path>	New location of the <CT_UUID> directory with container private area.

3.2.11 prlctl problem-report

Generates a problem report for the specified container and either sends it to the Virtuozzo technical support team or displays it on the screen.

```
prlctl problem-report <CT_name> -d, --dump | -s, --send \
                      [--proxy [<user>[:<passwd>]@]<proxyhost>[:<port>] | --no-proxy]
```

Name	Description
<CT_name>	The name of the container for which to generate the problem report.
-d, --dump	Collect technical data about the specified container and display it on the screen. You can also pipe the output to a file and then send it to the technical support team to analyze your problem.

Table 3.2.11.1 -- continued from previous page

Name	Description
`-s, --send`	Send the generated problem report to the Virtuozzo technical support team.
`--proxy [<user>[:<passwd>]@<proxyhost>[:<port>]`	Use the specified information to send the generated report through a proxy server, if you use one to connect to the Internet.
`--no-proxy`	Do not use a proxy server to send the generated report. This is the default behavior, so you can omit this parameter.

3.2.12 prlctl register, unregister

The `register` command is used to register a container with Virtuozzo Hybrid Server.

The `unregister` command removes a container from the Virtuozzo Hybrid Server registry.

```
prlctl register <path> [--preserve-uuid | --uuid <UUID>]
prlctl unregister <CT_name>
```

Name	Description
`<path>`	Full path to the container directory.
`<CT_name>`	The name of the container to remove from the Virtuozzo Hybrid Server registry.
`--preserve-uuid`	Do not change the container UUID. If ommited, the UUID is regenerated.
`--uuid <UUID>`	Change the container UUID to the specified one. If ommited, the UUID is regenerated.

- Use the `register` command when you have a container on the server that does not show up in the list of the containers registered with the Virtuozzo Hybrid Server. This can be a container that was previously removed from the registry or that was copied from another location.

- The `unregister` command removes a container from the Virtuozzo Hybrid Server registry, but does not delete the container files from the server. You can re-register the container later using the `register` command.

3.2.13 prlctl reinstall

Recreates a container's root virtual disk, installs all applications previously installed from application templates, copies credentials from the old container (unless `--resetpwdb` is specified), and renames the old root directory `'/'` to `'/old'` (unless `--no-backup` is specified).

```
prlctl reinstall <ve_id|ve_name> [--no-backup] [--resetpwdb] [--ostemplates <name>]
```

Name	Description
`<ve_id|ve_name>`	Container ID and name.
`--no-backup`	Does not save the old private area contents to the `/old` directory.
`--resetpwdb`	Removes the container's user database and creates a clean database like for any new installation.
`--ostemplates <name>`	The name of OS templates.

3.2.14 prlctl set

This command is used for setting container parameters.

```
prlctl set <CT_name> <option> <value>
```

where <CT_name> is container name.

The command options specified in this file can be subdivided into the following categories:

- Miscellaneous
- Networking
- Resource management
- Hard disk drive management

3.2.14.1 General Options

The table below lists the general options you can use with `prlctl set`.

Name	Description
`--autostart <on|off|auto>`	Sets the container startup options: • `on` - automatically start the container when the hardware node starts. • `off` (default) - do not automatically start the container when the hardware node starts. • `auto` - let the container assume the state it has been in before the hardware node reboot/shutdown.
`--offline-management <yes|no>`	Enabling/disabling the direct managing of the container through a common Internet browser by means of Power Panels and the Plesk control panel (as defined by the `OFFLINE_SERVICE` parameter in the global or container configuration file).
`--offline-service <service_name>`	Defines whether the container can be managed by means of Power Panel or Plesk or both. Valid only if the `OFFLINE_MANAGEMENT` parameter is set to `yes`. The names of the available services can be taken from the file names (excluding the `.conf` extension) in the `/etc/vzredirect.d` directory on the server.
`--userpasswd <user>:<password>`	This setting creates a new user with the specified password in the container, or changes the password of an already existing user. This command modifies not the container configuration file, but the `/etc/passwd` and `/etc/shadow` files inside the container. In case the container root is not mounted, it is automatically mounted to apply the changes and then unmounted.
`--crypted`	Used with `--userpasswd`. Indicates that the specified password is already a hash.

Continued on next page

Table 3.2.14.1.1 -- continued from previous page

Name	Description
`--features {<name>:on|off}`	Enables/disables the support for the following functionality inside the container: • `nfs`: mounting NFS shares, • `ipip`: creating IPIP tunnels, • `sit`: using the Simple Internet Transition (SIT) mechanisms, • `bridge`: using bridges to connect virtual Ethernet devices, • `nfsd`: running an NFS-kernel-space server, • `time`: changing the system time.
`--name <new_name>`	Changes the container name. You can change the names of both stopped and running containers.
`--description <desc>`	Custom container description. Descriptions must be alphanumeric. Descriptions with white spaces must be enclosed in quotation marks.
`--vnc-mode <auto|manual|off>`	Enables or disables access to the container via the VNC protocol.
`--vnc-port <port>`	Sets the VNC port number for the container. Used with `--vnc-mode manual`.
`--vnc-passwd <passwd> --vnc-nopasswd`	Sets the VNC password for the container or specifies that no password is needed for VNC connections. Either of these options is mandatory for any VNC setup.
`--autocompact <on|off>`	Enables or disables compaction for all disks in the container. **Note:** For details on how to enable or disable compaction for a specific disk in the container, see *Hard Disk Drive Management Options* on page 60.

3.2.14.2 Resource Management Options

Resource management options control the amount of resources a container may consume. If the setting has `bar:lim` after it than this setting requires specifying both barrier and limit values separated by colons.

Name	Description
`--applyconfig <name>`	This option lets you set the resource parameters for the container not one by one, but by reading them from the container sample configuration file. All container sample configuration files are located in the `/etc/vz/conf` directory and are named according to the following pattern: `ve-<name>.conf-sample`, so you should specify only the `<name>` part of the corresponding sample name after the `--applyconfig` option. Note that the names of sample configuration files cannot contain spaces. The `--applyconfig` option applies all the parameters from the specified sample file to the given container, except for the `OSTEMPLATE`, `TEMPLATES`, `VE_ROOT`, `VE_PRIVATE`, `HOSTNAME`, `IP_ADDRESS`, `TEMPLATE`, `NETIF` parameters (if they exist in the configuration sample file).
`--cpuunits <units>`	CPU weight. This is a positive integer number that defines how much CPU time the container can get as compared to the other virtual machines and containers running on the server. The larger the number, the more CPU time the container can receive. Possible values range from 8 to 500000. If this parameter is not set, the default value of 1000 is used.
`--cpulimit {<percent>|<megahertz>}`	CPU limit, in per cent or megahertz (MHz), the container is not allowed to exceed. This parameter is not set for newly created containers; so they can consume all free CPU power of the server. By default, the limit is set in percent. To set the limit in MHz, specify `m` after the value. When setting this parameter in per cent, keep in mind that one CPU core makes up 100%. So if the server has 4 CPU cores, the total CPU power will equal 400%.

Continued on next page

Table 3.2.14.2.1 -- continued from previous page

Name	Description
`--cpus <number>`	Number of CPU cores per CPU socket defining the CPU limit for a container. The limit is calculated by multiplying the power of one CPU core by the number of the specified CPU cores. This option also defines the number of CPUs shown to container users. This parameter is not set for newly created containers; so they can consume all free CPU power of the server.
`--cpumask <number>`	CPU affinity mask. This mask defines the CPUs on the server that can be used to handle the processes running in the container. The CPU mask can be specified as both separate CPU index numbers (1,2,3) and CPU ranges (2-4,5-7).
`--nodemask {<number>|all}`	The NUMA node mask defining a NUMA node to bind the container to. Once you set the mask, the processes running in the container will be executed only on the CPUs that belong to the specified NUMA node. You can specify a list of NUMA nodes by their index numbers separated by commas and as a range (e.g., `0,1,2,3,4-6`). To make all NUMA nodes available for container's processes specify `--nodemask all`.
`--quotaugidlimit {<0|<N>}`	Enables (if set to a value other than `0`) or disables (if set to `0`) per-user/group quotas for further management with the standard Linux `quota` utility. Keep in mind the following: • Enabling per-user and per-group quotas for a container requires restarting the container. • If you delete a registered user but some files with their ID continue residing inside your container, the current number of UGIDs (user and group identities) inside the container will not decrease. • If you copy an archive containing files with user and group IDs not registered inside your container, the number of UGIDs inside the container will increase by the number of these new IDs.

Continued on next page

Table 3.2.14.2.1 -- continued from previous page

Name	Description
`--ioprio <number>`	The container priority for disk I/O operations. The allowed range of values is `0-7`. The greater the priority, the more time the container has for writing to and reading from the disk. The default container priority is `4`.
`--iolimit <number>`	The bandwidth a container is allowed to use for its disk input and output (I/O) operations. By default, the limit is set in megabytes per second. You can use the following suffixes to specify measurement units: • `G`: sets the limit in gigabytes per second • `M`: sets the limit in megabytes per second • `K`: sets the limit in kilobytes per second • `B`: sets the limit in bytes per second In the current version of Virtuozzo Hybrid Server, the maximum I/O bandwidth limit you can set for a container is 2 GB per second. The default I/O bandwidth limit for all newly created containers is set to 0, which means that no limits are applied to any containers.
`--iopslimit <number>`	The maximum number of disk input and output operations per second a container is allowed to perform. By default, any newly created container does not have the IOPS limit set and can perform so many disk I/O operations per second as necessary.
`--rate <class>:<Kbits>`	If traffic shaping is turned on, then this parameter specifies bandwidth guarantee for the container. The format is `<class>:<Kbits>` where `<class>` is the network class (group of IP addresses) and `<Kbits>` is the traffic bandwidth.
`--ratebound <yes|no>`	If set to "yes", the bandwidth guarantee is also the limit for the container and the container cannot borrow the bandwidth from the `TOTALRATE` bandwidth pool.

Table 3.2.14.2.1 -- continued from previous page

Name	Description
`--memsize <size>`	The amount of RAM that can be used by the processes of a container, in megabytes. You can use the following suffixes to specify measurement units: • `G` for gigabytes • `M` for megabytes • `K` for kilobytes • `B` for bytes
`--memguarantee <size>`	Sets a percentage of container's RAM that said container is guaranteed to have. By default, set to 0%.
`--swappages <pages>`	The amount of swap space that can be used by the container for swapping out memory once the RAM is exceeded, in 4KB pages. You can use the following suffixes to specify measurement units: • `G` for gigabytes • `M` for megabytes • `K` for kilobytes • `B` for bytes
`--swap <size>`	The amount of swap space that can be used by the container for swapping out memory once the RAM is exceeded, in bytes. You can use the following suffixes to specify measurement units: • `G` for gigabytes • `M` for megabytes • `K` for kilobytes • `B` for bytes

3.2.14.3 Network Options

Network-related options allow you to set the hostname and configure `iptables` modules.

Name	Description
`--hostname <name>`	Sets the hostname to the specified name.
`--apply-iponly <yes|no>`	If set to `yes`, the hostname, nameserver, and search domain settings from the container configuration file are ignored.
`--netfilter <disabled|stateless|stateful|full>`	Indicates which `iptables` modules are allowed for the container. If some of the allowed modules are not loaded on the destination Hardware Node after migration or restoration from backup, they will be automatically loaded on the migrated or restored container start. The following modes are available: • **disabled**: none. • **stateless**: all modules except `conntrack` and NAT-related. • **stateful**: all modules except NAT-related. • **full**: (default) all modules.

You can also set other network parameters, such as the DNS server address and the IP addresses (both IPv4 and IPv6), when adding a new or configuring the existing container network adapter:

```
prlctl set <CT_name> {--device-add net | --device-set net<N>} <options>
```

Name	Description
`<CT_name>`	Container name.
`--device-add net`	Adds a new virtual network adapter to the container.
`--device-set net<N>`	Modifies an existing virtual network adapter. To obtain the list of the available adapters, use the `prlctl list` command with the `--info` option.
`--type routed`	Sets the networking mode for the virtual network adapter to "routed". In this mode, the network adapter is communicating with the outside world through an internal virtual network adapter.

Continued on next page

Table 3.2.14.3.2 -- continued from previous page

Name	Description
`--network <network_ID>`	Sets the networking mode for the virtual network adapter to "virtual_network". In this mode the adapter is connected to a virtual network specified by `<network_ID>`.
`--mac {<addr>|auto}`	Specifies the MAC address to assign to an existing network adapter. Specify a desired MAC address using the `addr` parameter value or use the `auto` option to generate the existing address automatically.
`--ipadd <addr>[/<mask>]`	Adds an IP address and a mask (optional) to the network adapter.
`--ipdel <addr>[/<mask>]`	Deletes an IP address from the network adapter.
`--dhcp <yes|no>`	Specifies whether the virtual network adapter should obtain the IPv4 settings through a DHCP server.
`--dhcp6 <yes|no>`	Specifies whether the virtual network adapter should obtain the IPv6 settings through a DHCP server .
`--gw <gw>`	The default gateway to be used by the container.
`--gw6 <gw>`	The default IPv6 gateway to be used by the container.
`--nameserver <addr>`	The default DNS server address to be used by the container.
`--searchdomain <addr>`	The default search domain to be used by the container.
`--configure <yes|no>`	If set to `yes`, the settings above are applied to the virtual network adapter instead of its original settings. Configuring any of the settings above automatically sets this option to `yes`.
`--ipfilter <yes|no>`	Determines if the specified network adapter is configured to filter network packages by IP address. If set to `yes`, the adapter is allowed to send packages only from IPs in the network adapter IP addresses list.
`--macfilter <yes|no>`	Determines if the specified network adapter is configured to filter network packages by MAC address. If set to `yes`, the adapter is allowed to send packages only from its own MAC address.
`--preventpromisc <yes|no>`	Determines if the specified network adapter should reject packages not addressed to the container. If set to `yes`, the adapter will drop such packages.

3.2.14.4 Hard Disk Drive Management Options

This group of options is used to manage virtual hard disks in a container. The first syntax uses a file to emulate a hard disk drive. The second syntax connects a physical hard disk on the host server to the container.

```
prlctl set <CT_name> {--device-add hdd |--device-set hdd<N>}
          [--image <file> [--recreate]] [--size <size>]
          [--mnt <path>] [--iface <ide|scsi|virtio>]
          [--position <pos>] [--autocompact <on|off>]
prlctl set <CT_name> ----device-add hdd --device <dev_name>
          [--mnt <path>] [--iface <ide|scsi|virtio>] [--position <pos>]
prlctl set <CT_name> --backup-add <backup_ID> [--disk <disk_name>]
prlctl set <CT_name> --device-del hdd<N> [--detach-only|--destroy-image]
prlctl set <CT_name> --backup-del {<backup_ID>|all}
```

Name	Description
`<CT_name>`	Container name.

`--device-add hdd`	Adds a virtual hard disk to the container. If no other options are specified, the command creates a new unmounted disk with the following parameters:
	  • name: `hdd<N>` where `<N>` is the next available disk index.
	  • size: 65536 MB
	  • image location: `/vz/private/<CT_UUID>/disk-<ID>.hdd`.

`--device-set hdd<N>`	Modifies the parameters of the virtual hard disk `hdd<N>`. **Note:** For the list of disks, use the `prlctl list -i` command.
`--image <file>`	Specifies an existing image file that will be used to emulate the virtual disk. To recreate the image file, add the `--recreate` option.
`--device <dev_name>`	This option is used to connect a physical hard disk on the hardware node to the container. You can obtain the names of the existing hard disks on the server using the `prlsrvctl info` command.
`--size <size>`	Specifies the size of the virtual hard disk, in megabytes.

Table 3.2.14.4.1 -- continued from previous page

Name	Description
`--mnt <path>`	Specifies the mount point of the virtual hard disk inside the container. A corresponding entry is also added to container's `/etc/fstab` file, so the disk is mounted automatically on container start.
`--autocompact <on|off>`	Used with `--device-set hdd<N>`. Enables or disables compaction for the specified disk in the container. **Note:** For details on how to enable or disable compaction for all disks in the container, see *General Options* on page 51.
`--backup-add <backup_ID>`	Attach the backup with the identifier `<backup_ID>` to the virtual machine as a virtual hard disk. To obtain the backup ID, use the `prlctl backup-list -f` command.
`--disk <disk_name>`	Used with `--backup-add`. The name of the disk in the backup to attach. If a disk is not specified, all disks contained in the backup will be attached. To obtain the disk name, use the `prlctl backup-list -f` command.
`--device-del hdd<N>`	Deletes a virtual hard disk from the stopped container.
`--detach-only`	Removes the virtual disk from the container configuration but leaves its image file intact.
`--destroy-image`	Removes the virtual disk from the container configuration and deletes its image file.
`--backup-del {<backup_ID>|all}`	Detach either the backup with the identifier `<backup_ID>` or detach all backups from the virtual machine.

3.2.15 prlctl snapshot, snapshot-list, snapshot-switch, snapshot-delete

Takes, displays, reverts to, and deletes container snapshots.

Note: Taking a snapshot of a running container saves not only its file system state but in-memory state (checkpoint) as well.

```
prlctl snapshot <CT_name> [-n, --name <name>] [-d, --description <desc>]
prlctl snapshot-list <CT_name> [-t, --tree] [-i, --id <snapshot_ID>]
prlctl snapshot-switch <CT_name> -i, --id <snapshot_ID>
prlctl snapshot-delete <CT_name> -i, --id <snapshot_ID> [-c,--children]
```

Name	Description
`<CT_name>`	Container name.
`-n, --name <name>`	User-defined snapshot name. Names with white spaces must be enclosed in quotation marks.
`-d, --description <desc>`	User-defined snapshot description. Descriptions with white spaces must be enclosed in quotation marks.
`-t, --tree`	Displays the snapshot list as a tree. The default display format is tabular with **Parent Snapshot ID** and **Snapshot ID** as columns.
`-i, --id <snapshot_ID>`	- Use with `prlctl snapshot-list` to specify the ID of the snapshot to use as the root. If this parameter is omitted, the entire snapshot tree will be displayed. - Use with `prlctl snapshot-switch` to specify the ID of the snapshot to revert to. - Use with `prlctl snapshot-delete` to specify the ID of the snapshot to delete.
`-c,--children`	If the snapshot you want to delete has children snapshots derived from it, they will be deleted. If the option is omitted, they become the children of the deleted snapshot's parent.

3.2.16 prlctl start, stop, restart, status

These commands start, stop, restart, and show the current state of containers, respectively.

```
prlctl start <CT_name> [--wait]
prlctl stop <CT_name> [--fast]
prlctl restart <CT_name>
prlctl status <CT_name>
```

Name	Description
<CT_name>	Container name.

The first command is used to start a container. It will set up all network interfaces inside the container, initialize the container quota, if needed, start the `init` process inside the container, and exit. You can also make the `prlctl start` command wait for all the necessary startup processes to complete and the container to boot into the default runlevel by passing the `--wait` option to this command.

`prlctl stop` shuts the container down. If the container is not down after a two-minute timeout due to an error in an application, for example, `prlctl` will forcibly kill all the processes inside the container. To avoid waiting for two minutes in case of a corrupted container, you may use the `--fast` option with this command.

The `prlctl restart <CT_name>` command consecutively performs the stopping and starting of the corresponding container.

Note: The commands described above can trigger the execution of action scripts (see *Action Scripts* on page 68).

The `prlctl status` command shows the current container state. It outputs the following information: whether the container private area exists, whether it is mounted, and whether the container is running.

3.2.17 prlctl suspend, resume

The `prlctl suspend` command is used to save the state of a running container.

```
prlctl suspend <CT_name>
```

Name	Description
<CT_name>	Container name.

During the `prlctl suspend` execution, the current container state is saved to a special dump file and the container itself is stopped. The created dump file is saved to the `Dump` file in the `/vz/private/<CT_UUID>/dump` directory on the server.

The `prlctl resume` command is used to restore the container from its dump file created with the `prlctl suspend` command.

```
prlctl resume <CT_name>
```

When executed, `prlctl resume` searches for the `Dump` file in the `/vz/private/<CT_UUID>/dump` directory on the server and restores the container from this file.You can restore the container dump file on the Source Server, i.e. on the server where this container was running before its dumping, or transfer the dump file to another server and restore it there.

Note: Before restoring a container from its dump file, make sure that the file system on the Destination Server is identical to that at the moment of the container dumping. Otherwise, the container restoration may fail.

3.2.18 prlctl list

Displays a list of containers on the Hardware Node. Displays information on containers on the Hardware Node.

```
prlctl list --vmtype ct [-a, --all] [-o, --output <field>[,...]]
                        [-s, --sort <field>|-<field>] [-t, --template] [-j, --json]
prlctl list -i, --info --vmtype ct [<CT_name>] [-f, --full] [-t, --template]
                                   [-j, --json]
```

Name	Description
`-a, --all`	List all running, stopped, suspended, and paused containers. If this and the rest of the parameters are omitted, only the running containers will be displayed.
`-t, --template`	List available container templates instead of actual containers.
`-o, --output <field>[,…]`	Display only the specified fields. Type field names in lower case. Separate multiple fields with commas. For the list of fields, see *prlctl list Output Parameters* on page 65.
`-s, --sort {<field>|-<field>}`	Sort containers by the specified field in either ascending or descending order.
`-i, --info`	Display detailed information about the specified container.
`-f, --full`	Display detailed information about network cards in containers. Used with the `--info` option.

Continued on next page

Table 3.2.18.1 -- continued from previous page

Name	Description
`<CT_name>`	Thename of the container for which to display the detailed information. If not specified, the information will be displayed for all registered containers.
`-j, -json`	Produce machine-readable output in the JSON format.

3.2.18.1 prlctl list Output Parameters

Listed below are the parameters that can be specified after the `-o` switch.

Name	Output Column	Description
`uuid`	`UUID`	Container UUID.
`hostname`	`HOSTNAME`	Container hostname.
`name`	`NAME`	Container name.
`description`	`DESCRIPTION`	Container description.
`ostemplate`	`OSTEMPLATE`	Specifies the name of the OS template your container is based on (e.g., `almalinux-8-x86_64`).
`ip`	`IP_ADDR`	Container IP address.
`status`	`STATUS`	Container status (e.g., running or stopped).
`numproc`	`NPROC`	The number of threads allowed.
`mac`	`MAC`	Network device's MAC address.
`netif`	`NETIF`	Network devices in the container.
`iolimit`	`IOLIMIT`	The bandwidth a container is allowed to use for its disk input and output (I/O) operation, in bytes per second.
`ha_enable`	`HA_ENABLE`	Indicates whether the container is joined to the High Availability Cluster.
`ha_prio`	`HA_PRIO`	Container priority in the High Availability Cluster (0 is the lowest). Higher-priority virtual environments are restarted first in case of failures.

3.2.19 prlctl statistics

Print statistics for running containers on the server.

```
prlctl statistics {<CT_UUID_or_name>|-a, --all} [--loop] [--filter <filter>]
```

Name	Description
`-a, --all`	Print statistics for all virtual machines and containers on the server.
`--loop`	Print statistics every second until the program is terminated.
`--filter <pattern>`	Specifies the subset of performance statistics to collect and print. If omitted, all available statistics is shown. Asterisks (*) can be used as wildcards for any number of arbitrary characters. The available filters are listed below (`<N>` is the device or file system index).

3.2.19.1 Available Filters

Storage device statistics

- `devices.{ide|scsi|sata}<N>.read_requests` - Total count of read requests to IDE, SCSI, or SATA controller

- `devices.{ide|scsi|sata}<N>.read_total` - Total count of read bytes for IDE, SCSI, or SATA controller

- `devices.{ide|scsi|sata}<N>.write_requests` - Total count of write requests to IDE, SCSI, or SATA controller

- `devices.{ide|scsi|sata}<N>.write_total` - Total count of written bytes for IDE, SCSI, or SATA controller

Network statistics

- `net.nic<N>.pkts_in` - Total number of incoming packets for network adapter

- `net.nic<N>.pkts_out` - Total number of outgoing packets for network adapter

- `net.nic<N>.bytes_in` - Total number of incoming bytes for network adapter

- `net.nic<N>.bytes_out` - Total number of outgoing bytes for network adapter

Classful network statistics

The result is provided in five columns: **Class**, **Input(bytes)**, **Input(packets)**, **Output(bytes)**, **Output(packets)**.

- `net.classful.traffic` - Total counters for IPv4 and IPv6 traffic

- `net.classful.traffic.ipv4` - Counters for IPv4 traffic

- `net.classful.traffic.ipv6` - Counters for IPv6 traffic

CPU statistics

- `guest.cpu.usage` - Guest OS CPU usage, in percent

- `guest.cpu.time` - Sum of guest CPU time differences since the last query for each vCPU averaged by the number of host CPUs, in microseconds

- `host.cpu.time` - Sum of host CPU time differences since the last query for each vCPU averaged by the number of host CPUs, in microseconds

- `guest.vcpu<N>.time` - per-vCPU statistics, in nanoseconds

RAM statistics

- `guest.ram.usage` - Guest OS used RAM, in MiB

- `guest.ram.cached` - Guest OS cached RAM, in MiB

- `guest.ram.total` - Guest OS total RAM, in MiB

- `guest.ram.swap_in` - Guest OS virtual memory stats, in counts

- `guest.ram.swap_out` - Guest OS virtual memory stats, in counts

- `guest.ram.minor_fault` - Guest OS minor page fault count

- `guest.ram.major_fault` - Guest OS major page fault count

- `guest.ram.balloon_actual` - Guest OS balloon size, in MiB

Mounted file systems statistics

- `guest.fs<N>.name` - Device name as seen from inside the guest file system

- `guest.fs<N>.total` - Total size of the file system, in bytes

- `guest.fs<N>.free` - Amount of free space on the file system, in bytes

- `guest.fs<N>.disk.<N>` - Disk indices

3.2.20 Action Scripts

Action scripts can be used to perform actions on containers at various stages of container operation. The following `prlctl` commands can trigger action scripts: `start`, `stop`, `restart`, `mount`, and `umount`.

Two types of scripts are supported: global, triggered for each container on host, and per-container, triggered for specific containers.

Some of the scripts (default) are shipped with Virtuozzo Hybrid Server. Other are not present by default but can be created manually and must be assigned specific file names to be triggered.

If you need to extend the functionality of default action scripts, it is recommended to create additional custom scripts and call them from the default scripts.

3.2.20.1 Default Action Scripts

Default action scripts are located in the `/usr/libexec/libvzctl/scripts/` directory.

Name	Description
`vz-create_prvt`	Creates a container private area from a private area template.
`vz-net_add`	Sets up the necessary routing entries for container IP addresses and adds public ARP records on all interfaces.
`vz-net_del`	Deletes routing entries and ARP records for container IP addresses from all interfaces.
`vz-pci_configure`	Executed after a PCI device is added to or removed from a container.
`vz-setrate`	Configures the network traffic shaping for a container.
`vz-start`	Called just before a container is started; used for additional container configuration (such as network setup).
`vz-stop`	Called just after a container is shut down; used for additional container cleanup (such as network cleanup).

3.2.20.2 Manually Created Action Scripts

Action scripts recognized by Virtuozzo Hybrid Server but not shipped by default can be of two types:

- Global, executed for all containers on host. Such scripts must have the prefix `vps` (e.g., `vps.mount`) and need to be placed in `/etc/vz/conf/`.

- per-container, executed for specific containers. Such scripts must not have the prefix `vps` and need to be placed in `/vz/private/<CT_UUID>/scripts/`.

Name	Description
`vps.mount, mount`	Executed after a container is mounted. Can be global or container-specific.
`start`	Executed in container context on container start.
`stop`	Executed in container context on container stop.
`vps.umount, umount`	Executed before a container unmounted. Can be global or container-specific.

Note: All action scripts except `start` and `stop` are executed in the host context. The `start` and `stop` scripts are executed in the container context.

The environment passed to the `mount` and `umount` scripts is the standard environment of the parent (e.g., `prlctl`) with two additional variables: `$VEID` and `$VE_CONFFILE`. The first has the container UUID and the second has the full path to container's configuration file. Other container configuration parameters required for the script (such as `$VE_ROOT`) can be obtained from the global and per-container configuration files.

3.3 EZ Template Management Utilities

This section describes the utilities you can use to manage OS and application templates.

3.3.1 vzpkg

The `vzpkg` utility is used to manage OS and application EZ templates either inside your containers or on the server itself. This tool can also be used to manage standard software packages (e.g., `mysql.rpm`) inside containers.

```
vzpkg <command> [<options>] {<CT_name>|<object>}
vzpkg --help
```

Subcommands:

Name	Description
`install template`	Installs OS and application EZ templates on the server.
`update template`	Updates OS and application EZ templates installed on the server.
`remove template`	Removes OS and application EZ templates from the server.
`list`	Outputs a list of EZ templates, OS template caches with preinstalled application templates, or software packages either on the server or inside a particular container.
`info`	Outputs information on any EZ templates or software packages available on the server or inside the container.
`status`	Outputs information on updates for the packages installed inside a container.
`install`	Adds application EZ templates to or to install software packages inside the container.
`update`	Updates application EZ templates and software packages inside the container.
`remove`	Removes application EZ templates or software packages from the container.
`create cache`	Creates a tarball (cache) for the given OS EZ template.
`update cache`	Updates the existing tarball (cache) for the given OS EZ template.
`remove cache`	Removes a tarball (cache) for the given OS EZ template.
`create appcache`	Creates a cache of an OS EZ template with preinstalled application templates.
`update appcache`	Updates or recreates a cache of an OS EZ template with preinstalled application templates.
`remove appcache`	Removes a cache of an OS EZ template with preinstalled application templates.
`localinstall`	Installs a software package inside a container from the corresponding file on the server.
`localupdate`	Updates the software packages installed inside your container(s) by means of the `vzpkg install` or `vzpkg localinstall` commands.
`upgrade`	Upgrades an OS EZ template the container is based on to a newer version.
`fetch`	Downloads packages included in EZ templates to the server and to store them in the `vzpkg` local cache.
`clean`	Removes all locally cached data from the template directories on the server.
`update metadata`	Updates the local metadata on the server.

3.3.2 vzpkg install template

This command is used to install an OS or application EZ template on the server from an RPM package or Virtuozzo Hybrid Server repositories.

```
vzpkg install template [<options>] <object> [...]
```

where `<object>` is a path to an RPM package or an EZ template name.

Name	Description
-q, --quiet	Disables logging to the screen and to the log file.
-f, --force	Forces installation of the EZ template on the server.

Note: To install multiple EZ templates, specify multiple RPM package or EZ template names separated by white spaces.

3.3.3 vzpkg update template

This command is used to update an OS or application EZ template on the server from an RPM package or Virtuozzo Hybrid Server repositories.

```
vzpkg update template [<options>] <object> [...]
```

where `<object>` is a path to an RPM package or an EZ template name.

Name	Description
-q, --quiet	Disables logging to the screen and to the log file.
-f, --force	Forces update of the EZ template.

Note: To update multiple EZ templates, specify multiple RPM package or EZ template names separated by white spaces.

3.3.4 vzpkg remove template

This command removes an OS or application EZ template from the server.

```
vzpkg remove template [<options>] <template_name> [...]
```

Name	Description
`-F, --for-os` `<OS_template>`	Specifies the OS EZ template to delete the application EZ template from.
`-q, --quiet`	Disables logging to screen and file.
`-f, --force`	Forces deletion of the EZ template.

When executed, the `vzpkg remove template` command removes the specified OS EZ template from the server. To delete an application EZ template, additionally specify the name of the OS EZ template (`<OS_template>`) under which this application template is to be run.

3.3.5 vzpkg list

The `vzpkg list` command is used to list

- EZ templates installed on the server, in a container, or available in remote EZ template repositories
- YUM software groups or individual packages installed in a container

```
vzpkg list [<options>] [<OS_template>|<CT_name> [...]]
```

If you indicate a container name, the command will list all EZ templates applied to the specified container. If you indicate an OS EZ template, `vzpkg list` will display a list of application EZ templates available for this OS EZ template. Without any options, the utility lists all EZ templates installed on the server.

Name	Description
`-p, --package`	Lists the software packages installed in the container or included in the OS EZ template.
`-g, --groups`	Lists the YUM software groups installed in the container or available for the OS EZ template. The `-g` option works only for containers running RPM-based Linux distributions.
`-O, --os`	Displays the OS EZ template the container is based on.

Continued on next page

Table 3.3.5.1 -- continued from previous page

Name	Description
-A, --app	Displays the application EZ templates installed in the container or included in the OS EZ template.
-C, --cache	Lists the packages included in the specified EZ template or applied to the specified container from the local `vzpkg` cache. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file. Should be used along with the -p option.
-r, --remote	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg list` list the packages included in the specified EZ template or applied to the specified container in the remote repositories. Should be used along with the -p option.
-u, --custom-pkg	Displays a list of packages that are applied to the specified container but absent from the repository set to handle the EZ template(s) where these packages are included.
-i, --pkgid	Displays the ID assigned to the EZ template instead of its name; these IDs are unique within the given system. If the `<CT_name>` argument is given, the command shows the IDs of the EZ templates available inside the container. If the `<OS_template>` argument is given, the command displays the IDs of the OS EZ template specified and all its EZ application templates.
-S, --with-summary	In addition to listing the EZ templates available either in the container (if the `<CT_name>` argument is given) or installed on the server (if the `<CT_name>` argument is omitted), this option makes `vzpkg list` display the summary information on the corresponding EZ templates/packages.
-c, --cached	This option has no effect if the `<CT_name>` argument is given. If used for listing the EZ templates available on the server, it makes `vzpkg list` omit all application and OS EZ templates for which the cache has not been created (by running the `vzpkg create cache` command). In other words, with this option on, `vzpkg list` will list only the OS EZ templates ready to be used for the container creation.
appcache	Outputs a list of OS EZ template caches with preinstalled applications.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

3.3.6 vzpkg info

This command displays information about EZ templates, YUM software packages, and individual software packages.

```
vzpkg info [-F {<OS_template>|<CT_name>} -q|-d <app_template> [<parameters> ...]
vzpkg info -p|-g [-C|-r] [-F {<OS_template>|<CT_name>} -q|-d {<package_name>|<yum_package_group>} [-
```

Name	Description
`<CT_name>`	Container name.
`<OS_template>`	OS EZ template.
`<app_template>`	Application EZ template.
`<package_name>`	Software package name.
`<yum_package_group>`	YUM software group name.
`-F, --for-os` `{<OS_template>|<CT_name>}`	Displays information on the application EZ template or the software package (if the `-p` option is specified) included in the specified OS EZ template or applied to the indicated container.
`-p, --package`	Displays information about the specified software package. Must be used with the `-F` option.
`-g, --groups`	Displays information about the packages included in the specified YUM software group.
`-C, --cache`	Displays the information on the specified package from the local `vzpkg` cache. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
`-r, --remote`	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg info` get the information on the specified package from the remote repositories set for handling the EZ template where this package is included.
`-d, --debug <num>`	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
`-q, --quiet`	Disables logging to the screen and to the log file.

While executed, `vzpkg info` parses the subdirectories and files located in the
`/vz/template/<os_name>/<os_version>/<arch>/config` directory and containing the EZ template meta data. To
run the command, you should specify either the OS EZ template or the container name. In either case,
detailed information on the corresponding OS EZ template is displayed. You can also use the `-F` option to get
the necessary information on any application EZ template included into the OS EZ template or applied to the
container.

By default, `vzpkg info` displays all meta data on the EZ template/package specified. However, you can reduce
the amount of the output information by using special parameters (`<parameters>`) listed in the table below.

Output Parameters

Name	Description
name	The name of the EZ template/package.
packages	The packages included in the EZ template. For EZ templates only.
repositories	The repository where the packages comprising the EZ template are stored. For EZ templates only.
mirrorlist	The URL to the file containing a list of repositories from where the packages comprising the EZ template are to be downloaded. For EZ templates only.
distribution	The Linux distribution on the basis the OS EZ template has been created or under which the application EZ template is to be run. For EZ templates only.
summary	Brief information on the EZ template/package.
description	Detailed information on the EZ template/package. As distinct from `summary`, it can contain additional data on the EZ template/package.
technologies	For EZ templates only. Displays the following information: • The microprocessor architecture where the EZ template is to be used (`x86`, `x86_64`); • Specifies whether the EZ template can be used only on the servers with the Native POSIX Thread Library (NPTL) support. In this case the `nptl` entry is displayed after the `vzpkg info` execution.
version	The version of the software package.
release	The release of the software package.

Table 3.3.6.2 -- continued from previous page

Name	Description
`arch`	The system architecture where the EZ template/package is to be used. It can be one of the following: • `x86` if the EZ template/package is to be used on 32-bit platforms. • `x86_64` if the EZ template is to be used on 64-bit platforms.
`config_path`	Displays the path to the EZ template configuration directory containing the template meta data where the meta data for the base OS EZ template are stored (the default directory path is `/vz/template/<OS_name>/<OS_version>/<arch>/config/os/default`).
`package_manager_type`	For EZ templates only. The packaging system used to handle the packages included in the specified EZ template. It can be one of the following: • `rpm` for RPM-based Linux distributions (Fedora Core, Red Hat Enterprise Linux, etc.); • `dpkg` for Debian-based Linux distributions (e.g., Debian and Ubuntu).

Continued on next page

Table 3.3.6.2 -- continued from previous page

Name	Description
package_manager	The package manager type for managing the packages included in the specified EZ template. It can be one of the following: • `rpm49db5x86`: Fedora 17 • `rpm49x86`: Fedora 15 and 16 • `rpm47x86`: Red Hat Enterprise Linux 6 and CentOS 6 • `rpm44x86`: Red Hat Enterprise Linux 5 and CentOS 5 • `rpm43x86`: Red Hat Enterprise Linux 3 and 4, CentOS 3 and 4 • `rpmzypp44x86`: SUSE Linux Enterprise Server 11 with Service Pack 2 • `rpm41x86`: SUSE Linux Enterprise Server 10 and SUSE Linux 10.x • `rpm41s9x86`: SUSE Linux Enterprise Server 9 • `rpmzypp49x86`: openSUSE 12.1 • `dpkg`: Debian and Ubuntu • `rpm49db5x64`: Fedora 17 • `rpm49x64`: Fedora 15 and 16 • `rpm47x64`: Red Hat Enterprise Linux 6 and CentOS 6 • `rpm44x64`: Red Hat Enterprise Linux 5 and CentOS 5 • `rpm43x64`: Red Hat Enterprise Linux 3 and 4, CentOS 3 and 4 • `rpmzypp44x64`: SUSE Linux Enterprise Server 11 with Service Pack 2 • `rpm41x64`: SUSE Linux Enterprise Server 10 and SUSE Linux 10.x • `rpm41s9x64`: SUSE Linux Enterprise Server 9 • `rpmzypp49x64`: openSUSE 12.1 • `dpkgx64`: Debian and Ubuntu

3.3.7 vzpkg status

This command is used to check the status of the packages either installed inside a container or included in an OS EZ template.

```
vzpkg status [<options>] {<CT_name>|<OS_template>}
```

Name	Description
`-C, --cache`	Makes the `vzpkg status` command look for available updates in the local `vzpkg` cache only. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
`-r, --remote`	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg status` look for the package updates in the remote repositories set for handling the corresponding EZ template.
`-d, --debug <num>`	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
`-q, --quiet`	Disables logging to the screen and to the log file.

When executed, the command performs the following operations:

- Checks all the packages installed inside the specified container or included in the specified OS EZ template.

- Checks the repository used to install/update packages inside the container/OS EZ template.

- Compares the packages in the repository with those inside the container/OS EZ template.

- Lists the found packages updates for the container/OS EZ template, if any, or informs you that the container/OS EZ template is up-to-date.

Note: The `vzpkg status` command can be executed for running containers only.

3.3.8 vzpkg install

This command is used to install application EZ templates, YUM software groups, or individual software packages into containers.

```
vzpkg install [<options>] <CT_name> <object> [...]
```

The `vzpkg install` command will add an `<object>` to the specified container. An object can be an application EZ template, a YUM software group, or a standard software package. You can specify several objects to install into the container by separating them by spaces.

When executed, `vzpkg install` automatically handles the interdependencies among the packages to be installed into a container and ensures that all dependencies are satisfied. If the package dependencies cannot be resolved, the installation process fails and the corresponding message is displayed.

Name	Description
`-p, --package`	Installs a software package instead of an EZ template.
`-g, --groups`	Installs a YUM software group instead of an EZ template. The `-g` option works only for containers running RPM-based Linux distributions.
`-f, --force`	Forces the EZ template/package installation.
`-C, --cache`	Makes the `vzpkg install` command look for the packages included in the EZ template in the local `vzpkg` cache only. If there is a package not available locally, the command will fail. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
`-r, --remote`	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg install` look for the packages in the remote repositories set for handling the corresponding EZ template.
`-n, --check-only`	Simulates the same operations as `vzpkg install` completes without specifying this option (downloads the software packages to the server, handles the package interdependencies, etc.); however, the packages themselves are not installed in the specified the container.
`-d, --debug <num>`	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
`-q, --quiet`	Disables logging to the screen and to the log file.

By default, the specified object is treated by `vzpkg install` as an application EZ template. However, you can use the `-p` or `-g` option to explicitly specify the type of the object.

Note: A container has to be running in order to apply an application EZ template to or install a package inside this container.

3.3.9 vzpkg update

The `vzpkg update` command is used to update the following components of a container:

- OS EZ template

- application EZ templates

- YUM software groups

- individual software packages

```
vzpkg update [<options>] <CT_name> [<object> [...]]
```

Name	Description
`-C, --cache`	Makes the `vzpkg update` command look for the package updates in the local `vzpkg` cache only. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
`-r, --remote`	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg update` look for the package updates in the remote repositories set for handling the corresponding EZ templates.
`-p, --package`	Updates the packages installed in the container by using the `vzpkg install` command.
`-g, --groups`	Updates the YUM software group in the container. The `-g` option works only for containers running RPM-based Linux distributions.
`-f, --force`	Forces the EZ template/package update procedure.
`-n, --check-only`	Simulates the same operations as `vzpkg update` completes without specifying this option (downloads the updated packages to the server, handles their interdependencies, etc.); however, the packages themselves are not updated.
`-d, --debug <num>`	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
`-q, --quiet`	Disables logging to the screen and to the log file.

Without any options specified, `vzpkg update` updates all EZ templates (including the OS EZ template) in the specified container. However, you can make the command update a particular EZ template by specifying its name as `<object>`. You can also use the `-p` or `-g` option to update YUM software groups or individual

software packages in the container.

3.3.10 vzpkg remove

This command is used to remove an application EZ template, YUM software group, or a software package from a container.

```
vzpkg remove [<options>] <CT_name> <object> [...]
```

This command will remove `<object>` from the container with the name `<CT_name>`. The `<object>` can be an application EZ template, a YUM software group, or a software package installed with the `vzpkg install` command. You may specify a number of objects for removing.

Name	Description
`-p, --package`	Removes the specified package(s) from the container.
`-g, --groups`	Removes the specified YUM software group from the container. The `-g` option works only for containers running RPM-based Linux distributions.
`-w,` `--with-depends`	Removes also the packages having dependencies with the object specified.
`-f, --force`	Forces the EZ template/package deletion.
`-n, --check-only`	Simulates the same operations as `vzpkg remove` completes without specifying this option (handles interdependencies of the packages to be removed from the server, etc.); however, the packages themselves are not deleted from the specified container(s).
`-d, --debug <num>`	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
`-q, --quiet`	Disables logging to the screen and to the log file.

By default, the specified object is treated by `vzpkg remove` as an application EZ template. However, you can use the `-p` or `-g` option to explicitly specify the type of the object.

Note: A container has to be running in order to remove an application EZ template/package from it.

3.3.11 vzpkg create cache

This command is used to create tarballs (caches) for OS EZ templates. You should execute this command before you start using a newly installed OS EZ template for creating containers.

```
vzpkg create cache [<options>] [<OS_template> [...]]
```

Name	Description
-C, --cache	Makes the `vzpkg create cache` command check for the packages included in the EZ OS template in the local `vzpkg` cache only and use them for the cache creation. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file. In this case `vzpkg create cache` will also check the local `vzpkg` cache only.
-r, --remote	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg create cache` check for the packages included in the EZ OS template in the remote repositories set for its handling.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.
-f, --force	Forces the process of the cache creation.

`vzpkg create cache` checks the template area on the server (by default, the `/vz/template` directory is used) and if it finds an OS EZ template for which no tar archive exists, it creates a gzipped tarball for the corresponding OS EZ template and places it to the `/vz/template/cache` directory. When a container is being created, `prlctl` just unpacks the tar archive.

By default, `vzpkg create cache` checks the tar archive existence for all OS EZ templates installed on the server and creates some, if necessary. However, you can explicitly indicate what OS EZ template should be cached by specifying its name as `<OS_template>`. If the cache of the OS template specified already exists on the server, the command will fail and you will be presented with the corresponding error message.

3.3.12 vzpkg update cache

This command is used to update tarballs (caches) of the OS EZ templates installed on the server.

```
vzpkg update cache [<options>] [<OS_template> [...]]
```

Name	Description
-C, --cache	Makes the `vzpkg update cache` command check for the packages updates in the local `vzpkg` cache only and use them for the cache creation. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file. In this case `vzpkg update cache` will also check the local `vzpkg` cache only.
-r, --remote	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg update cache` check for the packages updates in the remote repositories set for handling the given EZ OS template.

The `vzpkg update cache` command checks the cache directory in the template area (by default, the template area is located in the `/vz/template` directory on the server) and updates all existing tarballs in this directory. However, you can explicitly indicate what OS EZ template tarball is to be updated by specifying its name as `<OS_template>`. Upon the `vzpkg update cache` execution, the old tarball is renamed by receiving the `-old` suffix (e.g., `centos-6-x86_64.tar.gz-old`).

If the `vzpkg update cache` command does not find a tarball for one or more OS EZ templates installed on the server, it creates the corresponding tar archive(s) and puts them to the `/vz/template/cache` directory.

3.3.13 vzpkg remove cache

This command removes the cache for the OS EZ templates specified.

```
vzpkg remove cache [<options>] [<OS_template> [...]]
```

Name	Description
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

By default, `vzpkg remove cache` deletes all caches located in the `/vz/template/cache` directory on the server. However, you can explicitly indicate what OS EZ template tar archive is to be removed by specifying its name as `<OS_template>`.

Note: The OS EZ template caches having the `-old` suffix are not removed from the `/vz/template/cache` directory. You should use the `-rm` command to delete these caches from the server.

3.3.14 vzpkg create appcache

This command combines an OS EZ template cache and one or more application EZ templates into a new OS and applications cache. If the OS EZ template cache has not been created yet, this will be done before application templates are added.

```
vzpkg create appcache --config <config> [<options>]
```

Name	Description
`--config <config>`	Specifies the path to the configuration file with the information on what OS and application templates to use.
`--ostemplate <OS_template>`	Specifies the OS EZ template to use in cache creation. This option redefines the OS EZ template specified in the configuration file.
`--apptemplate <app_template>`	Specifies one or more application EZ templates (comma-separated) to be added to the resulting cache.This option redefines the application EZ templates specified in the configuration file.
`-d, --debug <num>`	Sets the debugging level (0 to 10), 10 being the highest.
`-q, --quiet`	Disables logging to screen and log file.
`-f, --force`	Forces cache creation.

3.3.15 vzpkg update appcache

This command updates an existing OS EZ template cache with preinstalled application templates if the `--update-cache` option is provided. Otherwise, the cache is created anew.

```
vzpkg update appcache --config <config> [<options>]
```

Name	Description
`--config <config>`	Specifies the configuration file with the information on what OS and application templates to use.
`--ostemplate <OS_template>`	Specifies the OS EZ template, cache of which needs to be recreated or updated. This option redefines the OS EZ template specified in the configuration file.
`--apptemplate <app_template>`	Specifies all application EZ templates (comma-separated) preinstalled in the cache which needs to be updated. This option redefines the application EZ templates specified in the configuration file.
`--update-cache`	Instructs the command to check for updates for the existing OS and application cache. Otherwise, the cache is created anew.
`-d, --debug <num>`	Sets the debugging level (0 to 10), 10 being the highest.
`-q, --quiet`	Disables logging to screen and log file.
`-f, --force`	Forces cache creation.

3.3.16 vzpkg remove appcache

This command removes an existing OS EZ template cache with preinstalled application templates.

```
vzpkg remove appcache --config <config> [<options>]
```

Name	Description
`--config <config>`	Specifies the configuration file with the information on what OS and application templates to use.
`--ostemplate <OS_template>`	Specifies the OS EZ template, cache of which needs to be removed. This option redefines the OS EZ template specified in the configuration file.
`--apptemplate <app_template>`	Specifies all application EZ templates (comma-separated) preinstalled in the cache which needs to be removed. This option redefines the application EZ templates specified in the configuration file.
`-d, --debug <num>`	Sets the debugging level (0 to 10), 10 being the highest.
`-q, --quiet`	Disables logging to screen and log file.
`-f, --force`	Forces cache creation.

3.3.17 vzpkg localinstall

The `vzpkg localinstall` command is used to install a software package inside a container from the corresponding file on the server.

```
vzpkg localinstall [<options>] <CT_name> <rpm_file_path> [...]
```

Name	Description
-C, --cache	When handling the package interdependencies, makes the `vzpkg localinstall` command look for the needed packages in the local `vzpkg` cache only. If there is a package not available locally, the command will fail. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
-r, --remote	If the elapsed time from the last `vzpkg` local cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg localinstall` look for the packages in the remote repository.
-n, --check-only	Simulates the same operations as `vzpkg localinstall` completes without specifying this option (e.g., handles the package interdependencies); however, the package itself is not installed in the specified container.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

When executed, the command installs the package, the full path to which is specified as <rpm_file_path>, inside the container with the name <CT_name>. You may specify multiple packages to be installed inside the container.

During its execution, `vzpkg localinstall` automatically handles the interdependencies among the packages to be installed inside a container and ensures that all dependencies are satisfied. If the package dependencies cannot be resolved, the installation process will fail and the corresponding message will be displayed.

3.3.18 vzpkg localupdate

The `vzpkg localupdate` command is used to update the software packages installed inside your container(s) by means of the `vzpkg install` or `vzpkg localinstall` commands.

```
vzpkg localupdate [<options>] <CT_name> <rpm_file_path> [...]
```

Name	Description
-C, --cache	When handling the package interdependencies, makes the `vzpkg localupdate` command look for the needed packages in the local `vzpkg` cache only. If there is a package not available locally, the command will fail. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
-r, --remote	If the elapsed time from the last `vzpkg` local cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg localupdate` look for the packages in the remote repository.
-n, --check-only	Simulates the same operations as `vzpkg localupdate` completes without specifying this option (e.g., handles the package interdependencies); however, the package itself is not installed in the specified container.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

When executed, `vzpkg localupdate` compares the file on the server the full path to which is specified as `<rpm_file_path>` with the corresponding package inside the container with the name `<CT_name>` and updates it, if necessary. You may specify a number of packages at once to be updated inside your container.

3.3.19 vzpkg upgrade

The `vzpkg upgrade` command is used to upgrade an OS EZ template the container is based on to a newer version.

```
vzpkg upgrade [<options>] <CT_name>
```

Name	Description
`-C, --cache`	Makes the `vzpkg upgrade` command check for the packages included in the OS EZ template in the local `vzpkg` cache only. If any package is not available locally, the command will fail. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file; in this case `vzpkg upgrade` will also check the local `vzpkg` cache only.
`-r, --remote`	If the elapsed time from the last local `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg upgrade` check for the packages in the remote repositories set for handling the given EZ OS template.
`-n, --check-only`	Simulates the same operations as `vzpkg upgrade` completes without specifying this option (downloads the packages to the server, handles their interdependencies, etc.); however, the packages themselves inside the container are not upgraded.
`-f, --force`	Forces the upgrade of the OS EZ template.
`-d, --debug <num>`	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
`-q, --quiet`	Disables logging to the screen and to the log file.

3.3.20 vzpkg fetch

This command is used to download packages included in the corresponding OS EZ template or their updates from the remote repository to the `vzpkg` local cache on the server and to prepare them for installation.

```
vzpkg fetch [<options>] <OS_template>
```

Name	Description
`-O, --os`	Download packages/updates for the specified EZ OS template.
`-A, --app`	Download packages/updates for EZ application templates used with the EZ specified OS template.
`-C, --cache`	Makes the `vzpkg fetch` command look for the metadata in the `vzpkg` local cache only. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.

Table 3.3.20.1 -- continued from previous page

Name	Description
-r, --remote	If the elapsed time from the last vzpkg cache update does not exceed the value of the METADATA_EXPIRE parameter specified in the /etc/vztt/vztt.conf file, you should use this option to make vzpkg fetch look for the OS EZ template metadata in the remote repositories set for handling the corresponding EZ template.
-f, --force	Forces the process of downloading packages and/or their updates to the server.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

You can make vzpkg fetch run as a cron job (e.g., nightly) checking for available packages or packages updates for your EZ templates and keeping them in the local cache. Having all the necessary packages in the vzpkg local cache can greatly speed up the execution of the vzpkg install, vzpkg update, or vzpkg create cache commands since the packages are available locally and there is no need to check for them in the corresponding remote repositories.

3.3.21 vzpkg clean

This command is used to remove the software packages, their headers, and metadata downloaded to the server from the repository during the vzpkg execution (e.g., while caching an OS EZ template or adding an application EZ template to a container for the first time).

```
vzpkg clean [<options>] [<OS_template> [...]]
```

Name	Description
-k, --clean-packages	Removes the packages, headers, and metadata of the specified EZ OS template from the local vzpkg cache. This is also the default behavior of vzpkg clean.
-t, --clean-template	Checks the template area for the specified EZ OS template (the template area has the default path of /vz/template) and removes all packages that are currently not used by any container on the server and not included in the EZ OS template cache.

Table 3.3.21.1 -- continued from previous page

Name	Description
-a, --clean-all	Removes both: • the packages, headers, and metadata of the specified EZ OS template from the `vzpkg` local cache, and • the packages that are currently not used by any container on the server and not included in the EZ OS template cache.
-f, --force	Forces the `vzpkg clean` execution.
-n, --check-only	Simulates the same operations as `vzpkg clean` completes without specifying this option; however, the packages and headers are not removed from the server.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

3.3.22 vzpkg update metadata

This command is used to update the OS EZ template local metadata on the server.

```
vzpkg update metadata [<options>] [<OS_template> [...]]
```

Name	Description
-C, --cache	Makes the `vzpkg update metadata` command look for available metadata updates in the local `vzpkg` cache only. You can omit this parameter if the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file.
-r, --remote	If the elapsed time from the last `vzpkg` cache update does not exceed the value of the `METADATA_EXPIRE` parameter specified in the `/etc/vztt/vztt.conf` file, you should use this option to make `vzpkg update metadata` look for the updated metadata in the remote repositories set for handling the corresponding OS EZ template.
-d, --debug <num>	Sets the debugging level to one of the specified values (from 0 to 10). 10 is the highest debug level and 0 sets the debug level to its minimal value.
-q, --quiet	Disables logging to the screen and to the log file.

When executed without any options, the command updates the metadata of all OS EZ templates installed on the server. If you specify one or more OS EZ templates, the command will update the metadata of the indicated OS templates only. You can run this command a cron job at regular intervals to be sure that your OS EZ templates metadata are always up-to-date.

3.4 Supplementary Tools

3.4.1 pcompact

Utility to compact containers by removing unused blocks from their virtual disks. By compacting virtual disks, you can increase free disk space on the physical server.

```
pcompact [-v] [-n] [-s] [-t <timeout>[s|m|h]
```

Name	Description
-v	Increase the command output verbosity. Multiple -v options can be specified to produce a more verbose output.
-n	Display the actions the command will execute but do not actually compact the disks.
-s	Stop the command execution after compacting the first virtual disk.
-t <timeout>[s|m|h]	Terminate the command after the specified timeout, in seconds (default), minutes or hours.

3.4.2 pfcache

Memory and IOPS deduplication management utility that enables/disables caching for container directories and files, verifies cache integrity, checks containers for cache errors, and purges the cache if needed.

Note: The utility does not support additional disks attached to container.

```
pfcache <command> {<file>|<dir> [<options>]
```

Subcommands:

Name	Description
mark	Enables caching of the specified files, directories or subdirectories in containers.
unmark	Disables caching of the specified files, directories or subdirectories in containers.
purge	Frees up space in the memory and IOPS deduplication cache image.
verify	Verifies the integrity of the specified mounted memory and IOPS deduplication cache and deletes corrupted files.
check	Checks for and fixes memory and IOPS deduplication cache errors in the specified container root directory.
stat	Displays inodes summary for a running container.
dump	In addition to the summary provided by stat, shows detailed information on PFCache inodes of a running container. If no options are specified, outputs full information on checksummed peer inodes.

3.4.2.1 pfcache check

Checks for and fixes memory and IOPS deduplication cache errors in the specified container root directory.

```
pfcache check <dir> [--dry-run]
```

Name	Description
<dir>	Container root directory.
--dry-run	Report errors but do not make changes to the file system.

3.4.2.2 pfcache dump

In addition to the summary provided by stat, shows detailed information on PFCache inodes of a running container. If no options are specified, outputs full information on checksummed peer inodes.

```
pfcache dump <dir> [--csummed{brvbar} -a, --all {brvbar} <csum>] [--column <col> [...]]
```

Name	Description
<dir>	Container root directory.
--csummed	Outputs information on checksummed inodes.
-a, --all	Outputs information on all inodes.
<csum>	Outputs information on inodes with the specified checksum.

Continued on next page

Table 3.4.2.2.1 -- continued from previous page

Name	Description
`--column <col> [...]`	Displays only the chosen column(s). `<col>` can be: • `a` - All • `h` - File handle • `p` - Page cache size, pages • `c` - Checksum • `f` - Filter state • `s` - File size, bytes

3.4.2.3 pfcache mark, unmark

Enables or disables caching of the specified files, directories or subdirectories in containers.

```
pfcache mark <file>
pfcache mark <dir> [<subdir> [...]] [--recursive]
pfcache unmark <file>
pfcache unmark <dir> [<subdir> [...]] [--recursive]
```

Name	Description
`<file>`	File to enable/disable caching of.
`<dir>`	Directory to enable/disable caching of.
`<subdir>`	Subdirectory to enable/disable caching of.
`--recursive`	Process directory's or subdirectory's current contents.

3.4.2.4 pfcache purge

Frees up space in the memory and IOPS deduplication cache image. If no options are specified, purges entire cache.

```
pfcache purge <cache_dir> [--unused | --size <size> | --expire <date>]
```

Name	Description
`<cache_dir>`	Memory and IOPS deduplication cache image location.
`--unused`	Remove only files unused at the moment.

Continued on next page

93

Name	Description
`--size <size>`	Attempt to free `size` bytes in the memory and IOPS deduplication cache image.
`--expire <date>`	Remove files accessed before the specified date. A date can be specified in the ISO 8601 format or in the format defined in your system locale, with optional hours, minutes, and seconds. Examples: • `05/21/12 [09:50[:33]]` • `2012-05-21 [09:50[:33]]`

3.4.2.5 pfcache stat

Displays a summary of all files in the specified file system which have been accessed recently or are being accessed now.

```
pfcache stat <dir>
```

Name	Description
`<dir>`	Container root directory.

Displayed Information

Name	Description
`csums`	The number of checksummed files and their percentage in the total number of files fetched by the command. **Note:** Only checksummed files can be cached.
`inodes`	The number of files which have been accessed recently or are being accessed now.
`size`	The size of the files, in kilobytes.
`RAM`	Memory used by the files, in kilobytes.
`fetched`	The number of files fetched by the command.
`uncached`	The number of files only in container's private area.

Name	Description
cached	The number of files which have copies in the PFCache area and their percentage in **fetched**.

3.4.2.6 pfcache verify

Verifies the integrity of the specified mounted memory and IOPS deduplication cache and deletes corrupt files.

```
pfcache verify <cache_dir>
```

Name	Description
<cache_dir>	Memory and IOPS deduplication cache image location.

3.4.2.7 Resizing pfcache

The default pfcache size limit is 10 gigabytes (GB). If the required pfcache size exceeds the default limit on the production host, the administrator must increase it. There is no single recommended pfcache size, as it depends on the workloads in containers. Thus, it is only required to expand the pfcache size when there is no space.

To add more space to pfcache:

1. Check space availability on the /vz partition:

   ```
   # df -h /vz
   ```

2. Resize the pfcache ploop device to a greater value by specifying its new full size:

   ```
   # ploop resize -s 50G /vz/pfcache.hdd/DiskDescriptor.xml
   ```

Note: 50G is an example. Specify smaller or greater values depending on the current pfcache size and the amount of space you want to add.

3.4.3 prl_disk_tool

The `prl_disk_tool` utility is used to manage virtual hard disk drives.

3.4.3.1 prl_disk_tool compact

Removes all empty blocks from the expanding virtual disk to reduce its size on the physical hard disk. The virtual disk must be formatted to NTFS, ext2/ext3/ext4, btrfs, or xfs.

```
prl_disk_tool compact --hdd <disk_path> [--force]
prl_disk_tool compact -i, --info --hdd <disk_path>
```

Name	Description
--hdd <disk_path>	Full path to the virtual disk.
--force	Forces the compacting operation for suspended virtual disks.
-i, --info	Do not compact the virtual disk; just display the information about the size the disk will have after compacting.

3.4.3.2 prl_disk_tool merge

Merges all snapshots of the virtual hard disk.

```
prl_disk_tool merge --hdd <disk_path>
```

Name	Description
--hdd <disk_path>	Full path to the virtual disk.

3.4.3.3 prl_disk_tool resize

Changes the capacity of the specified virtual disk. During resizing, all data present on the disk volumes are left intact. You can also resize the last partition using the `--resize_partition` option. The supported file systems are NTFS, ext2/ext3/ext4, btrfs, or xfs.

```
prl_disk_tool resize --size <size>[K|M|G|T] [--resize_partition]
                     --hdd <disk_path> [--force]
prl_disk_tool resize -i, --info [--units <K|M|G|T>] --hdd <disk_path>
```

Name	Description
`--size`	The new size of the virtual disk. It can be set in kilobytes (`K`), megabytes (`M`, default), gigabytes (`G`), or terabytes (`T`).
`--resize_partition`	Resizes the last partition of the specified virtual disk. **Note:** You cannot reduce XFS file systems (the default choice for CentOS 7 and Red Hat Enterprise Linux 7).
`--hdd <disk_path>`	Full path to the virtual disk.
`--force`	Forces the resizing operation for suspended virtual disks.
`-i, --info`	Do not resize the virtual disk; just show the size the disk will have after resizing.
`--units`	Displays the disk size in kilobytes (`K`), megabytes (`M`, default), gigabytes (`G`), or terabytes (`T`).

3.4.4 vzpid

This utility prints the ID of the container where the process is running.

```
vzpid <pid> [...]
```

Multiple process IDs can be specified as arguments.

3.4.5 vzps, vztop

These two utilities can be run on the server just as the standard Linux `ps` and `htop` utilities. For information on the `ps` and `htop` utilities, consult their man pages. The `vzps` and `vztop` utilities provide certain additional functionality related to monitoring separate containers running on the server.

The `vzps` utility has the following functionality added: the `-E <CT_name>` command-line switch can be used to show only the processes running inside the container with the specified ID.

The `vztop` utility has the **CTID** column added to display the container UUID where a particular process is running (`0` stands for the server itself).

3.4.6 vzsplit

This utility is used to generate a sample container configuration file with a set of system resource control parameters.

```
vzsplit [-n <num>] [-f <sample_name>] [-s <swap_size>]
```

This utility is used for dividing the server into equal parts. It generates a full set of containers system resource control parameters based on the total physical memory of the server it runs on and the number of containers the server shall be able to run even if the given number of containers consume all allowed resources.

Without any option the utility prompts for the desired number of containers and outputs the resulting resource control parameters to the screen.

Name	Description
-n <num>	Desired number of containers to be simultaneously run on the server.
-f <sample_name>	Name of the sample configuration to create.
-s <swap_size>	Size of the swap file on the server. It is recommended to specify the swap size to be taken into account when the utility generates sample configurations.

The resulting sample configuration will be created in the `/etc/vz/conf` directory. The file name will be `ve-<sample_name>.conf-sample`. Now you can pass `<sample_name>` as an argument to the `--config` option of the `prlctl create` command. If a sample with this name already exists, the utility will output an error message and will not overwrite the existing configuration.

CHAPTER 4

Managing Virtual Machines

4.1 Virtual Machine Utilities

Virtuozzo virtual machines can be managed using the `prlctl` command-line utility. The utility is installed on the hardware node during the product installation.

4.1.1 prlctl

The `prlctl` utility is used to perform administration tasks on virtual machines. The utility supports a full range of tasks from creating and administering virtual machines to getting statistics and generating problem reports.

```
prlctl <command> <VM_name> [<options>] [-v, --verbose <number>] [--timeout <sec>]
       [-l, --login [<user>[:_<passwd>_]@]<server>] [-p, --read-passwd <file>]]
```

Name	Description
`<command>`	The name of the command to execute.
`<VM_name>`	The name of the virtual machine to perform the operation on. To obtain the list of the available virtual machines, use the `prlctl list` command.
`<options>`	Command options. See individual commands for available options.
`-v, --verbose <number>`	Enables verbose output. The greater the `<number>`, the higher the verbosity.
`--timeout <sec>`	Use a custom operation timeout specified in seconds. The default timeout for all operations is unlimited.

Continued on next page

99

Name	Description
-l, --login [<user>[:<passwd>]@]<server>	Connect to a remote `<server>` with the specified credentials. If this flag is omitted, the `prlctl` command is assumed to be run locally.
-p, --read-passwd <file>	Use the password from the file `<file>` to log in to a remote hardware node, other credentials being specified with the `--login` option. The `--read-passwd` option can be specified multiple times in order to form a password stack for operations requiring multiple passwords. Each password must be supplied in a separate file.

To display help, enter `prlctl` without any options.

4.1.2 prlctl backup, backup-list, backup-delete, restore

Creates, lists, deletes or restores virtual machine backups.

```
prlctl backup <VM_name|VM_UUID> [-f, --full] [-i, --incremental]
            [-s, --storage [<user>[:<passwd>]@]<server>[:<port>]]
            [--description <desc>] [--no-compression] [--no-tunnel]
            [--no-reversed-delta]
prlctl backup-list [<VM_name|VM_UUID>] [-f, --full] [--localvms] [--vmtype vm]
                [-s, --storage [<user>[:<passwd>]@]<server>[:<port>]]
prlctl backup-delete {<VM_name|VM_UUID> | -t, --tag <backup_ID>} [--keep-chain]
prlctl restore {<VM_UUID> | -t, --tag <backup_ID>} [-n, --name <new_name>]
            [--dst=<path>] [-s, --storage [<user>[:<passwd>]@]<server>[:<port>]]
            [--no-tunnel] [--live]
```

Name	Description
`<VM_name|VM_UUID>`	Virtual machine name or UUID. • Use with `prlctl backup` to create a backup of the specified virtual machine. • Use with `prlctl backup-list` to list backups of the specified virtual machine. • Use with `prlctl backup-delete` to delete all backups of the specified virtual machine. • Use <VM_UUID> with `prlctl restore` to restore the most recent backup of the specified virtual machine.
`-s, --storage` `[<user>[:<passwd>]@]<server>[:<port>]`	Specifies a remote backup server address, port, and credentials. If this option is omitted, the backup will be saved on the default backup server that can be configured using the `prlsrvctl set` command.
`--description <desc>`	Backup description. Descriptions with white spaces must be enclosed in quotation marks.
`-f, --full`	• Use with `prlctl backup` to create a full backup of the virtual machine. A full backup contains all virtual machine data. • Use with `prlctl backup-list` to display full backup information.
`-i, --incremental`	Create an incremental backup of the virtual machine. An incremental backup contains only the files changed since the previous full or incremental backup. This is the default backup type.
`--localvms`	List local backups only.
`--vmtype vm`	List virtual machine backups on the server.
`-t, --tag <backup_ID>`	The ID of the backup to restore or delete.
`-n, --name <new_name>`	A new name to assign to the restored virtual machine. If omitted, the virtual machine will be restored with the original name.

Continued on next page

Table 4.1.2.1 -- continued from previous page

Name	Description
`--dst=<path>`	Restore the virtual machine to the specified directory on the hardware node. If this option is omitted, the virtual machine will be restored to `/vz/vmprivate/<VM_UUID>`.
`--no-compression`	Do not compress the created backup image.
`--no-tunnel`	Disables connection tunneling for backup. Connection tunneling provides secure data transmission. To use this option, configure the firewall of the destination server to allow incoming connections on any port on the corresponding network interface.
`--no-reversed-delta`	Disables creating an intermediate temporary image for reversed delta writes during backup.
`--keep-chain`	Preserve the rest of the backup chain when deleting specific backups.
`--live`	Starts the restored virtual machine right after launching the restore process to reduce the VM downtime.

4.1.3 prlctl capture

Captures a screenshot of a current virtual machine console in the Portable Network Graphics (PNG) format.

```
prlctl capture <VM_name> [--file <path>]
```

Name	Description
`<VM_name>`	Name of the virtual machine to take a screenshot of.
`--file <path>`	If the `--file` option is specified, a screenshot is stored to the path <path>; otherwise, it is dumped to stdout.

4.1.4 prlctl clone

Creates an exact copy or a template of the specified virtual machine.

Note: Running VMs cannot be cloned. Templates of running VMs can still be created.

```
prlctl clone <VM_name> --name <new_name> [--template] [--dst=<path>] [--changesid]
            [--detach-external-hdd <yes|no>] [--linked]
```

Name	Description
`<VM_name>`	Name of the virtual machine to clone.
`--name <new_name>`	Name to be assigned to the new virtual machine.
`--template`	Create a virtual machine template instead of a real virtual machine. Templates are used as a basis for creating new virtual machines.
`--dst=<path>`	Path to the directory where the `<VM_UUID>` directory with cloned VM files will be stored. If this option is omitted, the clone is created in the default directory `/vz/vmprivate`.
`--changesid`	Generate a new Windows security identifier (SID) for a Windows-based virtual machine. For this parameter to work, Virtuozzo Hybrid Server tools must be installed in the virtual machine. **Note:** For Windows 2019 VMs, use Microsoft's Sysprep tool instead of this option.
`--detach-external-hdd <yes|no>`	If set to `no`, hard disks located outside the source virtual machine are not removed from the configuration of the resulting clone. Setting the parameter to `yes` removes external hard disks from the configuration. **Note:** External hard disks are not copied to the cloned virtual machine.

Name	Description
`--linked`	Create a linked clone of the virtual machine. Linked clones store only changes to virtual machine disks. They occupy less disk space but require access to the original disks. **Note:** Migration, backup, restore, and unlink operations are not supported for linked clones.

4.1.5 prlctl create

Creates a new virtual machine. A virtual machine can be created from scratch or from a virtual machine template. When created from scratch, the target operating system type or version must be specified. To create a virtual machine from a template, the template name must be passed to the command.

```
prlctl create <VM_name> [<options>]
```

Name	Description
`<VM_name>`	User-defined new virtual machine name. If the name consists of two or more words separated by spaces, it must be enclosed in quotes.
`-d, --distribution {<name>|list}`	The operating system distribution the virtual machine will be optimized for. For the full list of supported distributions, refer to the `prlctl` man pages.
`--ostemplate <template_name>`	The name of the virtual machine template from which to create the new virtual machine. Use the `prlctl list --template` command to obtain the list of the available templates.
`--dst <vm_path>`	Path to the directory where the `<VM_UUID>` directory with VM files will be stored. If this parameter is omitted, the virtual machine is placed in the default directory `/vz/vmprivate`.
`--uuid <uuid>`	A custom UUID to assign to the virtual machine.

Note: You can use either `--distribution` or `--ostemplate`, not both options at once.

When creating a virtual machine from scratch, you may specify the operating system family or version. If an operating system version is specified using the `--distribution` parameter, the virtual machine will be configured for that operating system. If an operating system family is specified using the `--ostype` parameter, the virtual machine will be configured for the default version of this OS family. The default versions are determined internally by Virtuozzo Hybrid Server. The best way to find out the default versions used in your Virtuozzo Hybrid Server installation is by creating a sample virtual machine.

4.1.6 prlctl delete

Deletes a virtual machine from the hardware node. The command removes a virtual machine from the Virtuozzo Hybrid Server registry and permanently deletes all its files from the server. Once completed, this operation cannot be reversed.

```
prlctl delete <VM_name>
```

Name	Description
<VM_name>	The name of the virtual machine to delete.

4.1.7 prlctl installtools

Mounts the Virtuozzo guest tools image to virtual machine's optical disk drive so the user can install Virtuozzo Hybrid Server tools.

```
prlctl installtools <VM_name>
```

Name	Description
<VM_name>	The name of the virtual machine.

4.1.8 prlctl update-qemu

Updates KVM/QEMU hypervisor in a running virtual machine from the `qemu-kvm-vz` package installed on the node. Make sure to update the `qemu-kvm-vz` package first.

```
prlctl update-qemu <VM_name>
```

Name	Description
`<VM_name>`	The name of the virtual machine.

4.1.9 prlctl enter

Creates a command prompt channel to a virtual machine. By using this command, you can create a command prompt channel and execute commands in a virtual machine. Virtuozzo Hybrid Server tools must be installed in a virtual machine to use this utility.

```
prlctl enter <VM_name>
```

Name	Description
`<VM_name>`	The name of the virtual machine.

4.1.10 prlctl exec

Executes a command inside a virtual machine. Virtuozzo Hybrid Server tools must be installed in a virtual machine to use this utility. By default, running `prlctl exec <command>` is equivalent to executing `bash -c <command>` in a Linux VM or `cmd /c <command>` in a Windows VM.

```
prlctl exec <VM_name> [--without-shell] <command>
```

Name	Description
`<VM_name>`	The name of the virtual machine.
`<command>`	A command to execute.
`--without-shell`	Run commands directly without `bash` or `cmd` shell.

4.1.11 prlctl list

Displays a list of virtual machines on the Hardware Node. Displays information on virtual machines on the Hardware Node.

```
prlctl list --vmtype vm [-a, --all] [-o, --output <field>[,...]]
            [-s, --sort {<field>|-<field>}] [-t, --template] [-j, --json]
prlctl list -i, --info --vmtype vm [<VM_name>] [-f, --full] [-t, --template]
                                  [-j, --json]
```

Name	Description
-a, --all	List all running, stopped, suspended, and paused virtual machines. If this and the rest of the parameters are omitted, only the running virtual machines will be displayed.
-t, --template	List available virtual machine templates instead of actual virtual machines.
-o, --output <field>[,....]	Display only the specified fields. Type field names in lower case. Separate multiple fields with commas. For the list of fields, see *prlctl list Output Parameters* on page 107.
-s, --sort {<field>|-<field>}	Sort virtual machines by the specified field in either ascending or descending order.
-i, --info	Display detailed information about the specified virtual machine.
-f, --full	Display detailed information about network cards in virtual machines. Used with the --info option.
<VM_name>	The name of the virtual machine for which to display the detailed information. If not specified, the information will be displayed for all registered virtual machines.
-j, -json	Produce machine-readable output in the JSON format.

4.1.11.1 prlctl list Output Parameters

Listed below are the parameters that can be specified after the -o switch.

Name	Output Column	Description
uuid	UUID	Virtual machine UUID.
hostname	HOSTNAME	Virtual machine hostname.

Table 4.1.11.1.1 -- continued from previous page

Name	Output Column	Description
name	NAME	Virtual machine name.
description	DESCRIPTION	Virtual machine description.
ostemplate	OSTEMPLATE	Specifies the name of the OS template the virtual machine is based on (e.g., centos-6-x86_64).
ip	IP_ADDR	Virtual machine IP address.
status	STATUS	Virtual machine status (e.g., running or stopped).
numproc	NPROC	The number of threads allowed.
mac	MAC	Network device's MAC address.
netif	NETIF	Network devices in the virtual machine .
iolimit	IOLIMIT	The bandwidth the virtual machine is allowed to use for its disk input and output (I/O) operations, in bytes per second.
ha_enable	HA_ENABLE	Indicates whether the virtual machine is joined to the High Availability Cluster.
ha_prio	HA_PRIO	Virtual machine priority in the High Availability Cluster (0 is the lowest). Higher-priority virtual environments are restarted first in case of failures.

4.1.12 prlctl migrate

Migrates a virtual machine from one server to another.

```
prlctl migrate <VM_name> <destination_server>[/<VM_name>]
              [--dst=<path>] [--clone|--remove-src] [--changesid] [--no-compression]
              [--no-tunnel] [--ssh <options>]
```

Name	Description
<VM_name>	The source virtual machine name.
<source_server>	The source server information. Use the following format to specify this info: [<user>[:<password>]@]<server_IP_address_or_hostname>[:<port>].
<destination_server>	The destination server information. If omitted, the migration will be performed locally. Use the following format to specify this info: [<user>[:<password>]@]<server_IP_address_or_hostname>[:<port>].

Table 4.1.12.1 -- continued from previous page

Name	Description
`--dst=<path>`	Path to the directory on the destination server where the `<VM_UUID>` directory with VM files will be stored.
`--clone`	Clone the original virtual machine to the destination server and leave it intact on the source server. The clone will have a different UUID, MAC address, SID (for Windows-based VMs only; if `--changesid` is specified), and offline management disabled. If this option is omitted, the original virtual machine will be removed from the source server after migration. Cannot be used together with `--remove-src`.
`--remove-src`	Remove the original virtual machine from the source server. Enabled by default. Cannot be used together with `--clone`.
`--changesid`	Changes the resulting virtual machine SID. **Note:** For Windows 2019 VMs, use Microsoft's Sysprep tool instead of this option.
`--no-compression`	Disable data compression during migration.
`--no-tunnel`	Disables connection tunneling for migration. Connection tunneling provides secure data transmission. The option works only for VM live migration. To use this option, configure the firewall of the destination server to allow incoming connections on any port on the corresponding network interface.
`--ssh`	Additional options to pass to `ssh` to connect to the destination server. All standard `ssh` options are supported. **Note:** Do not specify the destination server hostname or IP address as an `ssh` option.

4.1.13 prlctl mount, umount

Mounts or unmounts the hard disks of a virtual machine to the `/vz/root/<UUID>` directory on the hardware node.

```
prlctl mount <VM_name> [-o <ro|rw> | --info]
prlctl umount <VM_name>
```

Name	Description
`<VM_name>`	Virtual machine name.
`-o <ro|rw>`	Sets access rights: • `ro` - read-only, • `rw` - read-write.
`--info`	Show information about the mounted virtual disks.

4.1.14 prlctl move

Moves the directory with virtual machine files to a new location on the same server.

```
prlctl move <VM_name> --dst=<path>
```

Name	Description
`<VM_name>`	Virtual machine name.
`--dst=<path>`	New location of the `<VM_UUID>` directory with virtual machine files.

4.1.15 prlctl pause, suspend, resume

Pause, suspend, and resume a virtual machine.

```
prlctl pause <VM_name>
prlctl suspend <VM_name>
prlctl resume <VM_name>
```

Name	Description
`<VM_name>`	The name of the virtual machine to pause, suspend, or resume.

The pause command pauses a virtual machine. To continue the virtual machine operation, use the prlctl start command.

The suspend command suspends the virtual machine operation. When a running virtual machine is suspended, the state of the virtual machine processes is saved to a file on the host. After that, the machine is stopped. To resume the machine, use the resume command.

4.1.16 prlctl problem-report

Obtains a problem report for the specified virtual machine and either sends it to the Virtuozzo technical support team or displays it on the screen.

```
prlctl problem-report <VM_name> -d, --dump | -s, --send \
                      [--proxy [<user>[:<passwd>]@]<proxyhost>[:<port>] | --no-proxy]
```

Name	Description
`<VM_name>`	The name of the virtual machine for which to obtain the problem report. If the name consists of separate words, it must be enclosed in quotes.
`-d, --dump`	Collect technical data about a virtual machine and display it on the screen. You can also pipe the output to a file and then send it to the Virtuozzo technical support team to analyze your problem.
`-s, --send`	Send the generated problem report to the Virtuozzo technical support team.
`--proxy [<user>[:<passwd>]@<proxyhost>[:<port>]`	Use the specified information to send the generated report through a proxy server, if you use one to connect to the Internet.
`--no-proxy`	Do not use a proxy server to send the generated report. This is the default behavior, so you can omit this parameter.

4.1.17 prlctl register, unregister

The `register` command is used to register a virtual machine with Virtuozzo Hybrid Server.

The `unregister` command removes a virtual machine from the Virtuozzo Hybrid Server registry.

```
prlctl register <path> [--preserve-uuid | --uuid <UUID>]
prlctl unregister <VM_name>
```

Name	Description
`<path>`	An absolute path to the virtual machine directory.
`<VM_name>`	The name of the virtual machine to remove from the Virtuozzo Hybrid Server registry.
`--preserve-uuid`	Do not change the virtual machine UUID. If ommited, the UUID is regenerated.
`--uuid <UUID>`	Change the virtual machine UUID to the specified one. If ommited, the UUID is regenerated.

Use the `register` command when you have a virtual machine on the server that does not show up in the list of the virtual machines registered with the Virtuozzo Hybrid Server. This can be a machine that was previously removed from the registry or a machine that was copied from another location.

The `unregister` command removes a virtual machine from the Virtuozzo Hybrid Server registry, but does not delete the virtual machine files from the server. You can re-register such a machine with Virtuozzo Hybrid Server later using the `register` command.

4.1.18 prlctl reset-uptime

Resets a virtual machine uptime counter as well as count start date and time.

```
prlctl reset-uptime <VM_name>
```

Name	Description
`<VM_name>`	Virtual machine name. Names consisting of multiple words must be enclosed in quotes.

4.1.19 prlctl set

The `prlctl set` command is used to modify the configuration of a virtual machine and manage virtual machine devices. The following subsections provide technical information on how to use the command to perform these tasks.

4.1.19.1 Modifying Virtual Machine Configuration

The `prlctl set` command is used to modify the virtual machine configuration parameters.

```
prlctl set <VM_name> [--cpus <number>] [--cpu-sockets <number>] [--memsize <number>]
          [--videosize <number>] [--memguarantee <percentage>]
          [--mem-hotplug <on|off>] [--description <desc>]
          [--autostart <on|off|auto>] [--autostart-delay <number>]
          [--autostop <stop|suspend>] [--applyconfig <conf>] [--name <new_name>]
          [--vnc-mode <auto|manual|off> {--vnc-passwd <passwd> | --vnc-nopasswd}]
          [--vnc-port <port>] [--vnc-address <address>] [--cpu-hotplug <on|off>]
          [--cpuunits <units>] [--cpulimit {<percent>|<megahertz>}]
          [--ioprio <priority>] [--iolimit <limit>] [--iopslimit <limit>]
          [--cpumask {<N>[,N,N1-N2] | all}] [--nodemask {<N>[,N,N1-N2] | all}]
          [--offline-management <on|off>] [--offline-service <service_name>]
          [--userpasswd <user>:<passwd>] [--rate <rate>]
          [--ratebound <on|off>_] [--apply-iponly <yes|no>] [--efi-boot <on|off>]
          [--tools-autoupdate <on|off>] [--on-crash <pause|restart>[:no-report]]
          [--nested-virt <on|off>]
```

Name	Description
<VM_name>	Target virtual machine name.
--cpus <number>	Number of CPU cores per CPU socket available to the virtual machine. The overall number of CPU cores available to a virtual machine is calculated by multiplying the number of CPU sockets by the number of CPU cores per socket and can be no greater than the number of CPU cores on the physical server. The default value is 2.
--cpu-sockets <number>	Number of CPU sockets available to the virtual machine. The default value is 1.

Table 4.1.19.1.1 -- continued from previous page

Name	Description
--memsize <number>	The amount of memory (RAM) available to the virtual machine, in megabytes. You can use the following suffixes to specify measurement units: • G for gigabytes • M for megabytes • K for kilobytes • B for bytes
--videosize <number>	The amount of video memory available to the virtual machine graphics card. You can use the following suffixes to specify measurement units: • G for gigabytes • M for megabytes • K for kilobytes • B for bytes
--memguarantee <size>	Sets a percentage of virtual machine's RAM that said VM is guaranteed to have. By default, set to 80%.
--mem-hotplug <on|off>	Enables or disables memory (RAM) hotplug support in the virtual machine. This feature is disabled in the virtual machine by default. The guest operating system must support memory hotplug for this functionality to work.
--description <desc>	Sets virtual machine description. Descriptions with white spaces must be enclosed in quotation marks.

Continued on next page

Table 4.1.19.1.1 -- continued from previous page

Name	Description
`--autostart <on|off|auto>`	Sets the virtual machine startup options: • `on` - automatically start the virtual machine when the hardware node starts or the Virtuozzo Hybrid Server component responsible for managing virtual machines is enabled. • `off` (default) - do not automatically start the virtual machine when the hardware node starts or the Virtuozzo Hybrid Server component responsible for managing virtual machines is enabled. • `auto` - let the virtual machine assume the state it has been in before the hardware node reboot/shutdown or disabling the Virtuozzo Hybrid Server component responsible for managing virtual machines.
`--autostart-delay <number>`	Sets the time delay used during the virtual machine automatic startup.
`--autostop <stop|suspend>`	Sets the automatic shutdown mode for the specified virtual machine: • `stop` - the virtual machine is stopped when you shut down the hardware node or disable the Virtuozzo Hybrid Server component responsible for managing virtual machines. • `suspend` - the virtual machine is suspended when the hardware node is shut down or the Virtuozzo Hybrid Server component responsible for managing virtual machines is disabled.
`--applyconfig <conf>`	Applies the resource parameter values from the specified VM configuration file to the virtual machine. The following parameters are applied: • all memory-related parameters (both RAM and video) • all CPU-related parameters • IO and IOPS limits • disk size

Table 4.1.19.1.1 -- continued from previous page

Name	Description
`--name <new_name>`	Changes the virtual machine name. You can only change the names of stopped virtual machines.
`--vnc-mode <auto|manual|off>`	Enables or disables access to the virtual machine via the VNC protocol.
`--vnc-port <port>`	Sets the VNC port number.
`--vnc-passwd <passwd> |` `--vnc-nopasswd`	Sets the VNC password or specifies that no password is needed for VNC connections. Either of these options is mandatory for any VNC connection.
`--vnc-address <address>`	Sets the IP address to use for logging in to the virtual machine via VNC. It must be one of the IP addresses assigned to the hardware node. By default, you can use any of the IP addresses of the hardware node to log in to the virtual machine.
`--cpu-hotplug <on|off>`	Enables or disables CPU hotplug support in the virtual machine. This feature is disabled by default. The guest operating system must support CPU hotplug for this functionality to work.
`--cpuunits <units>`	Sets the CPU weight for the virtual machine. This is a positive integer number that defines how much CPU time the virtual machine can get as compared to the other virtual machines and containers running on the server. The larger the number, the more CPU time the virtual machine can receive. Possible values range from 8 to 500000. If this parameter is not set, the default value of 1000 is used.
`--cpulimit` `{<percent>|<megahertz>}`	CPU limit, in percent or megahertz (MHz) the virtual machine is not allowed to exceed. By default, the limit is set in percent. To set the limit in MHz, specify "m" after the value. **Note:** If the server has 2 processors, the total CPU time equals 200%.
`--ioprio <priority>`	Disk I/O priority level from 0 to 7. The default is 4.

Continued on next page

Table 4.1.19.1.1 -- continued from previous page

Name	Description
`--iolimit <limit>`	Disk I/O bandwidth limit. The default is 0 (no limit). By default the limit is set in megabytes per second. You can use the following letters following the number to specify units of measure: • G - gigabytes per second (e.g., 1G). • K - kilobytes per second (e.g., 10K). • B - bytes per second (e.g., 100B). The default I/O bandwidth limit for all newly created virtual machines is set to 0, which means that no limits are applied to them.
`--iopslimit <limit>`	Maximum number of disk input and output operations per second a virtual machine is allowed to perform. By default, any newly created container does not have the IOPS limit set and can perform so many disk I/O operations per second as necessary.
`--cpumask {<N>[,N,N1-N2]	all}`	An affinity mask indicating what CPU(s) the virtual machine processes should be run on. You can specify a list of CPUs identified by their index numbers separated by commas (0, 1, 2, 3, etc.) or a range (4-6). To make all CPUs available for the virtual machine processes specify `--cpumask all`.
`--nodemask {<N>[,N,N1-N2]	all}`	The NUMA node mask defining a NUMA node to bind the virtual machine to. Once you set the mask, the processes running in the virtual machine will be executed only on the CPUs that belong to the specified NUMA node. You can specify a list of NUMA nodes by their index numbers separated by commas and as a range (e.g., `0,1,2,3,4-6`). To make all NUMA nodes available for virtual machine's processes specify `--nodemask all`.
`--offline-management <on|off>`	Turns the offline management on or off.
`--offline-service <service_name>`	The name of the service to use for offline management.
`--userpasswd <user>:<passwd>`	Sets the password for the specified user in the virtual machine. If the user account does not exist, it will be created. Virtuozzo Hybrid Server tools must be installed in the VM for the command to work.
`--rate <rate>`	Sets the guaranteed outgoing traffic rate in Kbps for the virtual machine.

Table 4.1.19.1.1 -- continued from previous page

Name	Description
`--ratebound <on|off>`	Turns the network traffic rate limitation set by the `--rate` parameter (above) on or off. The default value is `off`.
`--apply-iponly <yes|no>`	If set to `yes`, the hostname, nameserver, and search domain settings from the virtual machine configuration file are ignored.
`--efi-boot <on|off>`	If set to `on`, the virtual machine will boot using the EFI firmware. If set to `off` (default), the virtual machine will boot using the BIOS firmware.
`--tools-autoupdate <on|off>`	Enables or disables automatic install and update of Virtuozzo Hybrid Server guest tools in VMs. If set to `on`, Virtuozzo Hybrid Server tools are installed and updated automatically via a weekly `cron` job (requires that the `vz-guest-tools-updater` package be installed on the node). **Important:** During the installation or update, Virtuozzo Hybrid Server guest tools image is forcibly mounted to VM's optical disk drive even if it is already in use. If set to `off`, Virtuozzo guest tools are not installed and updated automatically, so that you can do it manually at a convenient time.
`--on-crash <pause|restart>[:no-report]`	Specifies VM behavior after it crashes: pause or restart. The problem report is sent by default. To omit sending the problem report, add `:no-report`.
`--nested-virt <on|off>`	Enables nested Intel VT-x virtualization in VM. **Important:** This feature is experimental and tested only on Linux guests.

4.1.19.2 Managing Virtual Devices

The `prlctl set` command allows to add, modify, and delete virtual devices of virtual machines.

```
prlctl set <VM_name> --device-add <dev_type> <options>
prlctl set <VM_name> --device-set <dev_name> <options>
prlctl set <VM_name> --device-del <dev_name> <options> --destroy-image-force
```

```
prlctl set <VM_name> --device-connect <dev_name>
prlctl set <VM_name> --device-disconnect <dev_name>
prlctl set <VM_name> --device-bootorder "dev_name1 dev_name2 [...]"
```

Name	Description
`<VM_name>`	Virtual machine name.
`--device-add <dev_type> <options>`	Adds a virtual device of the type `<dev_type>` to a virtual machine. `<dev_type>` can be: `hdd`, `cdrom`, `net`, `fdd`, `serial`, `usb`, `pci`. SCSI and VirtIO hard disks, network interfaces, and USB devices can be added to both running and stopped VMs, while other device types can only be added to stopped VMs.
`--device-set <dev_name> <options>`	Modifies the configuration of the virtual device `<dev_name>` in a virtual machine.
`--device-del <dev_name> <options>`	Deletes the virtual device `<dev_name>` from a virtual machine.
`--destroy-image-force`	Used with the `--device-del` option. Deletes a virtual machine HDD even if it is used in that virtual machine's snapshots.
`--device-connect <dev_name>`	Connects the virtual device <dev_name> to a running virtual machine.
`--device-disconnect <dev_name>`	Disconnects the virtual device <dev_name> from a running virtual machine.
`--device-bootorder "dev_name1 dev_name2 [...]"`	Specifies the boot order for a virtual machine.

Note: Device names can be obtained with the `prlctl list -i` command.

The device-related `<options>` can be subdivided into the following categories:

- hard disk drives

- optical disk drives

- network cards

- floppy disk drives

- serial ports

- USB devices

Each group of options is explained in the following subsections in detail.

Hard Disk Drive Management Options

This group of options is used to add and configure virtual hard disks in a virtual machine. The first syntax uses a file to emulate a hard disk drive. The second syntax connects a physical hard disk on the host server to the virtual machine.

```
prlctl set <VM_name> {--device-add hdd | --device-set hdd<N>}
          [--image <file> [--recreate]] [--size <size>] [--no-fs-resize]
          [--iface <ide|scsi|virtio>] [--subtype <virtio-scsi|hyperv>]
          [--position <pos>] [--enable|--disable]
prlctl set <VM_name> --device-add hdd --device <dev_name> [--position <pos>]
          [--iface <ide|scsi|virtio>]
prlctl set <VM_name> --backup-add <backup_ID> [--disk <disk_name>]
          [--iface <ide|scsi|virtio>] [--position <pos>]
prlctl set <VM_name> --backup-del {<backup_ID>|all}
```

Name	Description
`<VM_name>`	Virtual machine name.
`--device-add hdd`	Adds a virtual hard disk to the VM. New hard disks are created in the virtual machine directory and are automatically named `harddisk<N>.hdd`, where `<N>` is the next available disk index. SCSI and VirtIO hard disks can be added to both running and stopped VMs, IDE disks can only be added to stopped VMs.
`--device-set hdd<N>`	Modifies the parameters of an existing virtual hard disk. Virtual hard disks are named using the `hdd<N>` format where `<N>` is the drive index number starting from 0 (e.g., `hdd0`, `hdd1`). To obtain the list of disk names, use the `prlctl list` command with the `--info` option.
`--image <file>`	Specifies an existing image file that will be used to emulate the virtual disk. To recreate the image file, add the `--recreate` option.
`--device <dev_name>`	This option is used to connect a physical hard disk on the hardware node to the virtual machine. You can obtain the names of the existing hard disks on the server using the `prlsrvctl info` command.
`--size <size>`	The size of the virtual hard disk, in megabytes. The default size is 65536 MB.
`--no-fs-resize`	Used with `--size`. Do not resize the last partition on the hard disk during online resize.

Table 4.1.19.2.2 -- continued from previous page

Name	Description
`--enable`	Enables the specified virtual disk drive. All newly added disk drives are enabled by default (provided the `--disable` option is omitted).
`--disable`	Disables the specified virtual disk drive. The disk drive itself is not removed from the virtual machine configuration.
`--backup-add <backup_ID>`	Attaches the backup with the identifier <backup_ID> to the virtual machine as a virtual hard disk. To obtain the backup ID, use the `prlctl backup-list -f` command.
`--disk <disk_name>`	Used with `--backup-add`. The name of the disk in the backup to attach. If a disk is not specified, all disks contained in the backup will be attached. To obtain the disk name(s), use the `prlctl backup-list -f` command.
`--backup-del {<backup_ID>|all}`	Detach either the backup with the identifier <backup_ID> or detach all backups from the virtual machine.
`--iface <ide|scsi|virtio>`	Sets disk drive interface type. If omitted, the SCSI interface will be used.
`--subtype <virtio-scsi|hyperv>`	Sets paravirtualized SCSI controller type. If omitted, VirtIO SCSI will be used. Hyper-V is recommended for Windows virtual machines as it allows to use native hard disk drivers.
`--position <pos>`	The SCSI or IDE device identifier to be used for the virtual disk.

Optical Disk Drive Management Options

This group of options is used to add and configure virtual optical disk drives, such as DVD or CD drives.

```
prlctl set <VM_name> --device-add cdrom {--image <file> | --device <dev_name>}
          [--iface <ide|scsi>] [--subtype <virtio-scsi|hyperv>]
          [--position <pos>] [--enable|--disable] [--connect|--disconnect]
prlctl set <VM_name> --device-set cdrom<N> [--image <file>] [--device <dev_name>]
          [--iface <ide|scsi>] [--subtype <virtio-scsi|hyperv>]
          [--position <pos>] [--enable|--disable] [--connect|--disconnect]
```

Name	Description
`<VM_name>`	Virtual machine name.
`--device-add cdrom`	Adds a DVD/CD drive to a stopped virtual machine.

Continued on next page

Table 4.1.19.2.3 -- continued from previous page

Name	Description
`--device-set cdrom<N>`	Modifies the parameters of an existing virtual optical disk. The `<N>` postfix indicates the drive index number. To obtain the list of the available drives, use the `prlctl list` command with the `--info` option.
`--image <file>`	Mounts the specified disk image file in the virtual machine. Currently, the following image file formats are supported: `.iso`, `.cue`, `.ccd`, and `.dmg`. The image must not be compressed and/or encrypted.
`--device <dev_name>`	This option is used to connect a physical optical disk on the hardware node to the virtual machine. You can obtain the names of the existing optical disks on the server using the `prlsrvctl info` command.
`--iface <ide|scsi>`	Sets disk interface type: • `ide` - IDE disk. • `scsi` - SCSI disk (default).
`--subtype <virtio-scsi|hyperv>`	Sets paravirtualized SCSI controller type. If omitted, VirtIO SCSI will be used. Hyper-V is recommended for Windows virtual machines as it allows to use native optical disk drivers.
`--position <pos>`	The SCSI or IDE device identifier to be used for the DVD/CD drive. You can use one of the following formats for specifying IDs: `<ID>:<bus>`, `<ID>-<bus>`, `<ID>`. For example, if you specify `3:0` (or `3-0` or `3`) as `<number>` for a SCSI drive, the guest OS will see the drive as having ID 3 on SCSI bus 0.
`--enable`	Enables the specified DVD/CD drive. All newly added drives are enabled by default (provided the `--disable` option is omitted).
`--disable`	Disables the specified optical disk drive. The disk drive itself is not removed from the virtual machine configuration.
`--connect`	Automatically connect the specified optical disk drive during the virtual machine startup process.
`--disconnect`	Do not automatically connect the specified optical disk drive during the virtual machine startup process.

Network Adapter Management Options

This group of options is used to manage virtual network adapters in a virtual machine.

```
prlctl set <VM_name> {--device-add net | --device-set net<N>}
         {--type routed | --network <network_ID>} [--mac {<addr>|auto}]
         [{--ipadd <addr>[/<mask>] --ipdel <addr>[/<mask>] | --dhcp <yes|no>
         --dhcp6 <yes|no>}] [--gw <gw>] [--gw6 <gw>]
         [--nameserver <addr>] [--searchdomain <addr>]
         [--configure <yes|no>] [--ipfilter <yes|no>] [--macfilter <yes|no>]
         [--preventpromisc <yes|no>] [--enable|--disable]
         [--connect|--disconnect] [--adapter-type <e1000|rtl|virtio>]
```

Name	Description
`<VM_name>`	Virtual machine name.
`--device-add net`	Adds a new virtual network adapter to a running or stopped virtual machine.
`--device-set net<N>`	Modifies an existing virtual network adapter. To obtain the list of the available adapters, use the `prlctl list` command with the `--info` option.
`--type routed`	Sets the networking mode for the virtual network adapter to "routed". In this mode, the network adapter is communicating with the outside world through an internal virtual network adapter.
`--network <network_ID>`	Sets the networking mode for the virtual network adapter to "virtual_network". In this mode the adapter is connected to a virtual network specified by `<network_ID>`.
`--mac {<addr>|auto}`	Specifies the MAC address to assign to an existing network adapter. Specify a desired MAC address using the `addr` parameter value or use the `auto` option to generate the existing address automatically.
`--ipadd <addr>[/<mask>]`	Adds an IP address and a mask (optional) to the network adapter.
`--ipdel <addr>[/<mask>]`	Deletes an IP address from the network adapter.
`--dhcp <yes|no>`	Specifies whether the virtual network adapter should obtain the IPv4 settings through a DHCP server.
`--dhcp6 <yes|no>`	Specifies whether the virtual network adapter should obtain the IPv6 settings through a DHCP server .

Table 4.1.19.2.4 -- continued from previous page

Name	Description
`--gw <gw>`	The default gateway to be used by the virtual machine.
`--gw6 <gw>`	The default IPv6 gateway to be used by the virtual machine.
`--nameserver <addr>`	The default DNS server address to be used by the virtual machine.
`--searchdomain <addr>`	The default search domain to be used by the virtual machine.
`--configure <yes|no>`	If set to yes, the settings above are applied to the virtual network adapter instead of its original settings. Configuring any of the settings above automatically sets this option to yes.
`--ipfilter <yes|no>`	Determines if the specified network adapter is configured to filter network packages by IP address. If set to yes, the adapter is allowed to send packages only from IPs in the network adapter IP addresses list.
`--macfilter <yes|no>`	Determines if the specified network adapter is configured to filter network packages by MAC address. If set to yes, the adapter is allowed to send packages only from its own MAC address.
`--preventpromisc <yes|no>`	Determines if the specified network adapter should reject packages not addressed to its virtual machine. If set to yes, the adapter will drop such packages.
`--enable|--disable`	Enables or disable the network adapter. If omitted during the adapter creation, the adapter will be enabled.
`--connect|--disconnect`	Connects or disconnects the network adapter. When disconnected, the adapter is not removed from the virtual machine.

Continued on next page

<div align="center">Table 4.1.19.2.4 -- continued from previous page</div>

Name	Description
`--adapter-type <e1000|rtl|virtio>`	Emulated network adapter: • `e1000` - Intel 82545EM, • `rtl` - Realtek RTL8029, • `virtio` - VirtIO. **Note:** The VirtIO adapter is not supported on the Windows Server 2008 guest operating system.

Floppy Disk Drive Management Options

This group of options is used to add a floppy disk drive to a virtual machine and to modify the existing virtual floppy disk drive.

```
prlctl set <VM_name> {--device-add fdd | --device-set fdd0}
        [--image <file>] [--enable|--disable]
        [--connect|--disconnect]
```

Name	Description
`<VM_name>`	Virtual machine name.
`--device-add fdd`	Adds a new floppy disk drive to a stopped virtual machine.
`--device-set fdd0`	Modifies the parameters of the existing virtual floppy disk drive.
`--image <file>`	The name and path of an existing floppy disk image file (usually `floppy.fdd`) to mount in the virtual machine.
`--enable`	Enables the specified floppy disk drive. All newly added floppy drives are enabled by default (provided the `--disable` option was omitted during the drive creation).
`--disable`	Disables the specified floppy disk drive. The drive itself is not removed from the virtual machine configuration.
`--connect`	Connect the specified floppy disk drive automatically during the virtual machine startup process.
`--disconnect`	Use this option if you don't want the specified floppy disk drive automatically connected to the virtual machine on its start.

Serial Port Management Options

This group of options is used to manage serial ports in a virtual machine.

```
prlctl set <VM_name> {--device-add serial | --device-add serial<N>}
          {--device <dev_name> | --output <file> | --socket <name>}
          [--enable|--disable] [--connect|--disconnect]
```

Name	Description
<VM_name>	Virtual machine name.
--device-add serial	Adds a new serial port to a stopped virtual machine.
--device-set serial<N>	Modifies the parameters of an existing serial port.
--device <dev_name>	The name of the physical serial port to which to connect the virtual machine.
--output <file>	The name and path of the output file to which to connect the virtual serial port.
--socket <name>	The name of the physical socket to which to connect the virtual serial port.
--enable|--disable	Enables or disables the virtual serial port. All newly added serial ports are enabled by default (provided the `--disable` option is omitted).
--connect	Automatically connect the virtual serial port during the virtual machine startup process.
--disconnect	Do not automatically connect the virtual serial port during the virtual machine startup process.

USB Controller Management Options

This group of options is used to manage the USB controller in a virtual machine.

```
prlctl set <VM_name> --device-add usb [--enable|--disable]
```

Name	Description
<VM_name>	Virtual machine name.
--device-add usb	Add a USB device to a running or stopped virtual machine.
--enable|--disable	Enables or disables the USB controller. The controller is enabled by default (provided the `--disable` option is omitted).

Removing Devices from Virtual Machines

The `--device-del` option is used to remove virtual devices from a virtual machine. USB devices, optical and floppy disk drives can be removed from both running and stopped VMs while hard disk drives, network interfaces, and serial ports can only be removed from stopped VMs.

```
prlctl set <VM_name> --device-del <dev_name> [--detach-only|--destroy-image]
```

Name	Description
`<dev_name>`	The name of the virtual device to delete from the virtual machine. To obtain the list of virtual devices, use the `prlctl list -i` command.
`--detach-only`	Deletes the information about the specified device from the virtual machine configuration.
`--destroy-image`	Deletes the information about the specified device from the virtual machine configuration and removes the device from the server.

4.1.20 prlctl snapshot, snapshot-list, snapshot-switch, snapshot-delete

Takes, displays, reverts to, and deletes snapshots of a running virtual machine.

```
prlctl snapshot <VM_name> [-n, --name <name>] [-d, --description <desc>]
prlctl snapshot-list <VM_name> [-t, --tree] [-i, --id <snapshot_ID>]
prlctl snapshot-switch <VM_name> -i, --id <snapshot_ID>
prlctl snapshot-delete <VM_name> -i, --id <snapshot_ID> [-c,--children]
```

Name	Description
`<VM_name>`	Virtual machine name.
`-n, --name <name>`	User-defined snapshot name. Names with white spaces must be enclosed in quotation marks.
`-d, --description <desc>`	User-defined snapshot description. Descriptions with white spaces must be enclosed in quotation marks.
`-t, --tree`	Displays the snapshot list as a tree. The default display format is tabular with **Parent Snapshot ID** and **Snapshot ID** as columns.

Continued on next page

Table 4.1.20.1 -- continued from previous page

Name	Description
`-i, --id <snapshot_ID>`	- Use with `prlctl snapshot-list` to specify the ID of the snapshot to use as the root. If this parameter is omitted, the entire snapshot tree will be displayed. - Use with `prlctl snapshot-switch` to specify the ID of the snapshot to revert to. - Use with `prlctl snapshot-delete` to specify the ID of the snapshot to delete.
`-c,--children`	If the snapshot you want to delete has children snapshots derived from it, they will be deleted. If the option is omitted, they become the children of the deleted snapshot parent.

4.1.21 prlctl start, stop, restart, reset, status

Start, stop, reset, and check the status of a virtual machine.

```
prlctl start <VM_name>
prlctl stop <VM_name> [--kill]
prlctl restart <VM_name>
prlctl reset <VM_name>
prlctl status <VM_name>
```

Name	Description
`<VM_name>`	The name of the virtual machine to start, stop, restart, reset, or check the status of.
`--kill`	Perform a hard virtual machine shutdown. If this option is omitted, an attempt to perform a graceful shutdown will be made.

The `stop` command can perform a hard or a graceful virtual machine shutdown. If the `--kill` parameter is included, the hard shutdown will be performed. If the parameter is omitted, the outcome of the graceful shutdown attempt will depend on the following:

- If Virtuozzo Hybrid Server tools are installed in a virtual machine, the graceful shutdown will be performed using its facilities.

- If Virtuozzo Hybrid Server tools are not installed, the command will try to perform a graceful shutdown

using ACPI. Depending on the ACPI support availability in the guest operating system, this may work or not.

The `restart` command first gracefully shuts down a virtual machine and then starts it again.

The `reset` command resets a virtual machine without shutting it down.

Note: Resetting a VM may result in loss of unsaved data stored in that VM.

The `start` command can be used to start a stopped virtual machine or to resume a paused virtual machine).

4.1.22 prlctl statistics

Print statistics for running virtual machines on the server.

```
prlctl statistics {<VM_UUID_or_name>|-a, --all} [--loop] [--filter <filter>]
```

Name	Description
`-a, --all`	Print statistics for all virtual machines and containers on the server.
`--loop`	Print statistics every second until the program is terminated.
`--filter <pattern>`	Specifies the subset of performance statistics to collect and print. If omitted, all available statistics is shown. Asterisks (*) can be used as wildcards for any number of arbitrary characters. The available filters are listed below (`<N>` is the device or file system index).

4.1.22.1 Available Filters

Storage device statistics

- `devices.{ide|scsi|sata}<N>.read_requests` - Total count of read requests to IDE, SCSI, or SATA controller

- `devices.{ide|scsi|sata}<N>.read_total` - Total count of read bytes for IDE, SCSI, or SATA controller

- `devices.{ide|scsi|sata}<N>.write_requests` - Total count of write requests to IDE, SCSI, or SATA controller

- `devices.{ide|scsi|sata}<N>.write_total` - Total count of written bytes for IDE, SCSI, or SATA controller

Network statistics

- `net.nic<N>.pkts_in` - Total number of incoming packets for network adapter

- `net.nic<N>.pkts_out` - Total number of outgoing packets for network adapter

- `net.nic<N>.bytes_in` - Total number of incoming bytes for network adapter

- `net.nic<N>.bytes_out` - Total number of outgoing bytes for network adapter

Classful network statistics

The result is provided in five columns: **Class**, **Input(bytes)**, **Input(packets)**, **Output(bytes)**, **Output(packets)**.

- `net.classful.traffic` - Total counters for IPv4 and IPv6 traffic

- `net.classful.traffic.ipv4` - Counters for IPv4 traffic

- `net.classful.traffic.ipv6` - Counters for IPv6 traffic

CPU statistics

- `guest.cpu.usage` - Guest OS CPU usage, in percent

- `guest.cpu.time` - Sum of guest CPU time differences since the last query for each vCPU averaged by the number of host CPUs, in microseconds

- `host.cpu.time` - Sum of host CPU time differences since the last query for each vCPU averaged by the number of host CPUs, in microseconds

- `guest.vcpu<N>.time` - per-vCPU statistics, in nanoseconds

RAM statistics

- `guest.ram.usage` - Guest OS used RAM, in MiB

- `guest.ram.cached` - Guest OS cached RAM, in MiB

- `guest.ram.total` - Guest OS total RAM, in MiB

- `guest.ram.swap_in` - Guest OS virtual memory stats, in counts

- `guest.ram.swap_out` - Guest OS virtual memory stats, in counts

- `guest.ram.minor_fault` - Guest OS minor page fault count

- `guest.ram.major_fault` - Guest OS major page fault count

- `guest.ram.balloon_actual` - Guest OS balloon size, in MiB

Mounted file systems statistics

- `guest.fs<N>.name` - Device name as seen from inside the guest file system

- `guest.fs<N>.total` - Total size of the file system, in bytes

- `guest.fs<N>.free` - Amount of free space on the file system, in bytes

- `guest.fs<N>.disk.<N>` - Disk indices

4.2 Virtual Disk Management Utilities

The `prl_disk_tool` utility is used to manage virtual hard disk drives.

> **Warning:**   Only use `prl_disk_tool` on disks of stopped virtual machines.

```
prl_disk_tool <command> [<options>] --hdd <disk_path> [<options>]
prl_disk_tool --help
```

4.2.1 prl_disk_tool compact

Removes all empty blocks from the expanding virtual disk to reduce its size on the physical hard disk. The virtual disk must be formatted to NTFS, ext2/ext3/ext4, btrfs, or xfs.

```
prl_disk_tool compact --hdd <disk_path> [--force]
prl_disk_tool compact -i, --info --hdd <disk_path>
```

Name	Description
`--hdd <disk_path>`	Full path to the virtual disk.
`--force`	Forces the compacting operation for suspended virtual disks.
`-i, --info`	Do not compact the virtual disk; just display the information about the size the disk will have after compacting.

4.2.2 prl_disk_tool merge

Merges all snapshots of the virtual hard disk.

```
prl_disk_tool merge --hdd <disk_path>
```

Name	Description
--hdd <disk_path>	Full path to the virtual disk.

4.2.3 prl_disk_tool resize

Changes the capacity of the specified virtual disk. During resizing, all data present on the disk volumes are left intact. You can also resize the last partition using the --resize_partition option. The supported file systems are NTFS, ext2/ext3/ext4, btrfs, or xfs.

```
prl_disk_tool resize --size <size>[K|M|G|T] [--resize_partition]
                     --hdd <disk_path> [--force]
prl_disk_tool resize -i, --info [--units <K|M|G|T>] --hdd <disk_path>
```

Name	Description
--size	The new size of the virtual disk. It can be set in kilobytes (K), megabytes (M, default), gigabytes (G), or terabytes (T).
--resize_partition	Resizes the last partition of the specified virtual disk. **Note:** You cannot reduce XFS file systems (the default choice for CentOS 7 and Red Hat Enterprise Linux 7).
--hdd <disk_path>	Full path to the virtual disk.
--force	Forces the resizing operation for suspended virtual disks.
-i, --info	Do not resize the virtual disk; just show the size the disk will have after resizing.
--units	Displays the disk size in kilobytes (K), megabytes (M, default), gigabytes (G), or terabytes (T).