Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'containers.conf.5' command

$ man containers.conf.5

containers.conf(5) configuration containers.conf(5)

NAME

containers.conf - The container engine configuration file specifies de? fault configuration options and command-line flags for container en? gines.

DESCRIPTION

Container engines like Podman & Buildah read containers.conf file, if it exists and modify the defaults for running containers on the host. containers.conf uses a TOML format that can be easily modified and ver? sioned.

Container engines read the /usr/share/containers/containers.conf and /etc/containers/containers.conf, and /etc/containers/contain? ers.conf.d/.conf files if they exist. When running in rootless mode, they also read $HOME/.config/containers/containers.conf and $HOME/.con? fig/containers/containers.conf.d/.conf files.

Fields specified in containers conf override the default options, as well as options in previously read containers.conf files.

Config files in the .d directories, are added in alpha numeric sorted order and must end in .conf.

Not all options are supported in all container engines.

Note container engines also use other configuration files for configur? ing the environment.

? storage.conf for configuration of container and images stor?

age.

? registries.conf for definition of container registires to search while pulling. container images.

? policy.conf for controlling which images can be pulled to the system.

FORMAT

The TOML format ?https://github.com/toml-lang/toml? is used as the en? coding of the configuration file. Every option is nested under its ta? ble. No bare options are used. The format of TOML can be simplified to:

```
[table1]
option = value
[table2]
option = value
[table3]
option = value
[table3.subtable1]
option = value
```

CONTAINERS TABLE

The containers table contains settings to configure and manage the OCI runtime.

annotations = [] List of annotations. Specified as "key=value" pairs to be added to all containers.

Example: "run.oci.keep_original_groups=1"

apparmor_profile="container-default"

Used to change the name of the default AppArmor profile of container engines. The default profile name is "container-default".

base_hosts_file=""

The hosts entries from the base hosts file are added to the containers hosts file. This must be either an absolute path or as special values "image" which uses the hosts file from the container image or "none" which means no base hosts file is used. The default is "" which will use /etc/hosts.

cgroups="enabled"

Determines whether the container will create CGroups. Options are:

enabled Enable cgroup support within container

disabled Disable cgroup support, will inherit cgroups from parent

no-conmon Do not create a cgroup dedicated to conmon.

cgroupns="private"

Default way to to create a cgroup namespace for the container. Options are: private Create private Cgroup Namespace for the container. host Share host Cgroup Namespace with the container.

default_capabilities=[]

List of default capabilities for containers.

The default list is:

```
default_capabilities = [
    "CHOWN",
    "DAC_OVERRIDE",
    "FOWNER",
    "FSETID",
    "KILL",
    "NET_BIND_SERVICE",
    "SETFCAP",
    "SETGID",
    "SETPCAP",
    "SETUID",
]
```

Note, by default container engines using containers.conf, run with less capabilities than Docker. Docker runs additionally with "AUDIT_WRITE", "MKNOD", "NET_RAW", "CHROOT". If you need to add one of these capabili? ties for a particular container, you can use the --cap-add option or edit your system's containers.conf.

default_sysctls=[]

A list of sysctls to be set in containers by default, specified as "name=value".

Example:"net.ipv4.ping_group_range=0 1000".

default_ulimits=[]

A list of ulimits to be set in containers by default, specified as "name=soft-limit:hard-limit".

Example: "nofile=1024:2048".

devices=[]

List of devices. Specified as 'device-on-host:device-on-container:per?missions'.

Example: "/dev/sdc:/dev/xvdc:rwm".

dns_options=[]

List of default DNS options to be added to /etc/resolv.conf inside of the container.

dns_searches=[]

List of default DNS search domains to be added to /etc/resolv.conf in?side of the container.

dns_servers=[]

A list of dns servers to override the DNS configuration passed to the container. The special value ?none? can be specified to disable cre?ation of /etc/resolv.conf in the container.

env=["PATH=/usr/local/sbin:/usr/lo?cal/bin:/usr/sbin:/usr/bin:/sbin:/bin", "TERM=xterm"]

Environment variable list for the container process, used for passing environment variables to the container.

env_host=false

Pass all host environment variables into the container.

host_containers_internal_ip=""

Set the ip for the host.containers.internal entry in the containers /etc/hosts file. This can be set to "none" to disable adding this en?try. By default it will automatically choose the host ip.

NOTE: When using podman machine this entry will never be added to the containers hosts file instead the gvproxy dns resolver will resolve this hostname. Therefore it is not possible to disable the entry in this case.

http_proxy=true

Default proxy environment variables will be passed into the container.

The environment variables passed in include: http_proxy, https_proxy, ftp_proxy, no_proxy, and the upper case versions of these. The no_proxy option is needed when host system uses a proxy but container should not use proxy. Proxy environment variables specified for the container in any other way will override the values passed from the host.

init=false

Run an init inside the container that forwards signals and reaps pro? cesses.

init_path="/usr/libexec/podman/catatonit"

Path to the container-init binary, which forwards signals and reaps processes within containers. Note that the container-init binary will only be used when the --init for podman-create and podman-run is set.

ipcns="shareable"

Default way to to create a IPC namespace for the container. Options are:

host Share host IPC Namespace with the container.

none Create shareable IPC Namespace for the container without a private /dev/shm.

private Create private IPC Namespace for the container, other con? tainers are not allowed to share it.

shareable Create shareable IPC Namespace for the container.

keyring=true

Indicates whether the container engines create a kernel keyring for use within the container.

label=true

Indicates whether the container engine uses MAC(SELinux) container sep? aration via labeling. This option is ignored on disabled systems.

log_driver=""

Logging driver for the container. Currently available options are k8s- file, journald, none and passthrough, with json-file aliased to k8s- file for scripting compatibility. The journald driver is used by de? fault if the systemd journal is readable and writable. Otherwise, the k8s-file driver is used.

log_size_max=-1

Maximum size allowed for the container's log file. Negative numbers in?

dicate that no size limit is imposed. If it is positive, it must be >=

8192 to match/exceed conmon's read buffer. The file is truncated and

re-opened so the limit is never exceeded.

log_tag=""

Default format tag for container log messages. This is useful for cre?

ating a specific tag for container log messages. Container log messages

default to using the truncated container ID as a tag.

netns="private"

Default way to to create a NET namespace for the container. Options

are:

private Create private NET Namespace for the container.

host Share host NET Namespace with the container.

none Containers do not use the network.

no_hosts=false

Create /etc/hosts for the container. By default, container engines man?

age /etc/hosts, automatically adding the container's own IP ad?

dress.

pidns="private"

Default way to to create a PID namespace for the container. Options

are:

private Create private PID Namespace for the container.

host Share host PID Namespace with the container.

pids_limit=1024

Maximum number of processes allowed in a container. 0 indicates that no

limit is imposed.

prepare_volume_on_create=false

Copy the content from the underlying image into the newly created vol?

ume when the container is created instead of when it is started. If

false, the container engine will not copy the content until the con?

tainer is started. Setting it to true may have negative performance im?

plications.

read_only=true|false

Run all containers with root file system mounted read-only. Set to false by default.

seccomp_profile="/usr/share/containers/seccomp.json"

Path to the seccomp.json profile which is used as the default seccomp profile for the runtime.

shm_size="65536k"

Size of /dev/shm. The format is <number><unit>. number must be greater than 0. Unit is optional and can be: b (bytes), k (kilobytes), m(megabytes), or g (gigabytes). If you omit the unit, the system uses bytes. If you omit the size entirely, the system uses 65536k.

tz=""

Set timezone in container. Takes IANA timezones as well as local, which sets the timezone in the container to match the host machine. If not set, then containers will run with the time zone specified in the im? age.

Examples:

tz="local"

tz="America/New_York"

umask="0022"

Sets umask inside the container.

userns="host"

Default way to to create a USER namespace for the container. Options are:

private Create private USER Namespace for the container.

host Share host USER Namespace with the container.

utsns="private"

Default way to to create a UTS namespace for the container. Options are:

private Create private UTS Namespace for the container.

host Share host UTS Namespace with the container.

volumes=[]

List of volumes. Specified as "directory-on-host:directory-in-con?

tainer:options".

Example: "/db:/var/lib/db:ro".

NETWORK TABLE

The network table contains settings pertaining to the management of CNI plugins.

network_backend=""

Network backend determines what network driver will be used to set up and tear down container networks. Valid values are "cni" and "ne? tavark". The default value is empty which means that it will automati? cally choose CNI or netavark. If there are already containers/images or CNI networks preset it will choose CNI.

Before changing this value all containers must be stopped otherwise it is likely that iptables rules and network interfaces might leak on the host. A reboot will fix this.

cni_plugin_dirs=[]

List of paths to directories where CNI plugin binaries are located.

The default list is:

```
cni_plugin_dirs = [
  "/usr/local/libexec/cni",
  "/usr/libexec/cni",
  "/usr/local/lib/cni",
  "/usr/lib/cni",
  "/opt/cni/bin",
]
```

default_network="podman"

The network name of the default network to attach pods to.

default_subnet="10.88.0.0/16"

The subnet to use for the default network (named above in default_net? work). If the default network does not exist, it will be automatically created the first time a tool is run using this subnet.

default_subnet_pools=[]

DefaultSubnetPools is a list of subnets and size which are used to al? locate subnets automatically for podman network create. It will iter?

ate through the list and will pick the first free subnet with the given size. This is only used for ipv4 subnets, ipv6 subnets are always as? signed randomly.

The default list is (10.89.0.0-10.255.255.0/24):

```
default_subnet_pools = [
  {"base" = "10.89.0.0/16", "size" = 24},
  {"base" = "10.90.0.0/15", "size" = 24},
  {"base" = "10.92.0.0/14", "size" = 24},
  {"base" = "10.96.0.0/11", "size" = 24},
  {"base" = "10.128.0.0/9", "size" = 24},
 ]
```

network_config_dir="/etc/cni/net.d/"

Path to the directory where network configuration files are located. For the CNI backend the default is "/etc/cni/net.d" as root and "$HOME/.config/cni/net.d" as rootless. For the netavark backend "/etc/containers/networks" is used as root and "$graphroot/networks" as rootless.

dns_bind_port=53

Port to use for dns forwarding daemon with netavark in rootful bridge mode and dns enabled. Using an alternate port might be useful if other dns services should run on the machine.

ENGINE TABLE

The engine table contains configuration options used to set up con? tainer engines such as Podman and Buildah.

active_service=""

Name of destination for accessing the Podman service. See SERVICE DES? TINATION TABLE below.

cgroup_manager="systemd"

The cgroup management implementation used for the runtime. Supports cgroupfs and systemd.

conmon_env_vars=[]

Environment variables to pass into Conmon.

conmon_path=[]

Paths to search for the conmon container manager binary. If the paths are empty or no valid path was found, then the $PATH environment vari? able will be used as the fallback.

The default list is:

```
conmon_path=[
    "/usr/libexec/podman/conmon",
    "/usr/local/libexec/podman/conmon",
    "/usr/local/lib/podman/conmon",
    "/usr/bin/conmon",
    "/usr/sbin/conmon",
    "/usr/local/bin/conmon",
    "/usr/local/sbin/conmon",
    "/run/current-system/sw/bin/conmon",
]
```

detach_keys="ctrl-p,ctrl-q"

Keys sequence used for detaching a container. Specify the keys se? quence used to detach a container. Format is a single character [a-Z] or a comma separated sequence of ctrl-<value>, where <value> is one of: a-z, @, ^, [, \,], ^ or _

enable_port_reservation=true

Determines whether the engine will reserve ports on the host when they are forwarded to containers. When enabled, when ports are forwarded to containers, they are held open by conmon as long as the container is running, ensuring that they cannot be reused by other programs on the host. However, this can cause significant memory usage if a container has many ports forwarded to it. Disabling this can save memory.

env=[]

Environment variables to be used when running the container engine (e.g., Podman, Buildah). For example "http_proxy=internal.proxy.com? pany.com". Note these environment variables will not be used within the container. Set the env section under [containers] table, if you want to set environment variables for the container.

events_logfile_path=""

Define where event logs will be stored, when events_logger is "file".

events_logfile_max_size="1m"

Sets the maximum size for events_logfile_path. The unit can be b (bytes), k (kilobytes), m (megabytes) or g (gigabytes). The format for the size is <number><unit>, e.g., 1b or 3g. If no unit is included then the size will be in bytes. When the limit is exceeded, the log? file will be rotated and the old one will be deleted. If the maximumn size is set to 0, then no limit will be applied, and the logfile will not be rotated.

events_logger="journald"

The default method to use when logging events.

The default method is different based on the platform that Podman is being run upon. To determine the current value, use this command:

podman info --format {{.Host.EventLogger}

Valid values are: file, journald, and none.

events_container_create_inspect_data=true|false

Creates a more verbose container-create event which includes a JSON payload with detailed information about the container. Set to false by default.

helper_binaries_dir=["/usr/libexec/podman", ...]

A is a list of directories which are used to search for helper bina? ries.

The default paths on Linux are: - /usr/local/libexec/podman - /usr/lo? cal/lib/podman - /usr/libexec/podman - /usr/lib/podman

The default paths on macOS are: - /usr/local/opt/podman/libexec - /opt/homebrew/bin - /opt/homebrew/opt/podman/libexec - /usr/lo? cal/bin - /usr/local/libexec/podman - /usr/local/lib/podman - /usr/libexec/podman - /usr/lib/podman

The default path on Windows is: - C:\Program Files\RedHat\Podman

hooks_dir=["/etc/containers/oci/hooks.d", ...]

Path to the OCI hooks directories for automatically executed hooks.

image_default_format="oci"|"v2s2"|"v2s1"

Manifest Type (oci, v2s2, or v2s1) to use when pulling, pushing, build?

ing container images. By default images pulled and pushed match the format of the source image. Building/committing defaults to OCI. Note: image_build_format is deprecated.

image_default_transport="docker://"

Default transport method for pulling and pushing images.

image_parallel_copies=0

Maximum number of image layers to be copied (pulled/pushed) simultaneously. Not setting this field will fall back to containers/image defaults. (6)

image_volume_mode="bind"

Tells container engines how to handle the builtin image volumes.

- bind: An anonymous named volume will be created and mounted into the container.
- tmpfs: The volume is mounted onto the container as a tmpfs, which allows the users to create content that disappears when the container is stopped.
- ignore: All volumes are just ignored and no action is taken.

infra_command="/pause"

Infra (pause) container image command for pod infra containers. When running a pod, we start a /pause process in a container to hold open the namespaces associated with the pod. This container does nothing other then sleep, reserving the pods resources for the lifetime of the pod.

infra_image=""

Infra (pause) container image for pod infra containers. When running a pod, we start a pause process in a container to hold open the namespaces associated with the pod. This container does nothing other then sleep, reserving the pods resources for the lifetime of the pod. By default container engines run a builtin container using the pause executable. If you want override specify an image to pull.

lock_type="shm"

Specify the locking mechanism to use; valid values are "shm" and "file". Change the default only if you are sure of what you are doing,

in general "file" is useful only on platforms where cgo is not avail?
able for using the faster "shm" lock type. You may need to run "podman
system renumber" after you change the lock type.

multi_image_archive=false

Allows for creating archives (e.g., tarballs) with more than one image.
Some container engines, such as Podman, interpret additional arguments
as tags for one image and hence do not store more than one image. The
default behavior can be altered with this option.

namespace=""

Default engine namespace. If the engine is joined to a namespace, it
will see only containers and pods that were created in the same name?
space, and will create new containers and pods in that namespace. The
default namespace is "", which corresponds to no namespace. When no
namespace is set, all containers and pods are visible.

network_cmd_path=""

Path to the slirp4netns binary.

network_cmd_options=[]

Default options to pass to the slirp4netns binary.

Valid options values are:

- ? allow_host_loopback=true|false: Allow the slirp4netns to reach the host loopback IP (10.0.2.2). Default is false.
- ? mtu=MTU: Specify the MTU to use for this network. (Default is 65520).
- ? cidr=CIDR: Specify ip range to use for this network. (Default is 10.0.2.0/24).
- ? enable_ipv6=true|false: Enable IPv6. Default is true. (Re? quired for outbound_addr6).
- ? outbound_addr=INTERFACE: Specify the outbound interface slirp should bind to (ipv4 traffic only).
- ? outbound_addr=IPv4: Specify the outbound ipv4 address slirp should bind to.
- ? outbound_addr6=INTERFACE: Specify the outbound interface slirp should bind to (ipv6 traffic only).

? outbound_addr6=IPv6: Specify the outbound ipv6 address slirp should bind to.

? port_handler=rootlesskit: Use rootlesskit for port forwarding. Default. Note: Rootlesskit changes the source IP address of incoming packets to a IP address in the container network namespace, usually 10.0.2.100. If your application requires the real source IP address, e.g. web server logs, use the slirp4netns port handler. The rootlesskit port handler is also used for rootless containers when connected to user-defined networks.

? port_handler=slirp4netns: Use the slirp4netns port forwarding, it is slower than rootlesskit but preserves the correct source IP address. This port handler cannot be used for user-defined networks.

no_pivot_root=false

Whether to use chroot instead of pivot_root in the runtime.

num_locks=2048

Number of locks available for containers and pods. Each created con? tainer or pod consumes one lock. The default number available is 2048. If this is changed, a lock renumbering must be performed, using the podman system renumber command.

pod_exit_policy="continue"

Set the exit policy of the pod when the last container exits. Sup? ported policies are:

??

?Exit Policy ? Description ?

??

?continue ? The pod continues running ?

? ? when the last container ?

? ? exits. Used by default. ?

??

?stop ? The pod is stopped when ?

? ? the last container exits. ?

? ? Used in play kube. ?

??

pull_policy="always"|"missing"|"never"

Pull image before running or creating a container. The default is miss?

ing.

? missing: attempt to pull the latest image from the registries

listed in registries.conf if a local image does not exist.

Raise an error if the image is not in any listed registry and

is not present locally.

? always: pull the image from the first registry it is found in

as listed in registries.conf. Raise an error if not found in

the registries, even if the image is present locally.

? never: do not pull the image from the registry, use only the

local version. Raise an error if the image is not present lo?

cally.

remote = false Indicates whether the application should be running in

remote mode. This flag modifies the --remote option on container en?

gines. Setting the flag to true will default podman --remote=true for

access to the remote Podman service.

runtime=""

Default OCI specific runtime in runtimes that will be used by default.

Must refer to a member of the runtimes table. Default runtime will be

searched for on the system using the priority: "crun", "runc", "kata".

runtime_supports_json=["crun", "runc", "kata", "runsc", "youki",

"krun"]

The list of the OCI runtimes that support --format=json.

runtime_supports_kvm=["kata", "krun"]

The list of OCI runtimes that support running containers with KVM sepa?

ration.

runtime_supports_nocgroups=["crun", "krun"]

The list of OCI runtimes that support running containers without

CGroups.

image_copy_tmp_dir="/var/tmp"

Default location for storing temporary container image content. Can be overridden with the TMPDIR environment variable. If you specify "stor? age", then the location of the container/storage tmp directory will be used. If set then it is the users responsibility to cleanup storage. Configure tmpfiles.d(5) to cleanup storage.

service_timeout=5

Number of seconds to wait without a connection before the podman sys? tem service times out and exits

static_dir="/var/lib/containers/storage/libpod"

Directory for persistent libpod files (database, etc). By default this will be configured relative to where containers/storage stores contain? ers.

stop_timeout=10

Number of seconds to wait for container to exit before sending kill signal.

exit_command_delay=300

Number of seconds to wait for the API process for the exec call before sending exit command mimicking the Docker behavior of 5 minutes (in seconds).

tmp_dir="/run/libpod"

The path to a temporary directory to store per-boot container. Must be a tmpfs (wiped after reboot).

volume_path="/var/lib/containers/storage/volumes"

Directory where named volumes will be created in using the default vol? ume driver. By default this will be configured relative to where con? tainers/storage store containers. This convention is followed by the default volume driver, but may not be by other drivers.

chown_copied_files=true

Determines whether file copied into a container will have changed own? ership to the primary uid/gid of the container.

compression_format=""

Specifies the compression format to use when pushing an image. Sup? ported values are: gzip, zstd and zstd:chunked.

SERVICE DESTINATION TABLE

The service_destinations table contains configuration options used to set up remote connections to the podman service for the podman API.

[service_destinations.{name}] URI to access the Podman service uri="ssh://user@production.example.com/run/user/1001/podman/pod?man.sock"

Example URIs:

? rootless local - unix://run/user/1000/podman/podman.sock

? rootless remote - ssh://user@engineering.lab.com?pany.com/run/user/1000/podman/podman.sock

? rootful local - unix://run/podman/podman.sock

? rootful remote - ssh://root@10.10.1.136:22/run/podman/pod?man.sock

identity="~/.ssh/id_rsa

Path to file containing ssh identity key

[engine.volume_plugins]

A table of all the enabled volume plugins on the system. Volume plugins can be used as the backend for Podman named volumes. Individual plugins are specified below, as a map of the plugin name (what the plugin will be called) to its path (filepath of the plugin's unix socket).

[engine.platform_to_oci_runtime]

Allows end users to switch the OCI runtime on the bases of container image's platform string. Following config field contains a map of platform/string = oci_runtime.

SECRET TABLE

The secret table contains settings for the configuration of the secret subsystem.

driver=file

Name of the secret driver to be used. Currently valid values are:

* file

* pass

[secrets.opts]

The driver specific options object.

MACHINE TABLE

The machine table contains configurations for podman machine VMs

cpus=1 Number of CPU's a machine is created with.

disk_size=10

The size of the disk in GB created when init-ing a podman-machine VM

image=""

Default image URI when creating a new VM using podman machine init. Options: On Linux/Mac, testing, stable, next. On Windows, the major version of the OS (e.g 36) for Fedora 36. For all platforms you can al? ternatively specify a custom download URL to an image. Container en? gines translate URIs $OS and $ARCH to the native OS and ARCH. URI "https://example.com/$OS/$ARCH/foobar.ami" would become "https://exam? ple.com/linux/amd64/foobar.ami" on a Linux AMD machine. The default value is testing on Linux/Mac, and on Windows.

memory=2048

Memory in MB a machine is created with.

user=""

Username to use and create on the podman machine OS for rootless con? tainer access. The default value is user. On Linux/Mac the default is? core.

volumes=["$HOME:$HOME"]

Host directories to be mounted as volumes into the VM by default. En? vironment variables like $HOME as well as complete paths are supported for the source and destination. An optional third field :ro can be used to tell the container engines to mount the volume readonly. On Mac, the default volumes are: "/Users:/Users", "/private:/private", "/var/folders:/var/folders"

FILES

containers.conf

Distributions often provide a /usr/share/containers/containers.conf file to define default container configuration. Administrators can override fields in this file by creating /etc/containers/contain? ers.conf to specify their own configuration. Rootless users can further

override fields in the config by creating a config file stored in the $HOME/.config/containers/containers.conf file.

If the CONTAINERS_CONF path environment variable is set, just this path will be used. This is primarily used for testing.

Fields specified in the containers.conf file override the default op? tions, as well as options in previously read containers.conf files.

storage.conf

The /etc/containers/storage.conf file is the default storage configura? tion file. Rootless users can override fields in the storage config by creating $HOME/.config/containers/storage.conf.

If the CONTAINERS_STORAGE_CONF path environment variable is set, this path is used for the storage.conf file rather than the default. This is primarily used for testing.

SEE ALSO

containers-storage.conf(5), containers-policy.json(5), containers-reg? istries.conf(5), tmpfiles.d(5)

engine Container containers.conf(5)