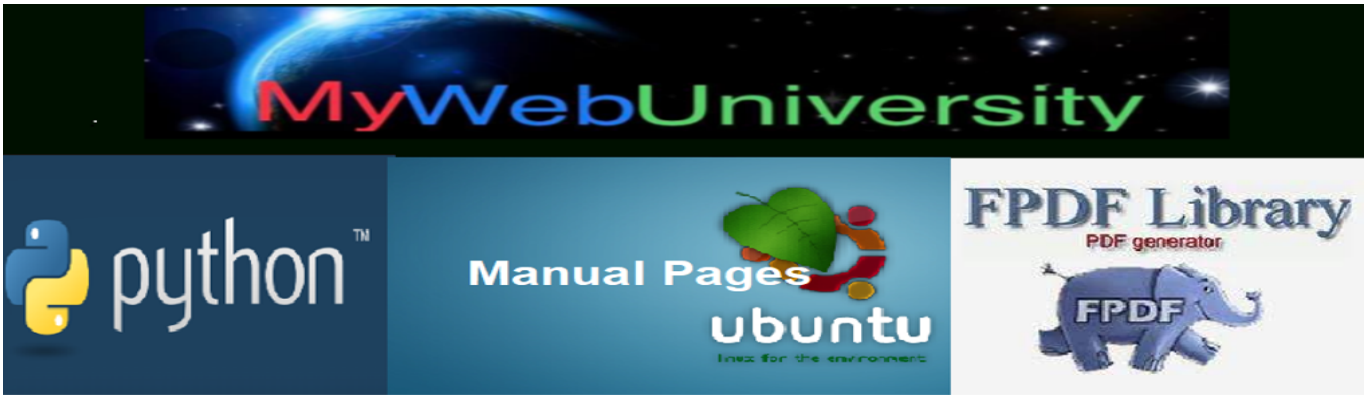

Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'containers.conf.5'

$ man containers.conf.5

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

NAME

containers.conf - The container engine configuration file specifies default configuration options and command-line flags for container engines.

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 versioned.

Container engines read the /usr/share/containers/containers.conf and /etc/containers/con? tainers.conf, and /etc/containers/containers.conf.d/.conf files if they exist. When run? ning in rootless mode, they also read $HOME/.config/containers/containers.conf and $HOME/.config/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 configuring the environment.

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

? 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 encoding of the config?

uration file. Every option is nested under its table. No bare options are used. The for?

mat 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 pertaining to the OCI runtime that can 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".

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 Cre?

ate 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 = [
```

"AUDIT_WRITE",

"CHOWN",

"DAC_OVERRIDE",

"FOWNER",

"FSETID",

"KILL",

"MKNOD",

"NET_BIND_SERVICE",

"NET_RAW",

"SETGID",

"SETPCAP",

"SETUID",

"SYS_CHROOT",

]

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:permissions'.

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 inside of the con?tainer.

dns_servers=[]

A list of dns servers to override the DNS configuration passed to the container. The spe?cial value ?none? can be specified to disable creation of /etc/resolv.conf in the con?tainer.

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

Environment variable list for the container process, used for passing environment vari?

ables to the container.

env_host=false

Pass all host environment variables into the container.

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 processes.

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

Path to the container-init binary, which forwards signals and reaps processes within con?

tainers. Note that the container-init binary will only be used when the --init for pod?

man-create and podman-run is set.

ipcns="private"

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

private Create private IPC Namespace for the container.

host Share host IPC Namespace with the container.

keyring=true

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

tainer.

label=true

Indicates whether the container engine uses MAC(SELinux) container separation via label?

ing. This option is ignored on disabled systems.

log_driver="k8s-file"

Logging driver for the container. Available options: k8s-file and journald.

log_size_max=-1

Maximum size allowed for the container's log file. Negative numbers indicate that no size

limit is imposed. If it is positive, it must be >= 8192 to match/exceed conmon's read buf?

fer. 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 creating 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 manage /etc/hosts, automatically adding the container's own IP address.

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 volume when the container is created instead of when it is started. If false, the container engine will not copy the content until the container is started. Setting it to true may have negative performance implications.

rootless_networking="slirp4netns"

Set type of networking rootless containers should use. Valid options are slirp4netns or cni.

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 image.

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.

userns_size=65536

Number of UIDs to allocate for the automatic container creation. UIDs are allocated from the ?container? UIDs listed in /etc/subuid & /etc/subgid.

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.

NETWORK TABLE

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

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 CNI network to attach pods to.

default_subnet="10.88.0.0/16"

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

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

Path to the directory where CNI configuration files are located.

volumes=[]

List of volumes. Specified as "directory-on-host:directory-in-container:options".

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

ENGINE TABLE

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

active_service=""

Name of destination for accessing the Podman service. See SERVICE DESTINATION 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 variable 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 sequence 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, Buil? dah). For example "http_proxy=internal.proxy.company.com". Note these environment vari? ables 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_logger="journald"

Default method to use when logging events. Valid values: file, journald, and none.

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

A is a list of directories which are used to search for helper binaries.

The default paths on Linux are: - /usr/local/libexec/podman - /usr/local/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/local/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, building container im? ages. By default images pulled and pushed match the format of the source image. Build? ing/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)

infra_command="/pause"

Command to run the infra container.

infra_image="k8s.gcr.io/pause:3.4.1"

Infra (pause) container image name 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.

lock_type="shm"

Specify the locking mechanism to use; valid values are "shm" and "file". Change the de? fault only if you are sure of what you are doing, in general "file" is useful only on platforms where cgo is not available for using the faster "shm" lock type. You may need to run "podman system renumber" after you change the lock type.

machine_enabled=false

Indicates if Podman is running inside a VM via Podman Machine. Podman uses this value to do extra setup around networking from the container inside the VM to to host.

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 con? tainers and pods that were created in the same namespace, and will create new containers and pods in that namespace. The default namespace is "", which corresponds to no name? space. 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.

Example "allow_host_loopback=true"

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 container 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.

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

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

? missing: attempt to pull the latest image from the registries listed in reg? istries.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 reg? istries.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 locally.

remote = false Indicates whether the application should be running in remote mode. This flag modifies the --remote option on container engines. Setting the flag to true will de? fault 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"]

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

runtime_supports_kvm=["kata"]

The list of OCI runtimes that support running containers with KVM separation.

runtime_supports_nocgroups=["crun"]

The list of OCI runtimes that support running containers without CGroups.

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 containers.

stop_timeout=10

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

tmp_dir="/run/libpod"

The path to a temporary directory to store per-boot container. Must be a tmpfs (wiped af?

ter reboot).

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

Directory where named volumes will be created in using the default volume driver. By de?

fault this will be configured relative to where containers/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 ownership to the primary

uid/gid of the container.

SERVICE DESTINATION TABLE

The service_destinations table contains configuration options used to set up remote con?

nections to the podman service for the podman API.

[service_destinations.{name}] URI to access the Podman service uri="ssh://user@produc?

tion.example.com/run/user/1001/podman/podman.sock"

Example URIs:

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

? rootless remote - ssh://user@engineering.lab.company.com/run/user/1000/pod?

man/podman.sock

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

? rootfull remote - ssh://root@10.10.1.136:22/run/podman/podman.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).

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="testing"

Default image used when creating a new VM using podman machine init. Options: testing, stable, next, or a custom path or download URL to an image

memory=2048

Memory in MB a machine is created with.

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/containers.conf to specify their own configuration. Rootless users can further override fields in the config by creating a config file stored in the $HOME/.con? fig/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 options, as well as op? tions in previously read containers.conf files.

storage.conf

The /etc/containers/storage.conf file is the default storage configuration file. Rootless users can override fields in the storage config by creating $HOME/.config/containers/stor? age.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-registries.conf(5)