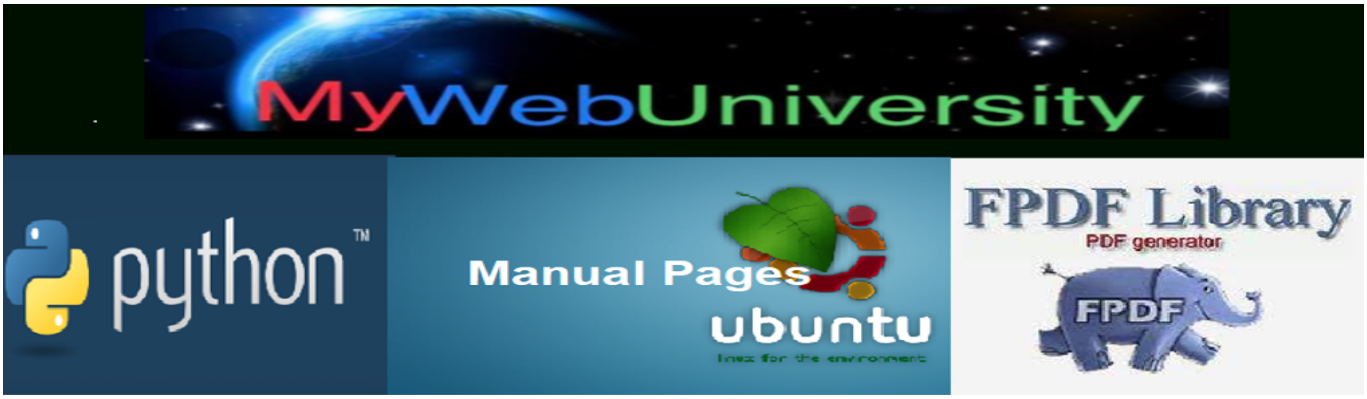

Rocky Enterprise Linux 9.2 Manual Pages on command 'podman-container-restore.1'

$ man podman-container-restore.1

podman-container-restore(1)() podman-container-restore(1)()

NAME

podman-container-restore - Restores one or more containers from a checkpoint

SYNOPSIS

podman container restore [options] container [container ...]

DESCRIPTION

podman container restore restores a container from a checkpoint. The container IDs or names are used as input.

OPTIONS

--all, -a

Restore all checkpointed containers.

The default is false.

IMPORTANT: This OPTION does not need a container name or ID as input argument.

--keep, -k

Keep all temporary log and statistics files created by CRIU during checkpointing as well as restoring. These files are not deleted if restoring fails for further debugging. If restoring succeeds these files are theoretically not needed, but if these files are needed Podman can keep the files for further analysis. This includes the checkpoint directory with all files created during checkpointing. The size required by the checkpoint directory is roughly the same as the amount of memory required by the processes in the checkpointed container.

Without the --keep, -k option the checkpoint will be consumed and cannot be used again.

The default is false.

--latest, -l

Instead of providing the container ID or name, use the last created container. If other tools than Podman are used to run containers such as CRI-O, the last started container could be from either tool.

The default is false.

IMPORTANT: This OPTION is not available with the remote Podman client. This OPTION does not need a container name or ID as input argument.

--ignore-rootfs

If a container is restored from a checkpoint tar.gz file it is possible that it also con? tains all root file-system changes. With --ignore-rootfs it is possible to explicitly dis? able applying these root file-system changes to the restored container.

The default is false.

IMPORTANT: This OPTION is only available in combination with --import, -i.

--ignore-static-ip

If the container was started with --ip the restored container also tries to use that IP address and restore fails if that IP address is already in use. This can happen, if a con? tainer is restored multiple times from an exported checkpoint with --name, -n.

Using --ignore-static-ip tells Podman to ignore the IP address if it was configured with --ip during container creation.

The default is false.

--ignore-static-mac

If the container was started with --mac-address the restored container also tries to use that MAC address and restore fails if that MAC address is already in use. This can happen, if a container is restored multiple times from an exported checkpoint with --name, -n.

Using --ignore-static-mac tells Podman to ignore the MAC address if it was configured with --mac-address during container creation.

The default is false.

--ignore-volumes

This option must be used in combination with the --import, -i option. When restoring con? tainers from a checkpoint tar.gz file with this option, the content of associated volumes will not be restored.

The default is false.

--import, -i=file

Import a checkpoint tar.gz file, which was exported by Podman. This can be used to import a checkpointed container from another host.

IMPORTANT: This OPTION does not need a container name or ID as input argument.

--import-previous=file

Import a pre-checkpoint tar.gz file which was exported by Podman. This option must be used with -i or --import. It only works on runc 1.0-rc3 or higher.

--name, -n=name

If a container is restored from a checkpoint tar.gz file it is possible to rename it with --name, -n. This way it is possible to restore a container from a checkpoint multiple times with different names.

If the --name, -n option is used, Podman will not attempt to assign the same IP address to the container it was using before checkpointing as each IP address can only be used once and the restored container will have another IP address. This also means that --name, -n cannot be used in combination with --tcp-established.

IMPORTANT: This OPTION is only available in combination with --import, -i.

--pod=name

Restore a container into the pod name. The destination pod for this restore has to have the same namespaces shared as the pod this container was checkpointed from (see **podman pod create --share. *IMPORTANT: This OPTION is only available in combination with --im? port, -i.*

This option requires at least CRIU 3.16.

--publish, -p=port

Replaces the ports that the container publishes, as configured during the initial con? tainer start, with a new set of port forwarding rules.

For more details please see podman run --publish.

--tcp-established

Restore a container with established TCP connections. If the checkpoint image contains es? tablished TCP connections, this option is required during restore. If the checkpoint im? age does not contain established TCP connections this option is ignored. Defaults to not restoring containers with established TCP connections.

The default is false.

EXAMPLE

Restores the container "mywebserver".

podman container restore mywebserver

Import a checkpoint file and a pre-checkpoint file.

podman container restore --import-previous pre-checkpoint.tar.gz --import checkpoint.tar.gz

Remove the container "mywebserver". Make a checkpoint of the container and export it. Re?

store the container with other port ranges from the exported file.

$ podman run --rm -p 2345:80 -d webserver

podman container checkpoint -l --export=dump.tar

podman container restore -p 5432:8080 --import=dump.tar

SEE ALSO

podman(1), podman-container-checkpoint(1), podman-run(1), podman-pod-create(1)

HISTORY

September 2018, Originally compiled by Adrian Reber areber@redhat.com ?mailto:areber@red?

hat.com?