



**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 'podman-container-checkpoint.1'***

### ***\$ man podman-container-checkpoint.1***

podman-container-checkpoint(1)()                      podman-container-checkpoint(1)()

#### NAME

podman-container-checkpoint - Checkpoints one or more running containers

#### SYNOPSIS

podman container checkpoint [options] container [container ...]

#### DESCRIPTION

podman container checkpoint checkpoints all the processes in one or more containers. A container can be restored from a checkpoint with podman-container-restore. The container IDs or names are used as input.

#### OPTIONS

--all, -a

Checkpoint all running containers.

The default is false.

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

--compress, -c=zstd | none | gzip

Specify the compression algorithm used for the checkpoint archive created with the --export, -e OPTION. Possible algorithms are zstd, none and gzip.

One possible reason to use none is to enable faster creation of checkpoint archives. Not compressing the checkpoint archive can result in faster checkpoint archive creation.

The default is zstd.

--export, -e=archive

Export the checkpoint to a tar.gz file. The exported checkpoint can be used to import the container on another system and thus enabling container live migration. This checkpoint

archive also includes all changes to the container's root file-system, if not explicitly disabled using `--ignore-rootfs`.

#### `--ignore-rootfs`

If a checkpoint is exported to a tar.gz file it is possible with the help of `--ignore-rootfs` to explicitly disable including changes to the root file-system into the checkpoint archive file.

The default is false.

**\*IMPORTANT:** This OPTION only works in combination with `--export, -e`.

#### `--ignore-volumes`

This OPTION must be used in combination with the `--export, -e` OPTION. When this OPTION is specified, the content of volumes associated with the container will not be included into the checkpoint tar.gz file.

The default is false.

#### `--keep, -k`

Keep all temporary log and statistics files created by CRIU during checkpointing. These files are not deleted if checkpointing fails for further debugging. If checkpointing succeeds these files are theoretically not needed, but if these files are needed Podman can keep the files for further analysis.

The default is false.

#### `--latest, -l`

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

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.

#### `--leave-running, -R`

Leave the container running after checkpointing instead of stopping it.

The default is false.

#### `--pre-checkpoint, -P`

Dump the container's memory information only, leaving the container running. Later operations will supersede prior dumps. It only works on runc 1.0-rc3 or higher.

The default is false.

--tcp-established

Checkpoint a container with established TCP connections. If the checkpoint image contains established TCP connections, this OPTION is required during restore. Defaults to not checkpointing containers with established TCP connections.

The default is false.

--with-previous

Check out the container with previous criu image files in pre-dump. It only works on runc 1.0-rc3 or higher.

The default is false.

\*IMPORTANT: This OPTION is not available with --pre-checkpoint\*.

## EXAMPLES

Make a checkpoint for the container "mywebserver".

```
# podman container checkpoint mywebserver
```

Dumps the container's memory information of the latest container into an archive.

```
# podman container checkpoint -P -e pre-checkpoint.tar.gz -l
```

Keep the container's memory information from an older dump and add the new container's memory information.

```
# podman container checkpoint --with-previous -e checkpoint.tar.gz -l
```

Dump the container's memory information of the latest container into an archive with the specified compress method.

```
# podman container checkpoint -l --compress=none --export=dump.tar
```

```
# podman container checkpoint -l --compress=gzip --export=dump.tar.gz
```

## SEE ALSO

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

## HISTORY

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

podman-container-checkpoint(1)()