



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

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

\$ man podman-container-ps.1

podman-ps(1) General Commands Manual podman-ps(1)

NAME

podman-ps - Prints out information about containers

SYNOPSIS

podman ps [options]

podman container ps [options]

podman container list [options]

podman container ls [options]

DESCRIPTION

podman ps lists the running containers on the system. Use the --all

flag to view all the containers information. By default it lists:

? container id

? the name of the image the container is using

? the COMMAND the container is executing

? the time the container was created

? the status of the container

? port mappings the container is using

? alternative names for the container

OPTIONS

--all, -a

Show all the containers created by Podman, default is only running containers.

Note: Podman shares containers storage with other tools such as Buildah and CRI-O. In some cases these external containers might also exist in the same storage. Use the --external option to see these external containers. External containers show the 'storage' status.

--external

Display external containers that are not controlled by Podman but are stored in containers storage. These external containers are generally created via other container technology such as Buildah or CRI-O and may depend on the same container images that Podman is also using. External containers are denoted with either a 'buildah' or 'storage' in the COMMAND and STATUS column of the ps output.

--filter, -f

Filter what containers are shown in the output. Multiple filters can be given with multiple uses of the --filter flag. Filters with the same key work inclusive with the only exception being label which is exclusive. Filters with different keys always work exclusive.

Valid filters are listed below:

????????????????????????????????????????????????????????

?Filter ? Description ?

????????????????????????????????????????????????????????

?id ? [ID] Container's ID (accepts ?

? ? regex) ?

????????????????????????????????????????????????????????

?name ? [Name] Container's name (accepts ?

? ? cepts regex) ?

????????????????????????????????????????????????????????

?label ? [Key] or [Key=Value] Label ?

? ? assigned to a container ?

????????????????????????????????????????????????????????

?exited ? [Int] Container's exit code ?
????????????????????????????????????????????????????????
?status ? [Status] Container's status: ?
? ? 'created', 'exited', ?
? ? 'paused', 'running', 'un? ?
? ? known' ?
????????????????????????????????????????????????????????
?ancestor ? [ImageName] Image or descen? ?
? ? dant used to create container ?
? ? (accepts regex) ?
????????????????????????????????????????????????????????
?before ? [ID] or [Name] Containers ?
? ? created before this container ?
????????????????????????????????????????????????????????
?since ? [ID] or [Name] Containers ?
? ? created since this container ?
????????????????????????????????????????????????????????
?volume ? [VolumeName] or [Mountpoint? ?
? ? Destination] Volume mounted ?
? ? in container ?
????????????????????????????????????????????????????????
?health ? [Status] healthy or unhealthy ?
????????????????????????????????????????????????????????
?pod ? [Pod] name or full or partial ?
? ? ID of pod ?
????????????????????????????????????????????????????????
?network ? [Network] name or full ID of ?
? ? network ?
????????????????????????????????????????????????????????

--format=format

Pretty-print containers to JSON or using a Go template
Valid placeholders for the Go template are listed below:

????????????????????????????????????????????????????????

?Placeholder ? Description ?
????????????????????????????????????????????????????????????
?.ID ? Container ID ?
????????????????????????????????????????????????????????????
?.Image ? Image Name/ID ?
????????????????????????????????????????????????????????????
?.ImageID ? Image ID ?
????????????????????????????????????????????????????????????
?.Command ? Quoted command used ?
????????????????????????????????????????????????????????????
?.CreatedAt ? Creation time for container ?
????????????????????????????????????????????????????????????
?.RunningFor ? Time elapsed since con? ?
? ? tainer was started ?
????????????????????????????????????????????????????????????
?.Status ? Status of container ?
????????????????????????????????????????????????????????????
?.Pod ? Pod the container is asso? ?
? ? ciated with ?
????????????????????????????????????????????????????????????
?.Ports ? Exposed ports ?
????????????????????????????????????????????????????????????
?.Size ? Size of container ?
????????????????????????????????????????????????????????????
?.Names ? Name of container ?
????????????????????????????????????????????????????????????
?.Networks ? Show all networks connected ?
? ? to the container ?
????????????????????????????????????????????????????????????
?.Labels ? All the labels assigned to ?
? ? the container ?
????????????????????????????????????????????????????????????
?.Mounts ? Volumes mounted in the con? ?

? ? tainer ?

????????????????????????????????????????????????????????????

--help, -h

Print usage statement

--last, -n

Print the n last created containers (all states)

--latest, -l

Show the latest container created (all states) (This option is not available with the remote Podman client, including Mac and Windows (excluding WSL2) machines)

--namespace, --ns

Display namespace information

--no-trunc

Do not truncate the output (default false).

--noheading

Omit the table headings from the listing of containers.

--pod, -p

Display the pods the containers are associated with

--quiet, -q

Print the numeric IDs of the containers only

--size, -s

Display the total file size

--sort=created

Sort by command, created, id, image, names, runningfor, size, or status", Note: Choosing size will sort by size of rootFs, not alphabetically like the rest of the options

--sync

Force a sync of container state with the OCI runtime. In some cases, a container's state in the runtime can become out of sync with Podman's state. This will update Podman's state based on what the OCI runtime reports. Forcibly syncing is much slower, but can resolve inconsistent state issues.

--watch, -w

Refresh the output with current containers on an interval in seconds.

EXAMPLES

```
$ podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
02f65160e14ca	redis:alpine	"redis-server"	19 hours ago	Exited (-1)	19 hours ago	6379/tcp
k8s_podsandbox1-redis_podsandbox1_redhat.test.crio_redhat-test-crio_0						
69ed779d8ef9f	redis:alpine	"redis-server"	25 hours ago	Created		6379/tcp
k8s_container1_podsandbox1_redhat.test.crio_redhat-test-crio_1						

```
$ podman ps -a -s
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	SIZE
02f65160e14ca	redis:alpine	"redis-server"	20 hours ago	Exited (-1)	20 hours ago	6379/tcp	
k8s_podsandbox1-redis_podsandbox1_redhat.test.crio_redhat-test-crio_0							27.49 MB
69ed779d8ef9f	redis:alpine	"redis-server"	25 hours ago	Created		6379/tcp	
k8s_container1_podsandbox1_redhat.test.crio_redhat-test-crio_1							27.49 MB

```
$ podman ps -a --format "{{.ID}} {{.Image}} {{.Labels}} {{.Mounts}}"
```

```
02f65160e14ca redis:alpine tier=backend proc,tmpfs,devpts,shm,mqueue,sysfs,cgroup,/var/run/,/var/run/
```

```
69ed779d8ef9f redis:alpine batch=no,type=small proc,tmpfs,devpts,shm,mqueue,sysfs,cgroup,/var/run/,/var/run/
```

```
$ podman ps --ns -a
```

CONTAINER ID	NAMES	PID	CGROUP	IPC	MNT
NET	PIDNS	USER	UTS		
3557d882a82e3	k8s_container2_podsandbox1_redhat.test.crio_redhat-test-crio_1			29910	4026531835
4026532585	4026532593	4026532508	4026532595	4026531837	4026532594
09564cdae0bec	k8s_container1_podsandbox1_redhat.test.crio_redhat-test-crio_1			29851	4026531835
4026532585	4026532590	4026532508	4026532592	4026531837	4026532591
a31ebbee9cee7	k8s_podsandbox1-redis_podsandbox1_redhat.test.crio_redhat-test-crio_0			29717	4026531835
4026532585	4026532587	4026532508	4026532589	4026531837	4026532588

```
$ podman ps -a --size --sort names
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
69ed779d8ef9f	redis:alpine	"redis-server"	25 hours ago	Created		6379/tcp
k8s_container1_podsandbox1_redhat.test.crio_redhat-test-crio_1						
02f65160e14ca	redis:alpine	"redis-server"	19 hours ago	Exited (-1)	19 hours ago	6379/tcp
k8s_podsandbox1-redis_podsandbox1_redhat.test.crio_redhat-test-crio_0						

```
$ podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4089df24d4f3	docker.io/library/centos:latest	/bin/bash	2 minutes ago	Up 2 minutes	0.0.0.0:80->8080/tcp, 0.0.0.0:2000-2006->2000-2006/tcp	manyports
92f58933c28c	docker.io/library/centos:latest	/bin/bash	3 minutes ago	Up 3 minutes	192.168.99.100:1000-1006->1000-1006/tcp	zen_sanderson

```
$ podman ps --external -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
69ed779d8ef9f	redis:alpine	"redis-server"	25 hours ago	Created	6379/tcp	k8s_container1_podsandbox1_redhat.test.crio_redhat-test-crio_1
38a8a78596f9	docker.io/library/busybox:latest	buildah	2 hours ago	storage		busybox-working-container
fd7b786b5c32	docker.io/library/alpine:latest	buildah	2 hours ago	storage		alpine-working-container
f78620804e00	scratch	buildah	2 hours ago	storage		working-container

```
ps
```

Print a list of containers

SEE ALSO

podman(1), buildah(1), crio(8)

HISTORY

August 2017, Originally compiled by Urvashi Mohnani umohnani@redhat.com

[?mailto:umohnani@redhat.com?](mailto:umohnani@redhat.com)

podman-ps(1)