

**NAME****runc** – Open Container Initiative runtime**SYNOPSIS****runc** [*global-option* ...] *command* [*command-option* ...] [*argument* ...]**DESCRIPTION**

**runc** is a command line client for running applications packaged according to the Open Container Initiative (OCI) format and is a compliant implementation of the Open Container Initiative specification.

**runc** integrates well with existing process supervisors to provide a production container runtime environment for applications. It can be used with your existing process monitoring tools and the container will be spawned as a direct child of the process supervisor.

Containers are configured using bundles. A bundle for a container is a directory that includes a specification file named *config.json* and a root filesystem. The root filesystem contains the contents of the container.

To run a new instance of a container:

```
# runc run [ -b bundle ] container-id
```

Where *container-id* is your name for the instance of the container that you are starting. The name you provide for the container instance must be unique on your host.

Providing the bundle directory using **-b** is optional. The default value for *bundle* is the current directory.

**COMMANDS**

**checkpoint** : Checkpoint a running container. See **runc-checkpoint(8)**.

**create** : Create a container. See **runc-create(8)**.

**delete** : Delete any resources held by the container often used with detached containers. See **runc-delete(8)**.

**events** : Display container events such as OOM notifications, cpu, memory, IO and network stats. See **runc-events(8)**.

**exec** : Execute a new process inside the container. See **runc-exec(8)**.

**init** : Initialize the namespaces and launch the container init process. This command is not supposed to be used directly.

**kill** : Send a specified signal to the container's init process. See **runc-kill(8)**.

**list** : List containers started by runc with the given **--root**. See **runc-list(8)**.

**pause** : Suspend all processes inside the container. See **runc-pause(8)**.

**ps** : Show processes running inside the container. See **runc-ps(8)**.

**restore** : Restore a container from a previous checkpoint. See **runc-restore(8)**.

**resume** : Resume all processes that have been previously paused. See **runc-resume(8)**.

**run** : Create and start a container. See **runc-run(8)**.

**spec** : Create a new specification file (*config.json*). See **runc-spec(8)**.

**start** : Start a container previously created by **runc create**. See **runc-start(8)**.

**state** : Show the container state. See **runc-state(8)**.

**update** : Update container resource constraints. See **runc-update(8)**.

**help, h** : Show a list of commands or help for a particular command.

## GLOBAL OPTIONS

These options can be used with any command, and must precede the **command**.

**--debug** : Enable debug logging.

**--log path** : Set the log destination to *path*. The default is to log to stderr.

**--log-format text|json** : Set the log format (default is **text**).

**--root path** : Set the root directory to store containers' state. The *path* should be located on tmpfs. Default is */run/runc*, or *\$XDG\_RUNTIME\_DIR/runc* for rootless containers.

**--criu path** : Set the path to the **criu(8)** binary used for checkpoint and restore. Default is **criu**.

**--systemd-cgroup** : Enable systemd cgroup support. If this is set, the container spec (*config.json*) is expected to have **cgroupsPath** value in the *slice:prefix:name* form (e.g. **system.slice:runc:434234**).

**--rootless true|false|auto** : Enable or disable rootless mode. Default is **auto**, meaning to auto-detect whether rootless should be enabled.

**--help|-h** : Show help.

**--version|-v** : Show version.

runc (8"")

runc (8"")

## SEE ALSO

**runc-checkpoint(8), runc-create(8), runc-delete(8), runc-events(8), runc-exec(8), runc-kill(8),  
runc-list(8), runc-pause(8), runc-ps(8), runc-restore(8), runc-resume(8), runc-run(8),  
runc-spec(8), runc-start(8), runc-state(8), runc-update(8).**