Rocky Enterprise Linux 9.2 Manual Pages on command 'oci-hooks.5'

$ man oci-hooks.5

oci-hooks(5) Configuration oci-hooks(5)

W. Trevor King MAY 2018

NAME

oci-hooks - OCI hooks configuration directories

SYNOPSIS

/usr/share/containers/oci/hooks.d/*.json

DESCRIPTION

Provides a way for users to configure the intended hooks for Open Container Initiative containers so they will only be executed for containers that need their functionality, and then only for the stages where they're needed.

Directories

Hooks are configured with JSON files (ending with a .json extension) in a series of hook directories. The default directory is /usr/share/containers/oci/hooks.d, but tools con? suming this format may change that default, include additional directories, or provide their callers with ways to adjust the configuration directories.

If multiple directories are configured, a JSON filename in a preferred directory masks en? tries with the same filename in directories with lower precedence. For example, if a con? suming tool watches for hooks in /etc/containers/oci/hooks.d and /usr/share/contain? ers/oci/hooks.d (in order of decreasing precedence), then a hook definition in /etc/con? tainers/oci/hooks.d/01-my-hook.json will mask any definition in /usr/share/contain? ers/oci/hooks.d/01-my-hook.json.

Tools consuming this format may also opt to monitor the hook directories for changes, in which case they will notice additions, changes, and removals to JSON files without needing to be restarted or otherwise signaled. When the tool monitors multiple hooks directories, the precedence discussed in the previous paragraph still applies. For example, if a consuming tool watches for hooks in /etc/containers/oci/hooks.d and /usr/share/containers/oci/hooks.d (in order of decreasing precedence), then writing a new hook definition to /etc/containers/oci/hooks.d/01-my-hook.json will mask the hook previously loaded from /usr/share/containers/oci/hooks.d/01-my-hook.json. Subsequent changes to /usr/share/containers/oci/hooks.d/01-my-hook.json will have no effect on the consuming tool as long as /etc/containers/oci/hooks.d/01-my-hook.json exists. Removing /etc/containers/oci/hooks.d/01-my-hook.json will reload the hook from /usr/share/containers/oci/hooks.d/01-my-hook.json.

Hooks are injected in the order obtained by sorting the JSON file names, after converting them to lower case, based on their Unicode code points. For example, a matching hook defined in 01-my-hook.json would be injected before matching hooks defined in 02-another-hook.json and 01-UPPERCASE.json. It is strongly recommended to make the sort order unambiguous depending on an ASCII-only prefix (like the 01/02 above).

Each JSON file should contain an object with one of the following schemas.

1.0.0 Hook Schema

version (required string)

Sets the hook-definition version. For this schema version, the value be 1.0.0.

hook (required object)

The hook to inject, with the hook-entry schema defined by the 1.0.1 OCI Runtime Specification.

when (required object)

Conditions under which the hook is injected. The following properties can be specified, and at least one must be specified:

? always (optional boolean)

If set true, this condition matches.

? annotations (optional object)

If all annotations key/value pairs match a key/value pair from the configured annotations, this condition matches.

Both keys and values must be POSIX extended regular expressions.

? commands (optional array of strings)

If the configured process.args[0] matches an entry, this condition matches.

Entries must be POSIX extended regular expressions.

? hasBindMounts (optional boolean)

If hasBindMounts is true and the caller requested host-to-container bind mounts, this condition matches.

stages (required array of strings)

Stages when the hook must be injected. Entries must be chosen from the 1.0.1 OCI Run?time Specification hook stages or from extension stages supported by the package consumer.

If all of the conditions set in when match, then the hook must be injected for the stages set in stages.

0.1.0 Hook Schema

hook (required string)

Sets path in the injected hook.

arguments (optional array of strings)

Additional arguments to pass to the hook. The injected hook's args is hook with argu?ments appended.

stages (required array of strings)

Stages when the hook must be injected. stage is an allowed synonym for this property, but you must not set both stages and stage. Entries must be chosen from the 1.0.1 OCI Runtime Specification hook stages or from extension stages supported by the package con?sumer.

cmds (optional array of strings)

The hook must be injected if the configured process.args[0] matches an entry. cmd is an allowed synonym for this property, but you must not set both cmds and cmd. Entries must be POSIX extended regular expressions.

annotations (optional array of strings)

The hook must be injected if an annotations entry matches a value from the configured annotations. annotation is an allowed synonym for this property, but you must not set both annotations and annotation. Entries must be POSIX extended regular expressions.

hasbindmounts (optional boolean)

The hook must be injected if hasBindMounts is true and the caller requested host-to-con?tainer bind mounts.

EXAMPLE

1.0.0 Hook Schema

The following configuration injects oci-systemd-hook in the pre-start and post-stop stages if process.args[0] ends with /init or /systemd:

```
$ cat /etc/containers/oci/hooks.d/oci-systemd-hook.json
{
  "version": "1.0.0",
  "hook": {
    "path": "/usr/libexec/oci/hooks.d/oci-systemd-hook"
  },
  "when": {
    "commands": [".*/init$" , ".*/systemd$"]
  },
  "stages": ["prestart", "poststop"]
}
```

The following example injects oci-umount --debug in the pre-start stage if the container is configured to bind-mount host directories into the container.

```
$ cat /etc/containers/oci/hooks.d/oci-umount.json
{
  "version": "1.0.0",
  "hook": {
    "path": "/usr/libexec/oci/hooks.d/oci-umount",
    "args": ["oci-umount", "--debug"],
  },
  "when": {
    "hasBindMounts": true
  },
  "stages": ["prestart"]
}
```

The following example injects nvidia-container-runtime-hook prestart with particular envi?ronment variables in the pre-start stage if the container is configured with an annota?tions entry whose key matches ^com\.example\.department$ and whose value matches .*fluid-dynamics.*.

```
$ cat /etc/containers/oci/hooks.d/nvidia.json
{
  "version": "1.0.0",
  "hook": {
    "path": "/usr/sbin/nvidia-container-runtime-hook",
    "args": ["nvidia-container-runtime-hook", "prestart"],
    "env": [
      "NVIDIA_REQUIRE_CUDA=cuda>=9.1",
      "NVIDIA_VISIBLE_DEVICES=GPU-fef8089b"
    ]
  },
  "when": {
    "annotations": {
      "^com\\.example\\.department$": ".*fluid-dynamics$"
    }
  },
  "stages": ["prestart"]
}
```

0.1.0 Hook Schema

The following configuration injects oci-systemd-hook in the pre-start and post-stop stages if process.args[0] ends with /init or /systemd:

```
$ cat /etc/containers/oci/hooks.d/oci-systemd-hook.json
{
  "cmds": [".*/init$" , ".*/systemd$"],
  "hook": "/usr/libexec/oci/hooks.d/oci-systemd-hook",
  "stages": ["prestart", "poststop"]
}
```

The following example injects oci-umount --debug in the pre-start stage if the container is configured to bind-mount host directories into the container.

```
$ cat /etc/containers/oci/hooks.d/oci-umount.json
{
  "hook": "/usr/libexec/oci/hooks.d/oci-umount",
  "arguments": ["--debug"],
  "hasbindmounts": true,
  "stages": ["prestart"]
}
```

The following example injects nvidia-container-runtime-hook prestart in the pre-start stage if the container is configured with an annotations entry whose value matches .*fluid-dynamics.*.

```
$ cat /etc/containers/oci/hooks.d/osystemd-hook.json
{
  "hook": "/usr/sbin/nvidia-container-runtime-hook",
  "arguments": ["prestart"],
  "annotations: [".*fluid-dynamics.*"],
  "stages": ["prestart"]
}
```

SEE ALSO

oci-systemd-hook(1), oci-umount(1), locale(7)

? OCI Runtime Specification, 1.0.1, POSIX-platform hooks

? OCI Runtime Specification, 1.0.1, process

? POSIX extended regular expressions (EREs)

Hooks OCI oci-hooks(5)