Docker Remote API

Estimated reading time: 13 minutes

Docker's Remote API uses an open schema model. In this model, unknown properties in incoming messages are ignored. Client applications need to take this behavior into account to ensure they do not break when talking to newer Docker daemons.

The API tends to be REST, but for some complex commands, like attach or pull, the HTTP connection is hijacked to transport STDOUT, STDIN, and STDERR.

By default the Docker daemon listens on `unix:///var/run/docker.sock` and the client must have `root` access to interact with the daemon. If a group named `docker` exists on your system, `docker` applies ownership of the socket to the group.

To connect to the Docker daemon with cURL you need to use cURL 7.40 or later, as these versions have the `--unix-socket` flag available. To run `curl` against the daemon on the default socket, use the following:

When using cUrl 7.50 or later:

```
$ curl --unix-socket /var/run/docker.sock http://localhost/containers/
```

When using cURL 7.40, `localhost` must be omitted:

```
$ curl --unix-socket /var/run/docker.sock http://containers/json
```

If you have bound the Docker daemon to a different socket path or TCP port, you would reference that in your cURL rather than the default.

The current version of the API is v1.24 which means calling `/info` is the same as calling `/v1.24/info`. To call an older version of the API use `/v1.23/info`. If a newer daemon is installed, new properties may be returned even when calling older versions of the API.

Use the table below to find the API version for a Docker version:

Refer to the GitHub repository for older releases.

Authentication

Authentication configuration is handled client side, so the client has to send the `authConfig` as a `POST` in `/images/(name)/push`. The `authConfig`, set as the `X-Registry-Auth` header, is currently a Base64 encoded (JSON) string with the following structure:

```
{"username": "string", "password": "string", "email": "string",
   "serveraddress" : "string", "auth": ""}
```

Callers should leave the `auth` empty. The `serveraddress` is a domain/ip without protocol. Throughout this structure, double quotes are required.

Using Docker Machine with the API

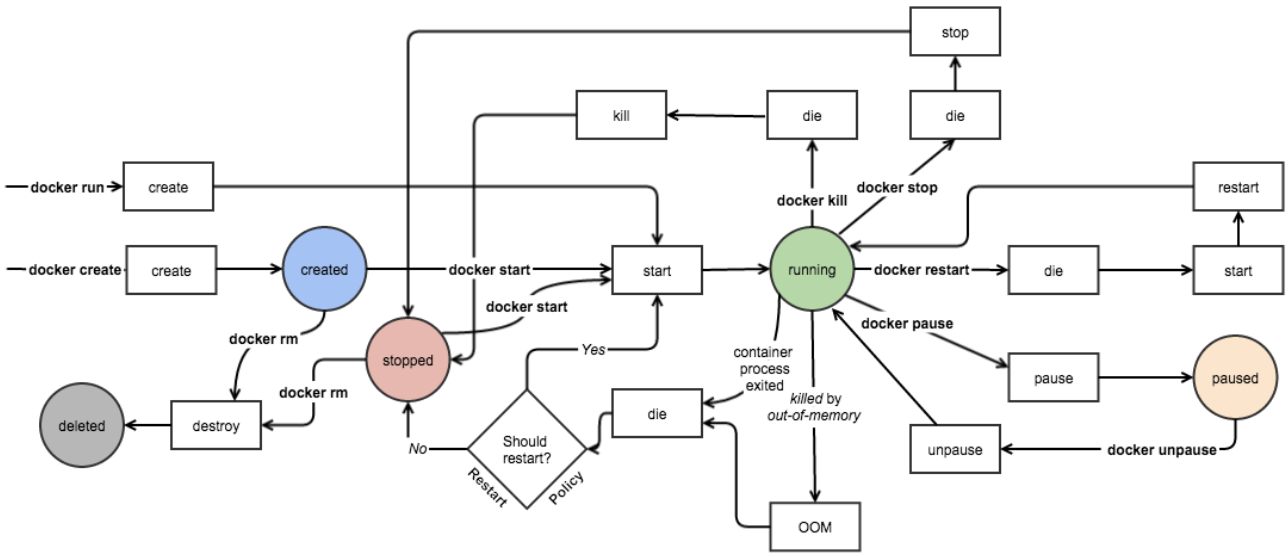
If you are using `docker-machine`, the Docker daemon is on a host that uses an encrypted TCP socket using TLS. This means, for Docker Machine users, you need to add extra parameters to `curl` or `wget` when making test API requests, for example:

```
curl --insecure \
     --cert $DOCKER_CERT_PATH/cert.pem \
     --key $DOCKER_CERT_PATH/key.pem \
     https://YOUR_VM_IP:2376/images/json

wget --no-check-certificate --certificate=$DOCKER_CERT_PATH/cert.pem \
     --private-key=$DOCKER_CERT_PATH/key.pem \
     https://YOUR_VM_IP:2376/images/json -O - -q
```

Docker Events

The following diagram depicts the container states accessible through the API.



Some container-related events are not affected by container state, so they are not included in this diagram. These events are:

- **export** emitted by `docker export`
- **exec_create** emitted by `docker exec`
- **exec_start** emitted by `docker exec` after **exec_create**
- **detach** emitted when client is detached from container process
- **exec_detach** emitted when client is detached from exec process

Running `docker rmi` emits an **untag** event when removing an image name. The `rmi` command may also emit **delete** events when images are deleted by ID directly or by deleting the last tag referring to the image.

> **Acknowledgment**: This diagram and the accompanying text were

> used with the permission of Matt Good and Gilder Labs. See Matt's original blog post Docker Events Explained.

Version history

This section lists each version from latest to oldest. Each listing includes a link to the full documentation set and the changes relevant in that release.

v1.24 API changes

Docker Remote API v1.24 documentation

- `POST /containers/create` now takes `StorageOpt` field.
- `GET /info` now returns `SecurityOptions` field, showing if `apparmor`, `seccomp`, or `selinux` is supported.
- `GET /info` no longer returns the `ExecutionDriver` property. This property was no longer used after integration with ContainerD in Docker 1.11.
- `GET /networks` now supports filtering by `label` and `driver`.
- `GET /containers/json` now supports filtering containers by `network` name or id.
- `POST /containers/create` now takes `IOMaximumBandwidth` and `IOMaximumIOps` fields. Windows daemon only.
- `POST /containers/create` now returns an HTTP 400 "bad parameter" message if no command is specified (instead of an HTTP 500 "server error")
- `GET /images/search` now takes a `filters` query parameter.
- `GET /events` now supports a `reload` event that is emitted when the daemon configuration is reloaded.
- `GET /events` now supports filtering by daemon name or ID.
- `GET /events` now supports a `detach` event that is emitted on detaching from container process.
- `GET /events` now supports an `exec_detach` event that is emitted on

detaching from exec process.

- `GET /images/json` now supports filters `since` and `before`.
- `POST /containers/(id or name)/start` no longer accepts a `HostConfig`.
- `POST /images/(name)/tag` no longer has a `force` query parameter.
- `GET /images/search` now supports maximum returned search results `limit`.
- `POST /containers/{name:.*}/copy` is now removed and errors out starting from this API version.
- API errors are now returned as JSON instead of plain text.
- `POST /containers/create` and `POST /containers/(id)/start` allow you to configure kernel parameters (sysctls) for use in the container.
- `POST /containers/<container ID>/exec` and `POST /exec/<exec ID>/start` no longer expects a “Container” field to be present. This property was not used and is no longer sent by the docker client.
- `POST /containers/create/` now validates the hostname (should be a valid RFC 1123 hostname).
- `POST /containers/create/` `HostConfig.PidMode` field now accepts `container:<name|id>`, to have the container join the PID namespace of an existing container.

v1.23 API changes

Docker Remote API v1.23 documentation

- `GET /containers/json` returns the state of the container, one of `created`, `restarting`, `running`, `paused`, `exited` or `dead`.
- `GET /containers/json` returns the mount points for the container.
- `GET /networks/(name)` now returns an `Internal` field showing whether the network is internal or not.
- `GET /networks/(name)` now returns an `EnableIPv6` field showing whether the network has ipv6 enabled or not.

- `POST /containers/(name)/update` now supports updating container's restart policy.
- `POST /networks/create` now supports enabling ipv6 on the network by setting the `EnableIPv6` field (doing this with a label will no longer work).
- `GET /info` now returns `CgroupDriver` field showing what cgroup driver the daemon is using; `cgroupfs` or `systemd`.
- `GET /info` now returns `KernelMemory` field, showing if "kernel memory limit" is supported.
- `POST /containers/create` now takes `PidsLimit` field, if the kernel is >= 4.3 and the pids cgroup is supported.
- `GET /containers/(id or name)/stats` now returns `pids_stats`, if the kernel is >= 4.3 and the pids cgroup is supported.
- `POST /containers/create` now allows you to override usernamespaces remapping and use privileged options for the container.
- `POST /containers/create` now allows specifying `nocopy` for named volumes, which disables automatic copying from the container path to the volume.
- `POST /auth` now returns an `IdentityToken` when supported by a registry.
- `POST /containers/create` with both `Hostname` and `Domainname` fields specified will result in the container's hostname being set to `Hostname`, rather than `Hostname.Domainname`.
- `GET /volumes` now supports more filters, new added filters are `name` and `driver`.
- `GET /containers/(id or name)/logs` now accepts a `details` query parameter to stream the extra attributes that were provided to the containers `LogOpts`, such as environment variables and labels, with the logs.
- `POST /images/load` now returns progress information as a JSON stream, and has a `quiet` query parameter to suppress progress details.

v1.22 API changes

Docker Remote API v1.22 documentation

- `POST /container/(name)/update` updates the resources of a container.
- `GET /containers/json` supports filter `isolation` on Windows.
- `GET /containers/json` now returns the list of networks of containers.
- `GET /info` Now returns `Architecture` and `OSType` fields, providing information about the host architecture and operating system type that the daemon runs on.
- `GET /networks/(name)` now returns a `Name` field for each container attached to the network.
- `GET /version` now returns the `BuildTime` field in RFC3339Nano format to make it consistent with other date/time values returned by the API.
- `AuthConfig` now supports a `registrytoken` for token based authentication
- `POST /containers/create` now has a 4M minimum value limit for `HostConfig.KernelMemory`
- Pushes initiated with `POST /images/(name)/push` and pulls initiated with `POST /images/create` will be cancelled if the HTTP connection making the API request is closed before the push or pull completes.
- `POST /containers/create` now allows you to set a read/write rate limit for a device (in bytes per second or IO per second).
- `GET /networks` now supports filtering by `name`, `id` and `type`.
- `POST /containers/create` now allows you to set the static IPv4 and/or IPv6 address for the container.
- `POST /networks/(id)/connect` now allows you to set the static IPv4 and/or IPv6 address for the container.
- `GET /info` now includes the number of containers running, stopped, and paused.

- `POST /networks/create` now supports restricting external access to the network by setting the `Internal` field.
- `POST /networks/(id)/disconnect` now includes a `Force` option to forcefully disconnect a container from network
- `GET /containers/(id)/json` now returns the `NetworkID` of containers.
- `POST /networks/create` Now supports an options field in the IPAM config that provides options for custom IPAM plugins.
- `GET /networks/{network-id}` Now returns IPAM config options for custom IPAM plugins if any are available.
- `GET /networks/<network-id>` now returns subnets info for user-defined networks.
- `GET /info` can now return a `SystemStatus` field useful for returning additional information about applications that are built on top of engine.

v1.21 API changes

Docker Remote API v1.21 documentation

- `GET /volumes` lists volumes from all volume drivers.
- `POST /volumes/create` to create a volume.
- `GET /volumes/(name)` get low-level information about a volume.
- `DELETE /volumes/(name)` remove a volume with the specified name.
- `VolumeDriver` was moved from `config` to `HostConfig` to make the configuration portable.
- `GET /images/(name)/json` now returns information about an image's `RepoTags` and `RepoDigests`.
- The `config` option now accepts the field `StopSignal`, which specifies the signal to use to kill a container.
- `GET /containers/(id)/stats` will return networking information respectively for each interface.
- The `HostConfig` option now includes the `DnsOptions` field to configure

the container's DNS options.

- `POST /build` now optionally takes a serialized map of build-time variables.
- `GET /events` now includes a `timenano` field, in addition to the existing `time` field.
- `GET /events` now supports filtering by image and container labels.
- `GET /info` now lists engine version information and return the information of `CPUShares` and `Cpuset`.
- `GET /containers/json` will return `ImageID` of the image used by container.
- `POST /exec/(name)/start` will now return an HTTP 409 when the container is either stopped or paused.
- `POST /containers/create` now takes `KernelMemory` in HostConfig to specify kernel memory limit.
- `GET /containers/(name)/json` now accepts a `size` parameter. Setting this parameter to '1' returns container size information in the `SizeRw` and `SizeRootFs` fields.
- `GET /containers/(name)/json` now returns a `NetworkSettings.Networks` field, detailing network settings per network. This field deprecates the `NetworkSettings.Gateway`, `NetworkSettings.IPAddress`, `NetworkSettings.IPPrefixLen`, and `NetworkSettings.MacAddress` fields, which are still returned for backward-compatibility, but will be removed in a future version.
- `GET /exec/(id)/json` now returns a `NetworkSettings.Networks` field, detailing networksettings per network. This field deprecates the `NetworkSettings.Gateway`, `NetworkSettings.IPAddress`, `NetworkSettings.IPPrefixLen`, and `NetworkSettings.MacAddress` fields, which are still returned for backward-compatibility, but will be removed in a future version.
- The `HostConfig` option now includes the `OomScoreAdj` field for adjusting the badness heuristic. This heuristic selects which processes the OOM killer kills under out-of-memory conditions.

v1.20 API changes

Docker Remote API v1.20 documentation

- `GET /containers/(id)/archive` get an archive of filesystem content from a container.
- `PUT /containers/(id)/archive` upload an archive of content to be extracted to an existing directory inside a container's filesystem.
- `POST /containers/(id)/copy` is deprecated in favor of the above `archive` endpoint which can be used to download files and directories from a container.
- The `hostConfig` option now accepts the field `GroupAdd`, which specifies a list of additional groups that the container process will run as.

v1.19 API changes

Docker Remote API v1.19 documentation

- When the daemon detects a version mismatch with the client, usually when the client is newer than the daemon, an HTTP 400 is now returned instead of a 404.
- `GET /containers/(id)/stats` now accepts `stream` bool to get only one set of stats and disconnect.
- `GET /containers/(id)/logs` now accepts a `since` timestamp parameter.
- `GET /info` The fields `Debug`, `IPv4Forwarding`, `MemoryLimit`, and `SwapLimit` are now returned as boolean instead of as an int. In addition, the end point now returns the new boolean fields `CpuCfsPeriod`, `CpuCfsQuota`, and `OomKillDisable`.
- The `hostConfig` option now accepts the fields `CpuPeriod` and `CpuQuota`
- `POST /build` accepts `cpuperiod` and `cpuquota` options

v1.18 API changes

[Docker Remote API v1.18](#) documentation

- `GET /version` now returns `Os`, `Arch` and `KernelVersion`.
- `POST /containers/create` and `POST /containers/(id)/start`allow you to set ulimit settings for use in the container.
- `GET /info` now returns `SystemTime`, `HttpProxy`,`HttpsProxy` and `NoProxy`.
- `GET /images/json` added a `RepoDigests` field to include image digest information.
- `POST /build` can now set resource constraints for all containers created for the build.
- `CgroupParent` can be passed in the host config to setup container cgroups under a specific cgroup.
- `POST /build` closing the HTTP request cancels the build
- `POST /containers/(id)/exec` includes `Warnings` field to response.

Feedback? Suggestions? Can't find something in the docs?
Edit this page ● Request docs changes ● Get support
Rate this page: