

NAME

docker-inspect - Return low-level information on Docker objects

SYNOPSIS

docker inspect [OPTIONS] NAME|ID [NAME|ID...]

DESCRIPTION

This displays the low-level information on Docker object(s) (e.g. container, image, volume, network, node, service, or task) identified by name or ID. By default, this will render all results in a JSON array. If the container and image have the same name, this will return container JSON for unspecified type. If a format is specified, the given template will be executed for each result.

EXAMPLES

Get information about an image when image name conflicts with the container name, e.g. both image and container are named rhel7:

```
$ docker inspect --type=image rhel7
[
  {
    "Id": "fe01a428b9d9de35d29531e9994157978e8c48fa693e1bf1d221dffbbb67b170",
    "Parent": "10acc31def5d6f249b548e01e8ffbaccfd61af0240c17315a7ad393d022c5ca2",
    ....
  }
]
```

Getting information on a container

To get information on a container use its ID or instance name:

```
$ docker inspect d2cc496561d6
[
  {
    "Id": "d2cc496561d6d520cbc0236b4ba88c362c446a7619992123f11c809cded25b47",
    "Created": "2015-06-08T16:18:02.505155285Z",
    "Path": "bash",
    "Args": [],
    "State": {
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2015-06-08T16:18:03.643865954Z",
      "FinishedAt": "2015-06-08T16:57:06.448552862Z"
    },
    "Image": "ded7cd95e059788f2586a51c275a4f151653779d6a7f4dad77c2bd34601d94e4",
    "NetworkSettings": {
```

```

"Bridge": "",
"SandboxID": "6b4851d1903e16dd6a567bd526553a86664361f31036eaaa2f8454d6f4611f6f",
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"Ports": {},
"SandboxKey": "/var/run/docker/netns/6b4851d1903e",
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "7587b82f0dada3656fda26588aee72630c6fab1536d36e394b2bfbcf898c971d",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:12:00:02",
"Networks": {
  "bridge": {
    "NetworkID": "7ea29fc1412292a2d7bba362f9253545fecdfa8ce9a6e37dd10ba8bee7129812",
    "EndpointID": "7587b82f0dada3656fda26588aee72630c6fab1536d36e394b2bfbcf898c971d",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "02:42:ac:12:00:02"
  }
}
},
"ResolvConfPath": "/var/lib/docker/containers/d2cc496561d6d520cbc0236b4ba88c362c446a7619992123f11c809cded25b47",
"HostnamePath": "/var/lib/docker/containers/d2cc496561d6d520cbc0236b4ba88c362c446a7619992123f11c809cded25b47",
"HostsPath": "/var/lib/docker/containers/d2cc496561d6d520cbc0236b4ba88c362c446a7619992123f11c809cded25b47",
"LogPath": "/var/lib/docker/containers/d2cc496561d6d520cbc0236b4ba88c362c446a7619992123f11c809cded25b47",
"Name": "/adoring_wozniak",
"RestartCount": 0,
"Driver": "devicemapper",
"MountLabel": "",
"ProcessLabel": "",
"Mounts": [
  {
    "Source": "/data",
    "Destination": "/data",
    "Mode": "ro,Z",
    "RW": false
  }
],
"Propagation": "",
"AppArmorProfile": "",
"ExecIDs": null,
"HostConfig": {
  "Binds": null,

```

```

"ContainerIDFile": "",
"Memory": 0,
"MemorySwap": 0,
"CpuShares": 0,
"CpuPeriod": 0,
"CpusetCpus": "",
"CpusetMems": "",
"CpuQuota": 0,
"BlkioWeight": 0,
"OomKillDisable": false,
"Privileged": false,
"PortBindings": {},
"Links": null,
"PublishAllPorts": false,
"Dns": null,
"DnsSearch": null,
"DnsOptions": null,
"ExtraHosts": null,
"VolumesFrom": null,
"Devices": [],
"NetworkMode": "bridge",
"IpcMode": "",
"PidMode": "",
"UTSMode": "",
"CapAdd": null,
"CapDrop": null,
"RestartPolicy": {
  "Name": "no",
  "MaximumRetryCount": 0
},
"SecurityOpt": null,
"ReadonlyRootfs": false,
"Ulimits": null,
"LogConfig": {
  "Type": "json-file",
  "Config": {}
},
"CgroupParent": ""
},
"GraphDriver": {
  "Name": "devicemapper",
  "Data": {
    "DeviceId": "5",
    "DeviceName": "docker-253:1-2763198-d2cc496561d6d520cbc0236b4ba88c362c446a7619992123f11c809cde",
    "DeviceSize": "171798691840"
  }
},
"Config": {
  "Hostname": "d2cc496561d6",
  "Domainname": "",
  "User": "",
  "AttachStdin": true,
  "AttachStdout": true,
  "AttachStderr": true,

```

```

    "ExposedPorts": null,
    "Tty": true,
    "OpenStdin": true,
    "StdinOnce": true,
    "Env": null,
    "Cmd": [
        "bash"
    ],
    "Image": "fedora",
    "Volumes": null,
    "VolumeDriver": "",
    "WorkingDir": "",
    "Entrypoint": null,
    "NetworkDisabled": false,
    "MacAddress": "",
    "OnBuild": null,
    "Labels": {},
    "Memory": 0,
    "MemorySwap": 0,
    "CpuShares": 0,
    "Cpuset": "",
    "StopSignal": "SIGTERM"
}
}
]

```

Getting the IP address of a container instance

To get the IP address of a container use:

```
$ docker inspect --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' d2cc496561d6
172.17.0.2
```

Listing all port bindings

One can loop over arrays and maps in the results to produce simple text output:

```
$ docker inspect --format='{{range $p, $conf := .NetworkSettings.Ports}} \
  {{$p}} -> {{(index $conf 0).HostPort}} {{end}}' d2cc496561d6
80/tcp -> 80
```

You can get more information about how to write a Go template from: <https://golang.org/pkg/text/template/>.

Getting size information on a container

```
$ docker inspect -s d2cc496561d6
[
  {
    "SizeRw": 0,
    "SizeRootFs": 972,
  }
]
```

Getting information on an image

Use an image's ID or name (e.g., repository/name[:tag]) to get information about the image:

```
$ docker inspect ded7cd95e059
[
  {
    "Id": "ded7cd95e059788f2586a51c275a4f151653779d6a7f4dad77c2bd34601d94e4",
    "Parent": "48ecf305d2cf7046c1f5f8fcbcd4994403173441d4a7f125b1bb0ceead9de731",
    "Comment": "",
    "Created": "2015-05-27T16:58:22.937503085Z",
    "Container": "76cf7f67d83a7a047454b33007d03e32a8f474ad332c3a03c94537edd22b312b",
    "ContainerConfig": {
      "Hostname": "76cf7f67d83a",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "ExposedPorts": null,
      "Tty": false,
      "OpenStdin": false,
      "StdinOnce": false,
      "Env": null,
      "Cmd": [
        "/bin/sh",
        "-c",
        "#(nop) ADD file:4be46382bcf2b095fcb9fe8334206b584eff60bb3fad8178cbd97697fcb2ea83 in /"
      ],
    },
    "Image": "48ecf305d2cf7046c1f5f8fcbcd4994403173441d4a7f125b1bb0ceead9de731",
    "Volumes": null,
    "VolumeDriver": "",
    "WorkingDir": "",
    "Entrypoint": null,
    "NetworkDisabled": false,
    "MacAddress": "",
    "OnBuild": null,
    "Labels": {}
  },
  {
    "DockerVersion": "1.6.0",
    "Author": "Lokesh Mandvekar \u003clsm5@fedoraproject.org\u003e",
    "Config": {
      "Hostname": "76cf7f67d83a",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "ExposedPorts": null,
      "Tty": false,
      "OpenStdin": false,
      "StdinOnce": false,
      "Env": null,
      "Cmd": null,
      "Image": "48ecf305d2cf7046c1f5f8fcbcd4994403173441d4a7f125b1bb0ceead9de731",
      "Volumes": null,
    }
  }
]
```

```
"VolumeDriver": "",
"WorkingDir": "",
"Entrypoint": null,
"NetworkDisabled": false,
"MacAddress": "",
"OnBuild": null,
"Labels": {}
},
"Architecture": "amd64",
"Os": "linux",
"Size": 186507296,
"VirtualSize": 186507296,
"GraphDriver": {
  "Name": "devicemapper",
  "Data": {
    "DeviceId": "3",
    "DeviceName": "docker-253:1-2763198-ded7cd95e059788f2586a51c275a4f151653779d6a7f4dad77c2bd34601",
    "DeviceSize": "171798691840"
  }
}
}
]
```

OPTIONS

-f, --format="" Format the output using the given Go template

-h, --help[=false] help for inspect

-s, --size[=false] Display total file sizes if the type is container

--type="" Return JSON for specified type

SEE ALSO

docker(1)