

NAME

docker-system-info - Display system-wide information

SYNOPSIS

docker system info [OPTIONS]

DESCRIPTION

This command displays system wide information regarding the Docker installation. Information displayed includes the kernel version, number of containers and images. The number of images shown is the number of unique images. The same image tagged under different names is counted only once.

If a format is specified, the given template will be executed instead of the default format. Go's **text/template** package describes all the details of the format.

Depending on the storage driver in use, additional information can be shown, such as pool name, data file, metadata file, data space used, total data space, metadata space used, and total metadata space.

The data file is where the images are stored and the metadata file is where the meta data regarding those images are stored. When run for the first time Docker allocates a certain amount of data space and meta data space from the space available on the volume where **/var/lib/docker** is mounted.

EXAMPLES**Display Docker system information**

The example below shows the output for a daemon running on Ubuntu Linux, using the **overlay2** storage driver. As can be seen in the output, additional information about the **overlay2** storage driver is shown:

```
$ docker info
```

```
Client: Docker Engine - Community
Version:      24.0.0
Context:      default
Debug Mode: false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.10.4
    Path:      /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v2.17.2
    Path:      /usr/libexec/docker/cli-plugins/docker-compose

Server:
Containers: 14
  Running: 3
  Paused: 1
  Stopped: 10
Images: 52
Server Version: 23.0.3
Storage Driver: overlay2
  Backing Filesystem: extfs
```

```

  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
Logging Driver: json-file
Cgroup Driver: systemd
Cgroup Version: 2
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
Swarm: inactive
Runtimes: io.containerd.runc.v2 runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 2806fc1057397dbaeefbea0e4e17bddfbd388f38
runc version: v1.1.5-0-gf19387a
init version: de40ad0
Security Options:
  apparmor
  seccomp
   Profile: builtin
  cgroupns
Kernel Version: 5.15.0-25-generic
Operating System: Ubuntu 22.04 LTS
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 991.7 MiB
Name: ip-172-30-0-91.ec2.internal
ID: 4cee4408-10d2-4e17-891c-a41736ac4536
Docker Root Dir: /var/lib/docker
Debug Mode: false
Username: gordontheturtle
Experimental: false
Insecure Registries:
  myinsecurehost:5000
  127.0.0.0/8
Live Restore Enabled: false

```

You can also specify the output format:

```

$ docker info --format '{{json .}}'
{"ID":"4cee4408-10d2-4e17-891c-a41736ac4536","Containers":14, ...}

```

OPTIONS

```

-f, --format=""          Format output using a custom template: 'json':          Print in JSON format
'TEMPLATE':              Print output using the given Go template. Refer to https://docs.docker.com/go/formatting/ for more information about formatting output with templates

```

SEE ALSO

`docker-system(1)`