

# How to sort or order results docker ps --format? [closed]

Asked 8 years, 1 month ago Modified 3 months ago Viewed 27k times



35



**Closed.** This question is [not about programming or software development](#). It is not currently accepting answers.

💡 This question does not appear to be about [a specific programming problem, a software algorithm, or software tools primarily used by programmers](#). If you believe the question would be on-topic on [another Stack Exchange site](#), you can leave a comment to explain where the question may be able to be answered.

Closed 4 months ago.

The community reviewed whether to reopen this question 4 months ago and left it closed:

Original close reason(s) were not resolved

[Improve this question](#)

I haven't found any way to order my results when using `docker ps`

In my case I want to order by .Ports

```
docker ps -a --format "table {{.ID}}\t{{.Names}}\t{{.Ports}}"
```

How do I order the result?

`docker` `sorting`

Share Improve this question Follow

edited Jul 8 at 23:41



CPlus

5,103 ● 48 ● 33 ● 76

asked Sep 12, 2017 at 9:54



Theo

2,970 ● 5 ● 28 ● 59

- if you pipe and sort it? or have a look at Andy's answer from Docker Inspect Template Magic from Adrian Mouat [container-solutions.com/docker-inspect-template-magic](https://container-solutions.com/docker-inspect-template-magic) I guess you look for a docker only solution – [user2915097](#) Sep 12, 2017 at 10:41

I don't think you can do anything goolang template specific to do this. You need to build your own stuff for this. What may work is hitting the API and using `jq` to sort results and display. But not sure how easy that would be – [Tarun Lalwani](#) Sep 12, 2017 at 11:38

from [webgeodatavore.com/...](#) I see `Sort only value, not the object (return array)` and `jq '[.features[]].properties.pop_est] | sort'` countries.geojson so I guess this is possible with jq (I do not have a docke ravailable at the moment, I will test later) – [user2915097](#) Sep 12, 2017 at 11:48

and `jq -R .|jq -sr 'sort_by(length)[[]]'` sort by length from [reddit.com/r/commandline/comments/3z9jff/jq\\_examples](#) – [user2915097](#) Sep 12, 2017 at 11:49

see also [stackoverflow.com/questions/43930011/...](#) for a similar basic `docker command | sort` and ther is an issue on that subject, see [github.com/moby/moby/issues/31062](#) – [user2915097](#) Sep 12, 2017 at 13:24

## 6 Answers

Sorted by: Highest score (default)



If it's enough to simply sort by output column, you can use the following:

41

```
docker ps -a --format "table {{.ID}}\t{{.Names}}\t{{.Ports}}" | (read -r; printf "%s\n" "$REPLY"; sort -k 3 )
```



I also added a code for skipping the table headers and sorting only ps output data.



Share Improve this answer Follow

edited Apr 23, 2018 at 5:47

answered Apr 23, 2018 at 3:43



Rob

27.4k ● 16 ● 89 ● 103



Igor Maximov

606 ● 1 ● 5 ● 11

Sign up to request clarification or add additional context in comments.

## 1 Comment

Add a comment



[user2977468](#) Over a year ago

For those wondering, the final statement of `sort -k 3` indicates that the Docker container results will be sorted ascending by the third column of the format template. So in this example, the container list will be sorted by the respective container's exposed ports.

7

Reply



I built a docker ps pretty print function that can be put into your `.bash_profile` or `.bashrc` file that works somewhat like an alias for `docker ps` (with color output). [@art-rock-guitar-superhero](#) suggestions shows how to sort, but I've included this answer since typing the `--format` options and piping into a `sort` every time is a bit tedious.

11



```
function docker () {
    if [[ "$@" == "ps -p" ]]; then
        command docker ps --all --format "
        {{.ID}}\t{{.Names}}\t{{.Image}}\t{{.Ports}}\t{{.Status}}" \
```



```

| (echo -e "CONTAINER_ID\tNAMES\tIMAGE\tPORTS\tSTATUS" && cat) \
| awk '{printf
"\033[1;32m%s\t\033[01;38;5;95;38;5;196m%s\t\033[00m\033[1;34m%s\t\033[01;90m%s %s %s
%s %s %s %s\033[00m\n", $1, $2, $3, $4, $5, $6, $7, $8, $9, $10;}' \
| column -s'\t' -t \
| awk 'NR<2{print $0;next}{print $0 | "sort --key=2"}'
else
command docker "$@"
fi
}

```

usage: `$ docker ps -p`.

**EDIT:** I added suggestions from the comments from [@BrianVosburgh](#). Also, I kept forgetting to type `-p` so I switched the flag for this function to be `-a`, which is my regular usage of `docker ps`.

Share Improve this answer Follow

edited Nov 28, 2018 at 4:02

answered Oct 20, 2018 at 18:21



David John Coleman II

639 ● 8 ● 16

## 7 Comments ▼

Add a comment



[Brian Vosburgh](#) Over a year ago

Nice function. :) But I think there is an extraneous backslash in the 'awk' pattern: `...196m%s\t...` should read `...196m%s\t...`.

1 Reply



[Brian Vosburgh](#) Over a year ago

Also, you could sort the table with something like this: `docker ps -p | awk 'NR<2{print $0;next}{print $0 | "sort --key=2"}'`

1 Reply



[David John Coleman II](#) Over a year ago

@BrianVosburgh, yes, thanks, I removed the extraneous backslash. I'm not sure why, but it had no impact on the output. Also, the extra stuff to sort by second column (key) of name, is a nice suggestion as well, and so I've updated the original answer with your sort command.

0 Reply



[Ser](#) Over a year ago

You cannot sort on the status, as this field contains `Up 2 days`, `Shutdown about a day ago` or whatever. It just works if all the containers were created on the same moment and they have all the same status. I create another answer that describes how to generate the ISO date from the `Status` value and sort on it.

0 Reply



[David John Coleman II](#) Over a year ago

@Ser the proposed solution sorts by `Name` only, it's not configurable to sort by `Status` intentionally to make it a simple use for general purpose.



|

## ▲ Format and Order *docker ps*

### 8 List containers



```
docker ps
```



### Synopsis



```
docker ps [--format="TEMPLATE"]

--format="TEMPLATE"
Pretty-print containers using a Go template.
Valid placeholders:
  .ID - Container ID
  .Image - Image ID
  .Command - Quoted command
  .CreatedAt - Time when the container was created.
  .RunningFor - Elapsed time since the container was started.
  .Ports - Exposed ports.
  .Status - Container status.
  .Size - Container disk size.
  .Names - Container names.
  .Labels - All labels assigned to the container.
  .Label - Value of a specific label for this container. For example {{.Label
"com.docker.swarm.cpu"}}
  .Mounts - Names of the volumes mounted in this container.
```

### Display containers with their commands

```
docker ps --format "{{.ID}}: {{.Command}}"
```

### Display containers with their labels in a table

```
docker ps --format "table {{.ID}}\t{{.Labels}}"
```

### Display containers with their node label in a table

```
docker ps --format 'table {{.ID}}\t{{(.Label "com.docker.swarm.node")}}'
```

Share Improve this answer Follow

edited Jul 25, 2020 at 13:42

answered Sep 12, 2017 at 13:45



Bhargav Rao

52.6k ● 29 ● 130 ● 142



Jinna Baalu

7,957 ● 40 ● 51

## 6 Comments ▾

Add a comment



ssc Over a year ago

what is "\$REPLY" ?

▲ 3 Reply ...



bradym Over a year ago

Read is a bash builtin that reads from stdin and stores the value in a variable. If you don't provide a name for the variable it uses REPLY. [ss64.com/bash/read.html](https://ss64.com/bash/read.html)

▲ 0 Reply ...



Benjamin W. Over a year ago

This answer didn't contain the sorting part at all, and just copied it from the [this answer](#), currently second highest voted, without attribution?

▲ 13 Reply ...



Jinna Baalu Over a year ago

@BenjaminW. So great that you found the answer I copied from, which I was unaware. I copied in Nov 2018, answered in Sep 2017.

▲ 0 Reply ...



user2977468 Over a year ago ✎

This does NOT sort. Consider concise and capable [answer here](#).

▲ 14 Reply ...

|



1

I usually want to sort the output by container name, which appears at the very end of each line, so I use a sed substitution to add a temporary copy of the desired column (the last column) to the front of the line, sort the new lines, then remove that temporary column:



```
docker ps |
sed -E 's/^(^.* )([^\ ]*$)/\2 \1\2/' |
LC_ALL=C sort |
cut -d' ' -f2-
```



The regex `(^.* )([^\ ]*$)` matches the last group of characters on each line that doesn't contain a space (`[^\ ]*$`), then captures that group (that regex is enclosed in parentheses), as well as all of the characters preceding it (`(^.* )`). The expression `\2 \1\2` is then the second group (the container name field), a space, and the original line including the original container name field.

Setting `LC_ALL=C` in the environment of `sort` ensures that fields are sorted lexicographically, so that "NAME" still appears as the column heading in the final output, rather than these headings appearing as some row other than the first row.

`cut -d' ' -f2-` then strips out the temporary leading "NAME" column from the final output: it prints fields/columns from number 2 onwards, where fields are delimited by space characters.

This is a pretty convenient way of sorting by container name, since I don't need to do something more cumbersome, like strip out extraneous spaces between columns, use `sort` to sort by a particular column, and then add back the spaces to format the columns nicely again.

I have this pipeline as a shell alias called `docker-pss`:

```
alias docker-pss="docker ps | sed -E 's/(\^.* )(\^ ]*$)/\2 \1\2/' | LC_ALL=C
sort | cut -d' ' -f2-"
```

Since you want to sort by ports, and your `docker ps` format string has this as the last column, you should be able to use this pipeline to achieve that. However, if you want to sort by the actual port numbers that appear in this column (e.g. `80` in `::1->80/tcp` or `80/tcp`), you'll have to match on the value that appears between the possible arrow (`->`) and guaranteed slash (`/`) in this field.

Here's how you could do that using a sed expression of a similar nature to the one above, using the default output of `docker ps`, not using a custom format string:

```
{
  echo -n 'PORT ' # Add extra heading for temporary column. This won't appear
  in the final output
  docker ps
} |
sed -E 's/(\^.*(->| ))([0-9]+)(\./.*$)/\3 \1\3\4/' |
sort -nt' ' -k1,1 |
cut -d' ' -f2-
```

This won't correctly sort lines with an empty "PORTS" field. Instead, it will strip their container ID from the output and sort them as if their container ID was their port number. If you want to exclude such lines from the output, you can filter them out with `grep` before running `sed`:

```
{
  echo -n 'PORT '
  docker ps
} |
grep -E '^PORT |(->| )([0-9]+/)' | # Filter out lines that lack a port number,
except the heading line
sed -E 's/(\^.*(->| ))([0-9]+)(\./.*$)/\3 \1\3\4/' |
sort -nt' ' -k1,1 |
cut -d' ' -f2-
```

Share Improve this answer Follow

answered Jul 8 at 15:30



Jivan Pal

190 ● 10

## Comments

Add a comment



0



## Prettify Docker ps output

- open terminal
- run export command to export **Env** Variable
- you can customise according to desired output needed



```
export
FORMAT="ID\t{ {.ID}}\nNAME\t{ {.Names}}\nIMAGE\t{ {.Image}}\nPORTS\t{ {.Ports}}\nCOMMA
```

- run command **docker ps --format="\$FORMAT"**

### e.g Output

```
khizerrehan@khizerrehans-MacBook-Pro ~ % docker ps --format="$FORMAT"
ID      9bfbfe65dce1
NAME    happy_joliot
IMAGE   nginx:alpine
PORTS   0.0.0.0:80->80/tcp, :::80->80/tcp
COMMAND "/docker-entrypoint. ...."
CREATED 2021-09-05 00:20:45 +0500 PKT
STATUS  Up 4 minutes
```

Share Improve this answer Follow

answered Sep 4, 2021 at 19:28



khizerrehandev

1,577 ● 17 ● 15

## Comments

Add a comment



0

I know this is not strictly related to the OP question, but if you want to sort on the Status date, you need to support date that are relatives : `Up 2 days ago` , `Shutdown about a day ago` etc.

This bash script generates an ISO date from the Status value, then we sort on the ISO date and finally we remove the ISO date.



```
# 1- List all containers
#   As I'm working with swarm, you may want to adapt to your docker cli usage.
#   Use "docker ps --format '{{ json . }}'" to see available fields
# 2- add a column that translate "Shutdown 2 days ago" into ISO date. Also
#   supports "Shutdown about a day ago".
# 3- sort on that date
# 4- remove the ISO date and add colors
# 5- make the output readable
docker stack ps --format '{{ .CurrentState }}\t{{ .Name }}\t{{ .Node }}\t{{ .Image }}' demo \
  | awk -F'\t' '{system("date --rfc-3339=seconds -u -d \"$(printf \"%10s\" $1)\" |
cut -d \" \" -f2- | sed \"s/about//g\") \" | tr -d \"\n\")\" ); printf
\"%t%s\t%s\t%s\t%s\t%s\n\", $1, $2, $3, $4, $5;}' \
  | sort -r \
  | awk -F'\t' '{ printf "\033[32m%s\033[39m\t%s\t%s\t%s\t%s\n", $2, $3, $4, $5,
$6;}' \
  | column -o " " -t -s '$\t'
```

It generates something like that :

```
Running 21 seconds ago rs-fem.1 vm-p
Running 2 days ago simple-worker.1 vm-perf-
Running 2 days ago rs-storage.1 vm-perf-
Running 2 days ago rs-registry.1 vm-perf-
Running 2 days ago rs-rabbitmq.1 vm-perf.
Running 2 days ago rs-postgres.1 vm-perf.
Running 2 days ago rs-phpgadmin.1 vm-perf.
Running 2 days ago rs-notifier.1 vm-perf.
```

Share Improve this answer Follow

answered Dec 1, 2021 at 12:55



Ser

2,920 ● 27 ● 33

## Comments

Add a comment

### Start asking to get answers

Find the answer to your question by asking.

Ask question

### Explore related questions

docker sorting

See similar questions with these tags.