In this scenario you will learn how to use the _--format__ parameters to pretty-print output from docker ps and docker inspect.

Example 1 - Names and Images as Table
The format of docker ps can be formatted to only display the information relevant to you.

Start Example Container
Start by launching a example container - docker run -d redis

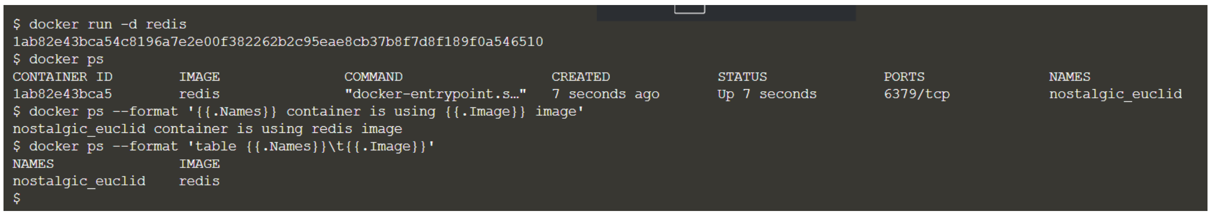
Format
The standard docker ps command outputs the name, image used, command, uptime and port information.

To limit which columns are displayed, use the _--format__ parameter. The parameter allows pretty-printing containers using a Go template syntax.

docker ps --format '{{.Names}} container is using {{.Image}} image'

As it's using Go templates, it includes helper functions such as table.

docker ps --format 'table {{.Names}}\t{{.Image}}'



```
$ docker run -d redis
1ab82e43bca54c8196a7e2e00f382262b2c95eae8cb37b8f7d8f189f0a546510
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
1ab82e43bca5        redis               "docker-entrypoint.s…"   7 seconds ago       Up 7 seconds        6379/tcp            nostalgic_euclid
$ docker ps --format '{{.Names}} container is using {{.Image}} image'
nostalgic_euclid container is using redis image
$ docker ps --format 'table {{.Names}}\t{{.Image}}'
NAMES               IMAGE
nostalgic_euclid    redis
$
```

Xargs : xargs is a Unix/Linux command that reads items from standard input and generates a command line for each item. The generated command line includes the item as an argument, and the command line is executed by xargs. This allows you to execute a command on a set of items that are passed through the standard input.

$ echo "file1 file2 file3" | xargs ls -l

In this example, the echo command outputs a list of files separated by whitespace to the standard output, which is then piped to xargs. The xargs command reads the list of files from the standard input and generates a command line for each file, which includes the file name as an argument. The command line generated by xargs is ls -l file1, ls -l file2, and ls -l file3. These command lines are executed by xargs, which in turn executes the ls -l command for each file.

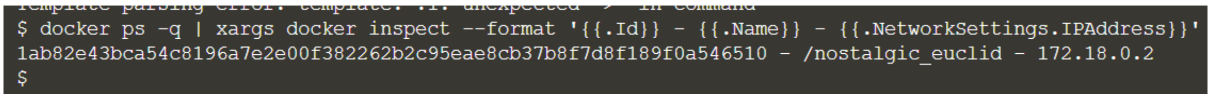
xargs is a powerful command and can be used to perform a wide range of tasks, such as processing large sets of files, executing commands in parallel, and generating complex command lines. However, it is important to use xargs with caution, as it can potentially generate long and complex command lines that may cause errors or security issues.

Example 2 - List IP addresses
However, the format parameter allow supports displaying data that is already exposed via the docker ps command. If you wanted to include additional information, such as the IP Address of the container, then the data needs to come via docker inspect.

Thankfully, the docker inspect also supports pretty-printing the results via a Go Template. The container IDs from docker ps can be piped into docker inspect.

The format parameter can then access all of the container information. Below is an example of listing all the IP addresses for the running containers.

docker ps -q | xargs docker inspect --format '{{ .Id }} - {{ .Name }} - {{ .NetworkSettings.IPAddress }}'



```
$ docker ps -q | xargs docker inspect --format '{{.Id}} - {{.Name}} - {{.NetworkSettings.IPAddress}}'
1ab82e43bca54c8196a7e2e00f382262b2c95eae8cb37b8f7d8f189f0a546510 - /nostalgic_euclid - 172.18.0.2
$
```