Name: Animesh Jain

Roll Number: R171218022

SAP ID: 500069453

Metadata and Labels

Add a single label

```
$ docker run -l user=12345 -d redis
```

```
$ docker run -l user=12345 -d redis
142d72c2e1ae56fe15c2cd72fd99088068e9758f19adcc1afbb75a0c39478baa
```

To create two labels in the file, one for the user and the second assigning a role.

```
$ echo 'user=123461' >> labels && echo 'role=cache' >> labels
```

The --label-file=<filename> option will create a label for each line in the file.

```
$ docker run --label-file=labels -d redis
```

```
$ echo 'user=123461' >> labels && echo 'role=cache' >> labels
$ docker run --label-file=labels -d redis
b7b3633abbb63d43034184d4c5868210eb492127d89a815da5255c2b3ff52e65
```

By providing the running container's friendly name or hash id, you can query all of it's metadata.

```
$ docker inspect rd
```

```
$ docker inspect rd
[
    {
        "Id": "4f07372c37d841880298c7fe11467c08ec86b790b72e43295f1f70f7c37634f5",
        "Created": "2021-04-30T11:28:07.401196298Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "redis-server"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 1075,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2021-04-30T11:28:07.874929167Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:4760dc956b2ddc9ac1c508936e39b63a22c6f0640ef58c1b10ff73f04
        "ResolvConfPath": "/var/lib/docker/containers/4f07372c37d841880298c7fe1146
7634f5/resolv.conf",
```

```
"Networks": {
    "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "b76c76e2e91035ca31be7c618f6e69aa8dcd379dcb87268c3eeb4a713c4ad47d",
        "EndpointID": "228a54315468065dd4c2772e12b7aac240d63edf179e95fe6b405fb4b0f17a66",
        "Gateway": "172.18.0.1",
        "IPAddress": "172.18.0.2",
        "IPPrefixLen": 24,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:12:00:02",
        "DriverOpts": null
    }
}
```

Using the -f option you can filter the JSON response to just the Labels section we're interested in.

```
$ docker inspect -f "{{json .Config.Labels }}" rd
```

```
$ docker inspect -f "{{json .Config.Labels }}" rd
{"com.katacoda.created":"automatically","com.katacoda.private-msg":"magic","user":"scrapbook"}
```

Inspecting images works in the same way however the JSON format is slightly different, naming it ContainerConfig instead of Config.

```
$ docker inspect -f "{{json .ContainerConfig.Labels }}" katacoda-label-
example
```

```
$ docker inspect -f "{{json .ContainerConfig.Labels }}" katacoda-label-example
{"com.katacoda.build-date":"2015-07-01T10:47:29Z","com.katacoda.course":"Docker","com.katacoda.private-msg":"Hel
loWorld","com.katacoda.version":"0.0.5","vendor":"Katacoda"}
```

To show running label

```
$ docker ps --filter "label=user=scrapbook"
```

```
$ docker ps --filter "label=user=scrapbook"
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
            NAMES
4f07372c37d8        redis               "docker-entrypoint.s…"   23 minutes ago      Up 23 minutes       6379/tc
p           rd
```

Filtering images

```
$ docker images --filter "label=vendor=Katacoda"
```

```
$ docker images --filter "label=vendor=Katacoda"
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
katacoda-label-example  latest              0c15a52168b0        25 minutes ago      112MB
```