

```

1 Lab1. Container vs Container Image
2
3 [실습1. ubuntu Base Image + git]
4 1. Ubuntu Base Image에 Git 설치하기
5 $ docker image ls
6 REPOSITORY TAG IMAGE ID CREATED SIZE
7 hello-world latest feb5d9fea6a5 19 months ago 13.3kB
8
9
10 $ docker pull ubuntu:latest
11 latest: Pulling from library/ubuntu
12 2ab09b027e7f: Pull complete
13 Digest: sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babbc295a0428a6d21
14 Status: Downloaded newer image for ubuntu:latest
15 docker.io/library/ubuntu:latest
16
17
18 $ docker image ls
19 REPOSITORY TAG IMAGE ID CREATED SIZE
20 ubuntu latest 08d22c0ceb15 6 weeks ago 77.8MB
21 hello-world latest feb5d9fea6a5 19 months ago 13.3kB
22
23
24 $ docker run -it --name git ubuntu:latest bash
25 root@9fb2faf09844:/# ls
26 bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr
27 var
28
29 root@9fb2faf09844:/# git
30 bash:git : command not found
31
32 root@9fb2faf09844:/# apt update
33 Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
34 Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
35 Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
36 Get:4 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [23.2 kB]
37 Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
38 Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
39 Get:7 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1000 kB]
40 Get:8 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [911 kB]
41 Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [959 kB]
42 Get:10 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
43 Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
44 Get:12 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
45 Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1301 kB]
46 Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1054 kB]
47 Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [28.6 kB]
48 Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1154 kB]
49 Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [49.4 kB]
50 Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [25.6 kB]
51 Fetched 26.8 MB in 6s (4624 kB/s)
52 Reading package lists... Done
53 Building dependency tree... Done
54 Reading state information... Done
55 4 packages can be upgraded. Run 'apt list --upgradable' to see them.
56
57 root@9fb2faf09844:/# apt-get install -y git
58
59 root@9fb2faf09844:/# git --version
60 git version 2.34.1

```

```
61
62
63
64 2. 다른 세션에서, 즉 또 다른 커넥션으로 연결하여
65     $ docker images | grep ubuntu
66     ubuntu      latest    08d22c0ceb15   6 weeks ago    77.8MB
67
68     $ docker commit git ubuntu:git
69     sha256:ada81e9a54411b13b0602e6b64dfa7f1523fdf747e160ec800ed6f486c0f806d
70
71     $ docker images | grep ubuntu
72     ubuntu      git      ada81e9a5441   41 seconds ago  196MB
73     ubuntu      latest   08d22c0ceb15   6 weeks ago    77.8MB
74
75     $ docker run -it --name git2 ubuntu:git bash
76     root@b51cafbbb61b:/# git --version
77     git version 2.34.1
78
79
80
```

81 [실습2. Container Image Layer 다루기]

82 1. Docker Image 정보 확인

```
83     $ docker pull nginx
84     Using default tag: latest
85     latest: Pulling from library/nginx
86     26c5c85e47da: Pull complete
87     4f3256bdf66b: Pull complete
88     2019c71d5655: Pull complete
89     8c767bdb9ae: Pull complete
90     78e14bb05fd3: Pull complete
91     75576236abf5: Pull complete
92     Digest: sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
93     Status: Downloaded newer image for nginx:latest
94     docker.io/library/nginx:latest
95
96
97     $ docker inspect nginx
98     [
99         {
100             "Id": "sha256:6efc10a0510f143a90b69dc564a914574973223e88418d65c1f8809e08dc0a1f",
101             "RepoTags": [
102                 "nginx:latest"
103             ],
104             "RepoDigests": [
105                 "nginx@sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94"
106             ],
107             "Parent": "",
108             "Comment": "",
109             "Created": "2023-04-12T08:51:21.83677948Z",
110             "Container": "6d1de575d4fe315e404a58482deb30ca38735c5a52309bf8b7f46b81fffc80d4",
111             "ContainerConfig": {
112                 "Hostname": "6d1de575d4fe",
113                 "Domainname": "",
114                 "User": "",
115                 "AttachStdin": false,
116                 "AttachStdout": false,
117                 "AttachStderr": false,
118                 "ExposedPorts": {
119                     "80/tcp": {}
120                 },
121                 "Tty": false,
```

```

122     "OpenStdin": false,
123     "StdinOnce": false,
124     "Env": [
125         "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
126         "NGINX_VERSION=1.23.4",
127         "NJS_VERSION=0.7.11",
128         "PKG_RELEASE=1~bullseye"
129     ],
130     "Cmd": [
131         "/bin/sh",
132         "-c",
133         "#(nop) ",
134         "CMD [W'nginxW' W'-gW' W'daemon off;W']"
135     ],
136     "Image": "sha256:20d4c097fb72a6a8c4d8c3989a40adcdfbdba4da700bd82c594d24d4effcdba",
137     "Volumes": null,
138     "WorkingDir": "",
139     "Entrypoint": [
140         "/docker-entrypoint.sh"
141     ],
142     "OnBuild": null,
143     "Labels": {
144         "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
145     },
146     "StopSignal": "SIGQUIT"
147 },
148 "DockerVersion": "20.10.23",
149 "Author": "",
150 "Config": {
151     "Hostname": "",
152     "Domainname": "",
153     "User": "",
154     "AttachStdin": false,
155     "AttachStdout": false,
156     "AttachStderr": false,
157     "ExposedPorts": {
158         "80/tcp": {}
159     },
160     "Tty": false,
161     "OpenStdin": false,
162     "StdinOnce": false,
163     "Env": [
164         "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
165         "NGINX_VERSION=1.23.4",
166         "NJS_VERSION=0.7.11",
167         "PKG_RELEASE=1~bullseye"
168     ],
169     "Cmd": [
170         "nginx",
171         "-g",
172         "daemon off;"
173     ],
174     "Image": "sha256:20d4c097fb72a6a8c4d8c3989a40adcdfbdba4da700bd82c594d24d4effcdba",
175     "Volumes": null,
176     "WorkingDir": "",
177     "Entrypoint": [
178         "/docker-entrypoint.sh"
179     ],
180     "OnBuild": null,
181     "Labels": {
182         "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"

```

```

183     },
184     "StopSignal": "SIGQUIT"
185 },
186 "Architecture": "amd64",
187 "Os": "linux",
188 "Size": 142151046,
189 "VirtualSize": 142151046,
190 "GraphDriver": {
191     "Data": {
192         "LowerDir":
193             "/var/lib/docker/overlay2/9fbd542e2b00e1b058bfefe2e2db910fba9c4a7ae2629566bbbeb3129b7d0afe
194             /diff:/var/lib/docker/overlay2/74457d7ff87c761a5d673265c09a985c7f267e4c30e1c7ddb5e86f514d962
195             6fb/diff:/var/lib/docker/overlay2/dd4882652704815d75803cff66cdfef92bc962beac459db8c27da04a6c
196             7bc9e9/diff:/var/lib/docker/overlay2/6818156a861295438c7a5bb66b9aa97fb8328f610636f3894bb475
197             ed824749e8/diff:/var/lib/docker/overlay2/f6795d3e6de6123376df02197ec0e99ae7780f86773e14c18bf
198             5c3f166d37c0e/diff",
199         "MergedDir":
200             "/var/lib/docker/overlay2/dd90ce838ca3f9f9a093701fd8250f4af050c138f7337d74fa23b5b0bf7d22ff/m
201             erged",
202         "UpperDir":
203             "/var/lib/docker/overlay2/dd90ce838ca3f9f9a093701fd8250f4af050c138f7337d74fa23b5b0bf7d22ff/di
204             ff",
205         "WorkDir":
206             "/var/lib/docker/overlay2/dd90ce838ca3f9f9a093701fd8250f4af050c138f7337d74fa23b5b0bf7d22ff/w
207             ork"
208     },
209     "Name": "overlay2"
210 },
211 "RootFS": {
212     "Type": "layers",
213     "Layers": [
214         "sha256:ed7b0ef3bf5bbec74379c3ae3d5339e666a314223e863c70644f7522a7527461",
215         "sha256:fb6d57d46ad57cbb50dfe16eba53a31d8808aa43e3a4a3ddd5c04d2d6ee0ecc5",
216         "sha256:935b5bd454e1973eb283fbc76d5613def62c190c8c4cc621576ef32f7c7dcb56",
217         "sha256:f12d4345b7f35505fc42fd613ae2c6749ddba30d6c9025e1a5b070c97c1ad2bb",
218         "sha256:79974a1a12aa3a3876d8dbcf16b2eda15ba5372574d38dd94bac68d339b6e124",
219         "sha256:9d907f11dc742442217bda25fde38f11851c7d495cacc6fc4e3869f4cd2c9ad9"
220     ]
221 },
222 "Metadata": {
223     "LastTagTime": "0001-01-01T00:00:00Z"
224 }
225 }
226 ]

```

2. Docker Image 저장소 위치 확인

```

219 $ docker info
220 Client:
221   Context:    default
222   Debug Mode: false
223   Plugins:
224     buildx: Docker Buildx (Docker Inc.)
225     Version: v0.10.4
226     Path:     /usr/libexec/docker/cli-plugins/docker-buildx
227     compose: Docker Compose (Docker Inc.)
228     Version: v2.17.2
229     Path:     /usr/libexec/docker/cli-plugins/docker-compose
230
231 Server:

```

```

232 Containers: 3
233   Running: 1
234   Paused: 0
235   Stopped: 2
236 Images: 4
237 Server Version: 23.0.4
238 Storage Driver: overlay2
239   Backing Filesystem: extfs
240   Supports d_type: true
241   Using metacopy: false
242   Native Overlay Diff: true
243   userxattr: false
244 Logging Driver: json-file
245 Cgroup Driver: systemd
246 Cgroup Version: 2
247 Plugins:
248   Volume: local
249   Network: bridge host ipvlan macvlan null overlay
250   Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
251 Swarm: inactive
252 Runtimes: io.containerd.runc.v2 runc
253 Default Runtime: runc
254 Init Binary: docker-init
255 containerd version: 2806fc1057397dbaeefbea0e4e17bddfbd388f38
256 runc version: v1.1.5-0-gf19387a
257 init version: de40ad0
258 Security Options:
259   apparmor
260   seccomp
261     Profile: builtin
262   cgroupns
263 Kernel Version: 5.15.0-1031-aws
264 Operating System: Ubuntu 22.04.2 LTS
265 OSType: linux
266 Architecture: x86_64
267 CPUs: 1
268 Total Memory: 966.2MiB
269 Name: ip-10-0-10-23
270 ID: bdf0c5e6-005e-41fa-af85-2194ba9394ee
271 Docker Root Dir: /var/lib/docker <----- 여기서 확인
272 Debug Mode: false
273 Registry: https://index.docker.io/v1/
274 Experimental: false
275 Insecure Registries:
276   127.0.0.0/8
277 Live Restore Enabled: false
278
279
280 $ sudo -i
281 # cd /var/lib/docker/overlay2
282
283
284 3. Disk size check
285 # du -sh image
286 root@ip-10-0-10-23:/var/lib/docker/overlay2# cd ..
287 root@ip-10-0-10-23:/var/lib/docker# ls
288 buildkit containers engine-id image network overlay2 plugins runtimes swarm tmp volumes
289
290 root@ip-10-0-10-23:/var/lib/docker# du -sh image
291 1.2M   image
292

```

```
293
294 # du -sh overlay2
295 root@ip-10-0-10-23:/var/lib/docker# du -sh overlay2
296 681M    overlay2
297
298
299 4. Directory 정리
300 1)/var/lib/docker/image --> imagedb
301                                --> layerdb : imagedb에 대한 정보 소유
302 2)/var/lib/docker/overlay2 --> layerdb에 대한 정보 소유
303 3)/var/lib/docker/overlay2/... --> Image에 대한 변경 사항들 저장
304 4)/var/lib/docker/overlay2/l(영문소문자 l) --> 실제 변경사항들 저장된 directory
305 5)/var/lib/docker/overlay2/l --> 실제 Container에 대한 파일 시스템 저장
306 6)해당 Image를 Download할 때마다 overlay2의 사이즈가 커짐.
```