

5 Docker の概要

- docker run hello-world : □-プルして実行して Docker Hub から pull して

docker images

```
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	bf756fb1ae65	13 months ago	13.3kB

↑ sha256 のハッシュ形式

docker ps -a

```
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b81642908c97	hello-world	"/hello"	3 seconds ago	Exited (0) 2 seconds ago		hoge
51de6f12971b	hello-world	"/hello"	About a minute ago	Exited (0) About a minute ago		happy_albattani
a5a69ab6f901	hello-world	"/hello"	8 minutes ago	Exited (0) 8 minutes ago		agitated_herschel

今動かして docker 7 行も確認 - a 行くと本当に今動いてるやつだ

Build

- Dockerfile を使って docker build -t node-app:0.1 .

name: Tag でタグをつける

```
student_01_b137e2b8881c@cloudshell:~/test (qwiklabs-gcp-01-09d20ff5303d) $ cat Dockerfile
```

```
# 上位イメージとして正式な Node ランタイムを使用します
FROM node:6

# コンテナの作業ディレクトリを /app に設定します
WORKDIR /app

# 現行ディレクトリの内容を /app のコンテナにコピーします
ADD . /app

# コンテナのポート 80 で外部からアクセスできるようにします
EXPOSE 80

# コンテナの起動時に node を使用して app.js を実行します
CMD ["node", "app.js"]
```

node:6 は上位イメージ

⇒ 結果として

node, node-app 2つのイメージで実行

/app は Docker 内のアプリの置き場

実行

- docker run -p 4000:80 --name my-app node-app:0.1

自ホスト 4000 番ポートと my-app の 80 番に接続

- curl http://localhost:4000

レスポンスとして返ってくる

```

hello world
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker stop my-app
my-app
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker rm my-app
my-app
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker run -p 4000:80 --name my-app -d node-app:0.1
8e144d7b09b676a7a3c83348aa6f13312b91dd233c067a1ee4f529eff6cf9fe1
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
8e144d7b09b6   node-app:0.1   "node app.js"           3 seconds ago Up 2 seconds    0.0.0.0:4000->80/tcp    my-app
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $

```

← インスタンスはめてインスタンス削除

← -d オプションでバックグラウンド実行

○ docker logs [コンテナID] : ログの確認

↑
-f : 識別できぬ最初の数文字をひたす
: tail 表示

○ テーブル

○ docker exec -it [コンテナID] bash

- Dockerfile で指定した WORKDIR に bash を開く

○ docker inspect [コンテナID]

- コンテナのメタ情報. 2つの docker インスタンス = コンテナ, 1:1 とは

○ 公開 GCR に DockerHub の Google 版

① 7.2.2 のインストール確認

```

student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ gcloud config list project
[core]
project = qwiklabs-gcp-01-09d20ff5303d

Your active configuration is: [cloudshell-4573]
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker tag node-app:0.2 gcr.io/qwiklabs-gcp-01-09d20ff5303d/node-app:0.2
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
node-app             0.2          55e01022ef26     9 minutes ago   884MB
gcr.io/qwiklabs-gcp-01-09d20ff5303d/node-app             0.2          55e01022ef26     9 minutes ago   884MB
node-app             0.1          581181f571a0     30 minutes ago   884MB
hello-world         latest       bf756fb1ae65     13 months ago   13.3kB
node                 6           ab290b853066     21 months ago   884MB
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $ docker push gcr.io/qwiklabs-gcp-01-09d20ff5303d/node-app:0.2
The push refers to repository [gcr.io/qwiklabs-gcp-01-09d20ff5303d/node-app]
bab115fdf745: Pushed
aabe375ccd8c: Pushed
f39151891503: Pushed
f1965d3c206f: Pushed
a27518e43e49: Layer already exists
910d7fd9e23e: Layer already exists
4230ff7f2288: Layer already exists
2c719774c1e1: Layer already exists
ec62f19bb3aa: Layer already exists
f94641f1felf: Layer already exists
0.2: digest: sha256:e53e08bfla0c4eaf66686e18ba3f283852692fa97a5f6a8794fb20f42a1175fb size: 2422
student_01_b137e2b8881c@cloudshell:~ (qwiklabs-gcp-01-09d20ff5303d) $

```

② node-app:0.2 に GCR に 3.7.2 をする TAG プリット

③ GCR に 7.2.2

1.5 確認

④ docker stop, rm で コンテナを削除

⑤ 7.2.2 に image 自体も削除 docker rmi [イメージ名]

⑥ docker pull gcr.io/... push したイメージを pull して
実行で動作確認

4 Kubernetes Engine : Quick Start

- gcloud config set compute/zone us-central1-a

- クラスター作成

gcloud container clusters create my-cluster

- credentials 取得

gcloud container clusters get-credentials my-cluster

- 作成済みの hello-app をインストールして hello-server という deployment 作成

```
kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:1.0
```

- Kubernetes Service (サービスを外部に expose)

```
kubectl expose deployment hello-server --type=LoadBalancer --port 8080
```

↑
Compute Engine の
ロードバランサー作成

↑
公開ポート

確認: kubectl get service

hello-server サービスは external-ip に対してリスニング

- クラスター削除

gcloud container clusters delete my-cluster

8 Kubernetes is a Cloud Native Platform

○ 1727作成

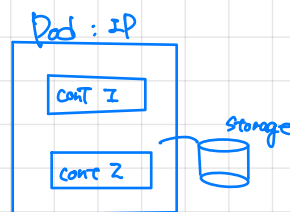
- `gcloud config set compute/zone us-central1-b`
- `gcloud container clusters create io`

↓

create deployment & expose deployment ref: chapter 21, L5.

- o Pod

- 100% 正の $\gamma = \tau_T \in \text{正数}$
- 正の $\gamma = \tau_T \in \mathbb{R}$ にして Bob の $\text{Ipr} = \tau'$ は変化する



- K8s is yaml z". `kubectl create -f [73112]`

kubectl get pods

kubectl describe pods [ポッド名]

で、 $\frac{1}{2}$ になる。

(external-IP)

- ・ クラウドのホストが2つあるホストのホストに接続

kernel port-forward [city "名"] 10080:80 → pod's port

- Service

- Podt = 1T3. 2T3. 3T3. 4T3. 5T3. 6T3. 7T3. 8T3. 9T3. 10T3. 11T3. 12T3. 13T3. 14T3. 15T3. 16T3. 17T3. 18T3. 19T3. 20T3. 21T3. 22T3. 23T3. 24T3. 25T3. 26T3. 27T3. 28T3. 29T3. 30T3. 31T3. 32T3. 33T3. 34T3. 35T3. 36T3. 37T3. 38T3. 39T3. 40T3. 41T3. 42T3. 43T3. 44T3. 45T3. 46T3. 47T3. 48T3. 49T3. 50T3. 51T3. 52T3. 53T3. 54T3. 55T3. 56T3. 57T3. 58T3. 59T3. 60T3. 61T3. 62T3. 63T3. 64T3. 65T3. 66T3. 67T3. 68T3. 69T3. 70T3. 71T3. 72T3. 73T3. 74T3. 75T3. 76T3. 77T3. 78T3. 79T3. 80T3. 81T3. 82T3. 83T3. 84T3. 85T3. 86T3. 87T3. 88T3. 89T3. 90T3. 91T3. 92T3. 93T3. 94T3. 95T3. 96T3. 97T3. 98T3. 99T3. 100T3. 101T3. 102T3. 103T3. 104T3. 105T3. 106T3. 107T3. 108T3. 109T3. 110T3. 111T3. 112T3. 113T3. 114T3. 115T3. 116T3. 117T3. 118T3. 119T3. 120T3. 121T3. 122T3. 123T3. 124T3. 125T3. 126T3. 127T3. 128T3. 129T3. 130T3. 131T3. 132T3. 133T3. 134T3. 135T3. 136T3. 137T3. 138T3. 139T3. 140T3. 141T3. 142T3. 143T3. 144T3. 145T3. 146T3. 147T3. 148T3. 149T3. 150T3. 151T3. 152T3. 153T3. 154T3. 155T3. 156T3. 157T3. 158T3. 159T3. 160T3. 161T3. 162T3. 163T3. 164T3. 165T3. 166T3. 167T3. 168T3. 169T3. 170T3. 171T3. 172T3. 173T3. 174T3. 175T3. 176T3. 177T3. 178T3. 179T3. 180T3. 181T3. 182T3. 183T3. 184T3. 185T3. 186T3. 187T3. 188T3. 189T3. 190T3. 191T3. 192T3. 193T3. 194T3. 195T3. 196T3. 197T3. 198T3. 199T3. 200T3. 201T3. 202T3. 203T3. 204T3. 205T3. 206T3. 207T3. 208T3. 209T3. 210T3. 211T3. 212T3. 213T3. 214T3. 215T3. 216T3. 217T3. 218T3. 219T3. 220T3. 221T3. 222T3. 223T3. 224T3. 225T3. 226T3. 227T3. 228T3. 229T3. 230T3. 231T3. 232T3. 233T3. 234T3. 235T3. 236T3. 237T3. 238T3. 239T3. 240T3. 241T3. 242T3. 243T3. 244T3. 245T3. 246T3. 247T3. 248T3. 249T3. 250T3. 251T3. 252T3. 253T3. 254T3. 255T3. 256T3. 257T3. 258T3. 259T3. 260T3. 261T3. 262T3. 263T3. 264T3. 265T3. 266T3. 267T3. 268T3. 269T3. 270T3. 271T3. 272T3. 273T3. 274T3. 275T3. 276T3. 277T3. 278T3. 279T3. 280T3. 281T3. 282T3. 283T3. 284T3. 285T3. 286T3. 287T3. 288T3. 289T3. 290T3. 291T3. 292T3. 293T3. 294T3. 295T3. 296T3. 297T3. 298T3. 299T3. 300T3. 301T3. 302T3. 303T3. 304T3. 305T3. 306T3. 307T3. 308T3. 309T3. 310T3. 311T3. 312T3. 313T3. 314T3. 315T3. 316T3. 317T3. 318T3. 319T3. 320T3. 321T3. 322T3. 323T3. 324T3. 325T3. 326T3. 327T3. 328T3. 329T3. 330T3. 331T3. 332T3. 333T3. 334T3. 335T3. 336T3. 337T3. 338T3. 339T3. 340T3. 341T3. 342T3. 343T3. 344T3. 345T3. 346T3. 347T3. 348T3. 349T3. 350T3. 351T3. 352T3. 353T3. 354T3. 355T3. 356T3. 357T3. 358T3. 359T3. 360T3. 361T3. 362T3. 363T3. 364T3. 365T3. 366T3. 367T3. 368T3. 369T3. 370T3. 371T3. 372T3. 373T3. 374T3. 375T3. 376T3. 377T3. 378T3. 379T3. 380T3. 381T3. 382T3. 383T3. 384T3. 385T3. 386T3. 387T3. 388T3. 389T3. 390T3. 391T3. 392T3. 393T3. 394T3. 395T3. 396T3. 397T3. 398T3. 399T3. 400T3. 401T3. 402T3. 403T3. 404T3. 405T3. 406T3. 407T3. 408T3. 409T3. 410T3. 411T3. 412T3. 413T3. 414T3. 415T3. 416T3. 417T3. 418T3. 419T3. 420T3. 421T3. 422T3. 423T3. 424T3. 425T3. 426T3. 427T3. 428T3. 429T3. 430T3. 431T3. 432T3. 433T3. 434T3. 435T3. 436T3. 437T3. 438T3. 439T3. 440T3. 441T3. 442T3. 443T3. 444T3. 445T3. 446T3. 447T3. 448T3. 449T3. 450T3. 451T3. 452T3. 453T3. 454T3. 455T3. 456T3. 457T3. 458T3. 459T3. 460T3. 461T3. 462T3. 463T3. 464T3. 465T3. 466T3. 467T3. 468T3. 469T3. 470T3. 471T3. 472T3. 473T3. 474T3. 475T3. 476T3. 477T3. 478T3. 479T3. 480T3. 481T3. 482T3. 483T3. 484T3. 485T3. 486T3. 487T3. 488T3. 489T3. 490T3. 491T3. 492T3. 493T3. 494T3. 495T3. 496T3. 497T3. 498T3. 499T3. 500T3. 501T3. 502T3. 503T3. 504T3. 505T3. 506T3. 507T3. 508T3. 509T3. 510T3. 511T3. 512T3. 513T3. 514T3. 515T3. 516T3. 517T3. 518T3. 519T3. 520T3. 521T3. 522T3. 523T3. 524T3. 525T3. 526T3. 527T3. 528T3. 529T3. 530T3. 531T3. 532T3. 533T3. 534T3. 535T3. 536T3. 537T3. 538T3. 539T3. 540T3. 541T3. 542T3. 543T3. 544T3. 545T3. 546T3. 547T3. 548T3. 549T3. 550T3. 551T3. 552T3. 553T3. 554T3. 555T3. 556T3. 557T3. 558T3. 559T3. 560T3. 561T3. 562T3. 563T3. 564T3. 565T3. 566T3. 567T3. 568T3. 569T3. 570T3. 571T3. 572T3. 573T3. 574T3. 575T3. 576T3. 577T3. 578T3. 579T3. 580T3. 581T3. 582T3. 583T3. 584T3. 585T3. 586T3. 587T3. 588T3. 589T3. 590T3. 591T3. 592T3. 593T3. 594T3. 595T3. 596T3. 597T3. 598T3. 599T3. 600T

- Pod = 2712 外置のCPUとメモリを共有するサービス

- Cluster IP : 10.20.1.10

- Node Port : 各ポートは外部IPアドレスとポートのIPと23

- Load balancer: 1-ト = 7-7/37セズ 37セズ