

Docker/Kubernetes 実践コンテナ開発入門

2019-08-27 更新 第5刷版

お詫びと訂正（正誤表）

下記の誤りがありました。ご迷惑をおかけいたしました。

P.37 # 2.1 のコンテナ停止の説明

誤 コマンド停止させられます。
正 コマンドで停止させられます。

P.37 # 2.1.1 の停止コマンドの例

誤 `docekr stop $(docker container ls -q)`
正 `docker stop $(docker container ls -q)`

P.46 # 2章の脚注 21

誤 `exmaple/echo`
正 `example/echo`

P.75 # 2.6.1 の最初のリストのjenkinsのイメージ名

書籍中で指定したイメージ名、バージョンでは動作しなくなっていました。
jenkinsci/jenkins:2.142-slim を利用してください。

誤 `image: jenkins:latest`

正 `image: jenkinsci/jenkins:2.142-slim`

P.78 # 2.6.3 の最初のリストの jenkins のイメージ名

書籍中で指定したイメージ名，バージョンでは動作しなくなっていました。

下記の通り修正してください。

```
services:
  master:
    container_name: master
    image: jenkinsci/jenkins:2.142-slim
    ports:
      - 8080:8080
    volumes:
      - ./jenkins_home:/var/jenkins_home
    links:
      - slave01

  slave01:
    container_name: slave01
    image: jenkinsci/ssh-slave
    environment:
      - JENKINS_SLAVE_SSH_PUBKEY=ssh-rsa AAAAB3NzaC1yc2EA.....
```

P.80 # 2.6.3 の「最終調整」の実行例

プラグインのバージョンが上がったことで書籍の通りには動作しなくなりました。

誤 「秘密鍵」は Jenkins マスター上の ~/.ssh からを選択します。

正 「秘密鍵」にはホストの ./jenkins_home/.ssh/id_rsa の内容を貼り付けます。

P.86 # 3.1.1 の Docker イメージビルド時の表示

誤 Step 7/7 : CMD ["touch", "/etc/cron.d/example", "&&", "cron", "-f"]

正 Step 7/7 : CMD ["cron", "-f"]

■P.86 # 3.1.1 の脚注 2 `docker stop` とすべき箇所が、`docekr stop` になっていました。

P.98 # 3.4.2 の最後の実行例

不要なバッククォートが残っていました。

誤	<code>docker container run -v `\${PWD}`:/tmp</code>
正	<code>docker container run -v \${PWD}:/tmp</code>

P.105 # 3.5.2 の説明文

誤	Docker イメージは registry コンテナに push してある
正	Docker イメージは registry コンテナに push してある

P.109 # 3.5.3 の Stack のサブコマンドに関する表

`rm` と `services` で項目が逆になっていました。

誤	<code>rm</code> Stack 内の Service 一覧を表示する
正	<code>rm</code> デプロイされている Stack を削除する

誤	<code>services</code> デプロイされている Stack を削除する
正	<code>services</code> Stack 内の Service 一覧を表示する

P.129 # 4.2.5 のスクリプト名の誤り

誤	<code>dd-server-id.sh</code> を <code>docker-entrypoint.sh</code> の前に実行できます。
正	<code>add-server-id.sh</code> を <code>docker-entrypoint.sh</code> の前に実行できます。

P.131 # 4.2.6 の `todo-mysql.yml` の表記

`MYSQL_ROOT_PASSWORD` を文中で二回設定していました。正しくは一回です。

誤

```
environment:
  MYSQL_MASTER_HOST: master
  MYSQL_ROOT_PASSWORD: gihyo
  MYSQL_DATABASE: tododb
  MYSQL_USER: gihyo
  MYSQL_PASSWORD: gihyo
  MYSQL_ROOT_PASSWORD: gihyo
  MYSQL_REPL_USER: repl
  MYSQL_REPL_PASSWORD: gihyo
```

正

```
environment:
  MYSQL_MASTER_HOST: master
  MYSQL_ROOT_PASSWORD: gihyo
  MYSQL_DATABASE: tododb
  MYSQL_USER: gihyo
  MYSQL_PASSWORD: gihyo
  MYSQL_REPL_USER: repl
  MYSQL_REPL_PASSWORD: gihyo
```

P.142 # 4.3.6 のリスト

ファイル名が誤っていました。

誤 todomysql_master

正 **todo_mysql_master**

誤 todomysql_slave

正 **todo_mysql_slave**

正しいリストの全体像を以下に掲載します。

```
version: "3"
services:
  api:
    image: registry:5000/ch04/todoapi:latest
    deploy:
      replicas: 2
```

```

environment:
  TODO_BIND: ":8080"
  TODO_MASTER_URL: "gihyo:gihyo@tcp(todo_mysql_master:3306)/tododb?parseTime=true"
  TODO_SLAVE_URL: "gihyo:gihyo@tcp(todo_mysql_slave:3306)/tododb?parseTime=true"
networks:
  - todoapp
networks:
  todoapp:
    external: true

```

P.146 # 4.4.1 の「バックエンドサーバの振り分けの設定値」コード部分と続く解説動作はしますが fails のつづりを誤っていました。

```

誤 BACKEND_MAX_FAILES
正 BACKEND_MAX_FAILS

```

P.150 # 4.4.3 のコード部分の設定値動作はしますが fails のつづりを誤っていました。

```

誤 BACKEND_MAX_FAILES
正 BACKEND_MAX_FAILS

```

P.156 # 4.5.3 の nuxt.conf.tpl の error_log の設定

```

誤 error_log /var/log/nginx/assets_error.log

正 error_log /var/log/nginx/assets_error.log

```

P.156 # 4.5.3 の Dockerfile-nuxt

本来バックスラッシュを挿入すべき箇所が v になっていました。また CMD 以降の指定の記法が nginx のものと異なっていました。正しいコードは下記です。コメントは実際には実行時にエラーになるので適宜外して実行してください。

```
FROM nginx:1.13
```

```
RUN apt-get update
```

```

RUN apt-get install -y wget
RUN wget https://github.com/progrium/entrykit/releases/download/v0.4.0/entrykit_0.4.0_linux_x86_64.t
RUN tar -xvzf entrykit_0.4.0_linux_x86_64.tgz
RUN rm entrykit_0.4.0_linux_x86_64.tgz
RUN mv entrykit /usr/local/bin/
RUN entrykit --symlink

RUN rm /etc/nginx/conf.d/*
COPY etc/nginx/nginx.conf.tpl /etc/nginx/
COPY etc/nginx/conf.d/ /etc/nginx/conf.d/

ENTRYPOINT [ \
  "render", \
    "/etc/nginx/nginx.conf", \
    "--", \
  "render", \
    "/etc/nginx/conf.d/upstream.conf", \
    "--", \
  "render", \
    "/etc/nginx/conf.d/nuxt.conf", \ # <-- public.conf から nuxt.conf に変更
    "--" \
]

CMD ["nginx", "-g", "daemon off;"]

```

P.158 # 4.5.4 のコード部分の設定値
動作はしますが fails のつづりを誤っていました。

誤 BACKEND_MAX_FAILES
正 BACKEND_MAX_FAILS

P.160 # 4.5.4 の docker コマンド実行例の 1 行目

誤 docker \container exec -it manager
正 docker container exec -it manager \

P.194 # 5.10 の実行例

-f オプションの位置が誤っていました。

誤 `kubectl -f apply`

正 `kubectl apply -f`

P.194 # 5.10.1 の冒頭の実行例

誤 `$ kubectl apply -f simple-deployment.yaml`

正 `$ kubectl apply -f simple-service.yaml`

P.211 # 6.3.3 の yml の反映ファイル名

誤 `mysql-master.yamb`

正 **`mysql-master.yaml`**

P.213~214 # 6.4 todo-api.yaml の内容

Deployment.spec.template.metadata.labels の指定で本来 app とすべき箇所が name になっていました。

正しいのは下記のコードです。

```
apiVersion: v1
kind: Service
metadata:
  name: todoapi
  labels:
    app: todoapi
spec:
  selector:
    app: todoapi
  ports:
    - name: http
      port: 80
```

```

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: todoapi
  labels:
    app: todoapi
spec:
  replicas: 2
  selector:
    matchLabels:
      app: todoapi
  template:
    metadata:
      labels:
        app: todoapi
    spec:
      containers:
        - name: nginx
          image: gihyodocker/nginx:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 80
          env:
            - name: WORKER_PROCESSES
              value: "2"
            - name: WORKER_CONNECTIONS
              value: "1024"
            - name: LOG_STDOUT
              value: "true"
            - name: BACKEND_HOST
              value: "localhost:8080"
        - name: api
          image: gihyodocker/todoapi:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
          env:
            - name: TODO_BIND
              value: ":8080"

```



```
- name: TODO_MASTER_URL
  value: "gihyo:gihyo@tcp(mysql-master:3306)/tododb?parseTime=true"
- name: TODO_SLAVE_URL
  value: "gihyo:gihyo@tcp(mysql-slave:3306)/tododb?parseTime=true"
```

P.218 # 6.5 のリスト

6.5 の最下部の `kubectl apply -f ingress.yaml` は正しくは 6.6 のファイル表記の下に配置されます。

P.227 # 7.1 の脚注 4

誤 失敗した iPod
正 失敗した **Pod**

P.235 # 7.2 の作業の流れに関する説明文

7.2.1 で実際に行う操作と矛盾が生じていました。

誤 以後、ローカル Kubernetes 環境を利用して実際に RBAC 関連リソースを作成し、認証ユーザーで認証を行った上での Kubernetes の操作を行います。次に、ServiceAccount を利用した Pod からの Kubernetes API 利用について解説します。

正 以後、パブリッククラウドを想定して実際に RBAC 関連リソースを作成し、認証ユーザーで認証を行った上での Kubernetes の操作を行います。次に、ServiceAccount を利用した Pod からの Kubernetes API 利用について解説します。

P.235 # 7.2.1 のリソースに関する表の上の説明文

本来はパブリッククラウドで検証する部分でしたが、誤ってローカル Kubernetes 環境としていました。

誤 これらのリソースを使用して、ローカル Kubernetes 環境で認証ユーザーの権限制御を行ってみましょう。

正 これらのリソースを使用して、認証ユーザーの権限制御を行ってみましょう。

P.293 # 8.1.5 の yml を適用する部分の説明

誤 次のように `elasticsearch.yaml` を反映します。Kibana の Service は 'kube-system' に配置されているので、'-n' オプションで Namespace である 'kube-system' を指定します。

正 次のように `kibana.yaml` を反映します。

P.310 # 8.3.2 の下の見出し

誤 Node 障害の際に Kubernetes の挙動
正 **Node 障害時の Kubernetes の挙動**

P.312 # 8.3.2 の脚注 13

誤 Pod AntiAaffinity
正 **Pod AntiAffinity**

P.321 # 9.1.1 のオートスケールの説明に関する文章

誤 新規のノードが追加されるます。
正 新規のノードが追加されます。

P.334 # 9.2.3 の「パッケージマネージャ apk を操作する」の apk del に関する解説
本来リスト外に出るべき解説箇所が、誤ってリスト内に入っていました。
下記部分はリスト外の解説分です。

apk del

apk del ではインストールされているパッケージをアンインストールします。apk add -virtual などと
組み合わせて使わなくなったパッケージの削除に用います。

P.352 # 10.2.1 のタグ付けのイメージ名

誤 ch09/jq:latest
正 **ch10/jq:latest**

P.358 # 10.3.1 の最後のコマンド実行例

誤 ch09-locust.yml
正 **ch10-locust.yml**

P.398 # helm package の書式

誤 helm pacakge [options] Chart へのパス
正 helm **package** [options] Chart へのパス
