

■ ハンズオン

1. Docker のインストール

① システムの Update

```
$ sudo apt-get update
```

② docker をインストール

```
$ sudo apt-get install docker.io
```

③ ログインユーザー” ubuntu” を docker グループに追加

```
$ sudo gpasswd -a ubuntu docker
```

④ 再起動

```
$ sudo reboot
```

2. Docker の基本的な操作

① docker hub の参照

<https://hub.docker.com/>

② DockerHub の Explorer リンクからアプリケーションを参照

③ アプリケーション検索

④ tag について

3. docker 基本コマンド

① docker の起動中のコンテナ一覧出力

```
$ docker ps
```

② 停止中も含めた docker コンテナの一覧出力

```
$ docker ps -a
```

③ ローカルにある docker イメージの一覧を出力

```
$ docker images
```

(参考) その他よく使う docker コマンドの例

※ docker コマンド.pdf もあわせて参照

docker pull	## 外部レジストリからイメージを取得
docker run	## イメージから docker を起動
docker stop	## 起動中の docker コンテナを停止
docker rm	## docker コンテナを削除
docker rmi	## ローカルの docker イメージを削除

4. WordPress のインストール

① MySQL のインストール

- i. 以下のコマンドでインストール

```
$ docker run --name <任意の名前> -v <任意のボリューム>
:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=<任意のパスワード>
-p <任意のポート>:3306 -d mysql:<任意のタグ>
```

- ii. 例:

```
$ docker run --name mysql -v /my/custom:/etc/mysql/conf.d -e
MYSQL_ROOT_PASSWORD=Macnica123 -p 3306:3306 -d
mysql:latest
```

② WordPress のインストール

- i. 以下のコマンドでインストール

```
$ docker run --name <任意の名前> -e WORDPRESS_DB_HOST=<
MySQL のホスト名 or IP>:<MySQL のポート> -e
WORDPRESS_DB_USER=<任意の DB ユーザー> -e
WORDPRESS_DB_PASSWORD=<任意の DB> -p <任意のポート>
:<任意のポート>:80 -d wordpress:<任意のタグ>
```

- ii. 例:

```
$ docker run --name wordpress -e
WORDPRESS_DB_HOST=1.1.1.1:3306 -e
WORDPRESS_DB_USER=root -e
WORDPRESS_DB_PASSWORD=Macnica123 -p 80:80 -d
wordpress:latest
```

※ 上記の「1.1.1.1」の箇所には docker ホストのプライベート IP アドレス(ifconfig で eth0 に設定されたアドレスで、172 から始まるもの)を代入

③ WordPress のサイトにアクセス

http://<WordPress の Docker ホスト名 or IP>

④ docker コンテナの中に入る

- i. 次のコマンドでコンテナ ID を確認
`$ docker ps`
- ii. 次のコマンドで WordPress のコンテナ内に入る
`docker exec -it <コンテナ ID> /bin/bash`
- iii. コンテナから Docker ホストに戻る
`$ exit`

⑤ 既存イメージの削除

- i. 次のコマンドでコンテナ ID を確認
`$ docker ps`
- ii. 次のコマンドで WordPress と MySQL のコンテナを停止する
`$ docker stop <コンテナ ID>`
- iii. 次のコマンドで WordPress と MySQL のコンテナを削除する
`$ docker rm <コンテナ ID>`
- iv. 次のコマンドでローカルの docker イメージを確認
`$ docker images`
- v. 次のコマンドで WordPress と MySQL のイメージを削除する
`$ docker rmi <イメージ名またはイメージ ID>`

5. Rancher サーバー(コンテナ版)の構築について

① 次のコマンドで Rancher サーバーを起動

```
$ docker run --restart=always -d -p 80:8080 rancher/server:latest
```

- シングルインスタンスで利用の場合、上記だけで最新版 Rancher が構築されます。(今回はリソースの関係上、あらかじめ用意したサーバーを利用します)

6. Rancher サーバーへのマシン登録

① Rancher サーバーへのアクセス

<http://52.199.242.86>

※ ログイン情報は別紙参照

※ ログインスクリーンで言語の指定が可能



② Rancher サーバーにて、ホストを追加(Add Host)をクリック



③ Custom ホスト登録用のコマンドをコピー



④ 自分の docker ホストに SSH でログインし、コピーしたコマンドを貼り付けて実行する

※ 登録完了まで数分を要する場合があります。

- ⑤ 登録完了後、インフラストラクチャ>ホストに登録したホストが表示される



7. Rancher のコミュニティカタログから WordPress をインストール

- ① カタログをクリックし、検索ボックスに Wordpress と入力し、表示されたメニューで” 詳細を見る ” ボタンをクリックする



② 必要情報を入力して、起動(デフォルトでも可)

v0.1-educaas1
デプロイするテンプレートのバージョンを選択してください

新しいスタック

名前*

wordpress

① 必要に応じて変更

設定オプション

Public Port*

80

public port to access the work

② 必要に応じて変更

作成後にサービスを起動

プレビュー

起動

③ 起動をクリック

③ 構築状況の確認

test-Default

スタック

カタログ

インフラストラクチャ

API

スタック: wordpress

サービスを追加

最新

Active

Active	サービス	1コンテナ
Active	サービス	1コンテナ

Active となれば起動完了

④ WordPress にアクセス

※ 通常は UI 上のポート番号のクリックでアクセスできますが、AWS 環境では Docker ホストのプライベート側 IP がリンクに紐づいているため、Global 側の IP を指定しなおす必要があります。

`http://<docker ホストの IP>:<指定したポート>`

上記で WordPress 初期設定画面が開くことを確認

8. WordPress のコンテナにアクセス

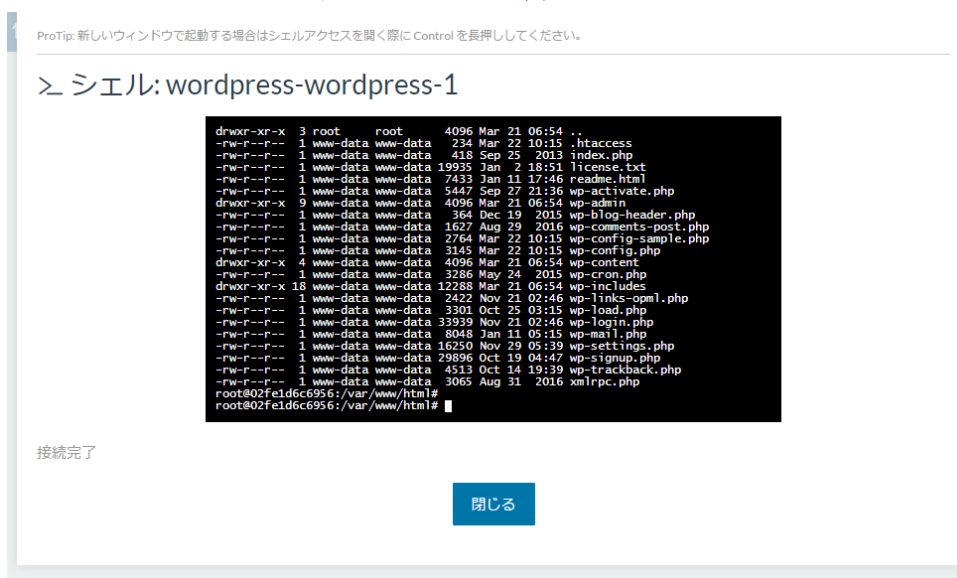
① WordPress の詳細画面に入る



② メニューより” シェルの実行” をクリック



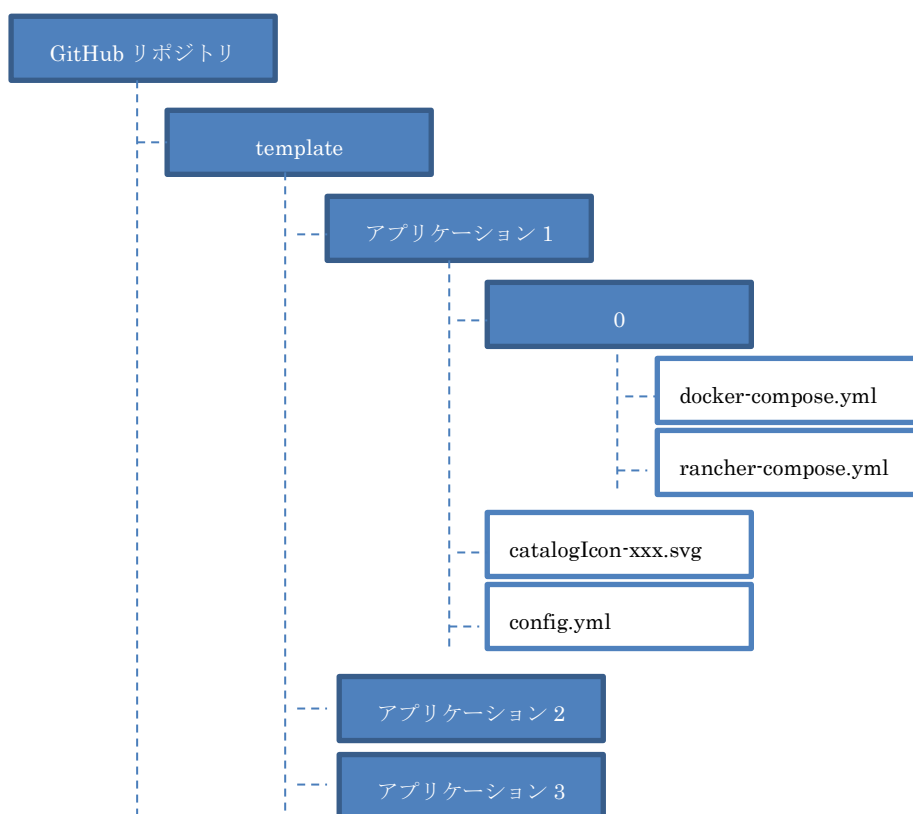
③ コンテナ内のシェルが実行されることを確認



9. Rancher Private カタログの作成(MySQL の登録)(デモ)

- ① GitHub にアクセスし、任意の名称のリポジトリを作成
- ② リポジトリ配下に `template` ディレクトリを作成
- ③ `template` ディレクトリ配下に任意のアプリケーション名のディレクトリを作成する
- ④ アプリケーション名のディレクトリ配下には、次のように設定ファイルが配置される

※ 設定ファイルのサンプル (MySQL) は、OwnCloud 中の `template>MySQL`



※ アプリケーションディレクトリ配下の数字(0)のディレクトリは、複数のバージョンを提供する際にバージョンごとに作成します。数値は 0、1、2・・・というように、0 から一つずつ数字を加算していきます。

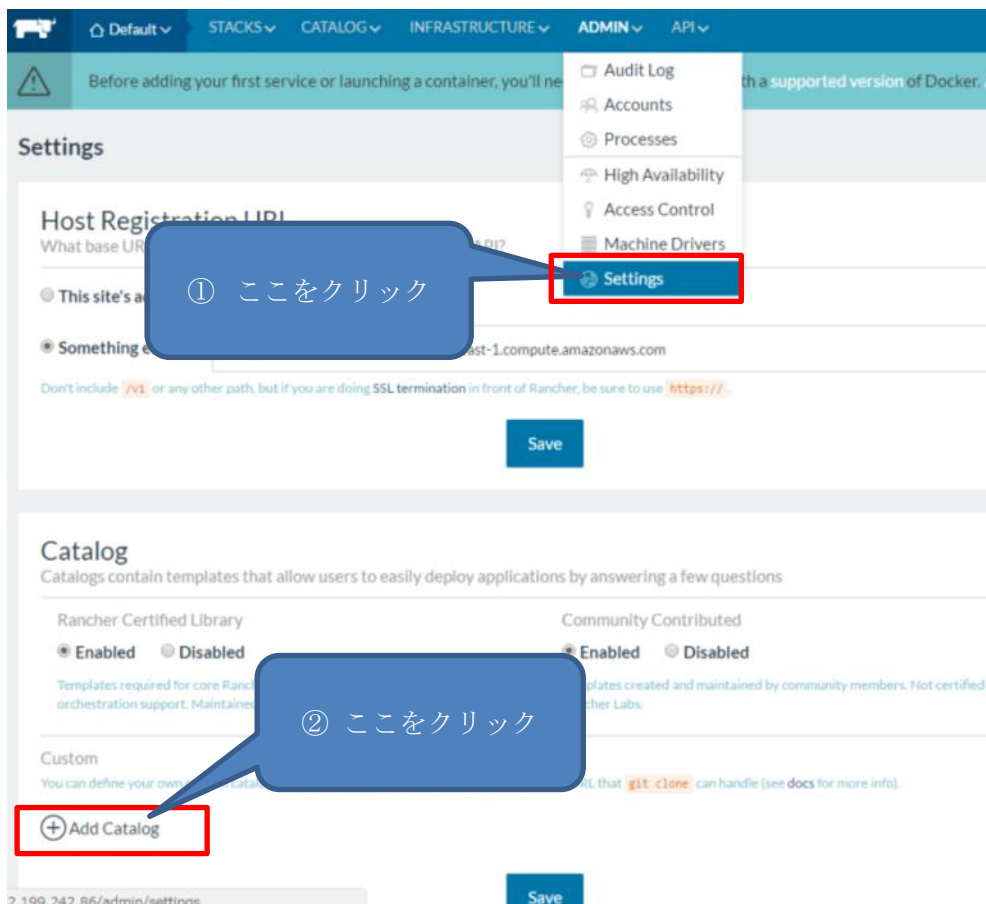
10. Rancher サーバーへのカタログ登録

- ① Templates ディレクトリの GitHub リポジトリ URL をコピーし、Rancher に admin 権限でログインする

(今回の Rancher サーバーでは、admin:Macnica123)

リポジトリ URL の例: <https://github.com/container-tech/templates.git>

- ② Rancher で ADMIN > Settings を開き、Add Catalog をクリックする



- ③ 任意の名称と GitHub のリポジトリ URL、ブランチを入力し、Save をクリック



- ④ カタログに MySQL が追加されていることを確認



11. 各 yml ファイルおよび svg ファイルの解説

- ① catalogIcon-xxx.svg → カタログのボタンに表示するアイコンの画像ファイル



- ② config.yml → カタログの全般を設定するファイル

```
name: "MySQLServer"
description: "MySQL Servr です。"
version: "5.7"
category: "SQLDBServer"
maintainer:
license:
projectURL: "https://hub.docker.com/_/mysql/"
```



- ③ docker-compose.yml → 通常の docker-compose ファイルと同じ。
Rancher-compose と組み合わせて変数を利用することが可能。

```
MySQLServer:
```

```
  ports:
```

```
    - ${public_port}:3306
```

ポートフォワード設定 (変数は rancher-compose で定義)

```
  environment:
```

DockerHub に記載の環境変数 (変数は rancher-compose で定義)

```
    MYSQL_ROOT_PASSWORD: ${root_password}
```

```
    MYSQL_DATABASE: ${db_name}
```

```
    MYSQL_USER: ${db_username}
```

```
    MYSQL_PASSWORD: ${db_password}
```

```
  labels:
```

```
    io.rancher.container.pull_image: always
```

```
  tty: true
```

```
  image: mysql:5.7
```

レジストリ名とタグ

```
  stdin_open: true
```

④ rancher-compose.yml □ rancher の UI での表示や設定を定義するファイル

```
MySQLServer:
.catalog:
  name: "MySQLServer"
  version: "5.7"
  description: "MySQL Server."
  questions:
    - variable: root_password
      description: "MySQL root password"
      label: "MySQL Root Password"
      required: true
      default: "password"
      type: "string"
    - variable: db_name
      description: "MySQL Database Name"
      label: "MySQL Database Name"
      required: true
      default: "DB"
      type: "string"
    - variable: db_username
      description: "MySQL Username"
      label: "MySQL Username"
      required: false
      default: "CMS_USER"
      type: "string"
    - variable: db_password
      description: "MySQL password for the above user"
      label: "MySQL Password"
      required: false
      default: "password"
      type: "string"
    - variable: public_port
      description: "Public port to access the MySQL Server"
      label: "MySQL Port"
      required: true
      default: "3306"
      type: "int"
```

Rancher UI に表示するバージョン情報

Rancher UI に表示する設定項目と変数指定

[参考情報]

- コミュニティカタログの GitHub 上のリポジトリ

<https://github.com/rancher/community-catalog/tree/master/templates>

Docker のプライベートレジストリインストール

<https://docs.docker.com/registry/>

```
$ docker run -d -p 5000:5000 --restart always --name registry registry:latest
```