■コンテナ技術(Docker)

今回、AWSのEC2(Linux)を使用してコンテナ技術(Docker)の演習を行います。

＊CentOS8ではDockerコマンドは推奨されていないため、おすすめはしないです。Windowsで使用する場合はWSL2を利用したDocker for Windowsを使用します。

■Dockerのインストール(Docker)

Dockerのインストールは各OSにより異なります。以下のURLを参照してインストールを行ってください。

参照URL：https://hub.docker.com/search?q=&type=edition&offering=community

|  |
| --- |
| sudo yum -y install docker　・・・　Dockerのインストール(Amazon Linux 2023 AMI)  ＊yum -y install docker　　　　　・・・　Dockerのインストール(CentOS)  ＊apt -y install docker　　　　　・・・　Dockerのインストール(debian系)  sudo systemctl start docker　　　　・・・　Dockerの起動  sudo systemctl enable docker　　　・・・　OSの起動時にDockerを起動するように設定 |

＊AWSで実行する時の注意点

|  |
| --- |
| sudo usermod -a -G docker ec2-user　　　・・・　sudoコマンドなしでdockerコマンドを実行する  ＊変更後に一度、ログアウトします。その後、再度ログインしてください。 |

sudo chown root:docker /var/run/docker.sock

sudo chmod 660 /var/run/docker.sock

■Docker実習例1

Dockerのnginxイメージをダウンロードして、コンテナを起動します。

|  |
| --- |
| docker pull イメージファイル名　・・・イメージをダウンロード(イメージがローカルにない場合はダウンロードされます)  <例>docker pull nginx  docker images　・・・ドッカーイメージを表示 (List)  docker history イメージ名　・・・　Dockerイメージの履歴を表示 (History)  <例>docker history nginx  docker rmi イメージ名　・・・Dockerイメージを削除 (Delete Docker Image)  <例>docker rmi nginx  docker run -p（ポート） ホスト:コンテナ側 --name コンテナ名 イメージ名  <例> docker run -p 80:80 --name nginx nginx  docker ps ・・・起動中のコンテナ一覧を表示（起動中のコンテナのみ）  \*docker ps --all(停止中のコンテナも含めて表示)  docker rm コンテナID　・・・　停止しているコンテナを削除  <例>docker rm nginx  docker run -p 80:80 --detach nginx ・・・コンテナをバックグランドで起動  <例>docker run --detach -p 80:80 --name nginx nginx  ＊docker stop ・・・停止  docker logs コンテナID　・・・コンテナのログを表示 |

■コンテナの確認

コマンドを使用して、確認をします(Webブラウザでも確認できます)。

|  |
| --- |
| curl localhost |

正常に動作していれば、nginxのトップページ(index.html)が表示されます。

■Docker実習2

ここでは、起動したコンテナにアクセスして、index.htmlページを変更します。

|  |
| --- |
| docker exec -it コンテナID bash　・・・　作業中のコンテナの中にシェル（bash）で入る  <例> docker exec -it nginx bash |

次のコマンドでindex.htmlを変更します。

|  |
| --- |
| echo "Hello World!!" > /usr/share/nginx/html/index.html  exit　　・・・　コンテナから抜ける |

■コンテナの確認

コマンドを使用して、確認をします(Webブラウザでも確認できます)。

|  |
| --- |
| curl localhost |

＊表示されるページが変更されていればOKです。

■Docker実習3

ここでは、変更したコンテナから新しくイメージを作成します。「hello\_world」というイメージを作成してコンテナを起動します。

|  |
| --- |
| docker commit コンテナ名 イメージ名  <例>docker commit nginx hello\_world　　　　・・・　hello\_worldというイメージを作成する  <例>docker run -p 8080:80 --rm -d --name hello\_world hello\_world  　　・・・　コンテナの起動、別のコンテナが80ポートを使用しているの8080を指定 |

■コンテナの確認

コマンドを使用して、確認をします(Webブラウザでも確認できます)。

|  |
| --- |
| curl localhost：8080 |

＊実習2で表示されたページがでればOKです。

■Docker実習4

ここでは、Dockerfileを使用してDockerの設定を行います。Dockerfileはイメージを作成する設計書のようなものです。次にDockerfileの一例になります。

|  |
| --- |
| FROM nginx:latest　　　　　　　　　　・・・イメージを指定  WORKDIR /usr/share/nginx/html　　　・・・起動するファシルのパス  COPY index.html index.html　　　　　・・・ホストからイメージにファイルをコピーする  RUN apt update && apt install -y curl　　・・・シェルコマンドを実行 |

Dockerfileからイメージを作成します。

|  |
| --- |
| docker build --tag dockerfile\_hello(イメージ名) .(パス：Dockerfileがあるパス)  <例> docker build --tag dockerfile\_hello **.** |

＊ENV　・・・環境変数のようなもの

CMD ・・・コンテナ起動に最初に起動するコマンド

＊1行ごとに別のレイヤが作成されます。

作成したイメージからコンテナを起動します。

|  |
| --- |
| docker run -d -p 9090:80 --name hello dockerfile\_hello |

■コンテナの確認

コマンドを使用して、確認をします(Webブラウザでも確認できます)。

|  |
| --- |
| curl localhost：9090 |

＊作成したでページが表示でればOKです。

■Dockerの削除

|  |
| --- |
| docker system prune　・・・　停止中のコンテナと無名のイメージを削除  docker rm -vf $(docker ps -a -q)　　・・・　起動中のコンテナ含め全コンテナを削除  docker rmi -f $(docker images -a -q)　・・・　全てのイメージを削除 |

■Docker Compose

Docker Composeのインストールは次のように行います。Docker ComposeはYAML形式のファイルで、複数のコンテナを一括管理ができます。

参照URL https://matsuand.github.io/docs.docker.jp.onthefly/compose/install/

|  |
| --- |
| sudo mkdir -p /usr/local/lib/docker/cli-plugins ・・・　インストール先のディレクトリ作成  sudo curl -SL https://github.com/docker/compose/releases/download/v2.4.1/docker-compose-linux-x86\_64 -o /usr/local/lib/docker/cli-plugins/docker-compose  ・・・　ダウンロードとインストール  sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose　　・・・　実行権限の設定  sudo chmod 666 /var/run/docker.sock |

■Docker Composeの確認

|  |
| --- |
| docker compose version ・・・　バージョンが表示されればインストールは成功 |

■Docker Composeで複数のコンテナを起動させる。

|  |
| --- |
| ・docker composeのYAMLに定義されたサービスを起動する  docker compose --file YAMLファイル up ・・・サービスを起動する。  docker compose -f YAMLファイル down・・・サービスを停止  ＊YAMLファイルが1つの場合は、-fは必要はない  <例>docker compose -f docker-compose.yml up　　・・・　コンテナの起動  docker compose -f YAMLファイル logs --follow(リアルタイム) 複数コンテナのログをまとめてみる  <例>docker compose -f docker-compose.yml logs -f  docker compose -f docker-compose.yml down  　・・・　docker-composeのYAMLに定義されたサービスを停止 |

\*Docker Composeで定義されたコンテナ同士で通信できるようになっています。

■docker-compose.ymlの例

|  |
| --- |
| version: '3.9' ・・・　バージョンの指定(dockerのバージョンにより異なります)  services :  php:　　・・・　サービス名(コンテナに相当する)と詳細  image: php:8.1-apache　　・・・　イメージ名  volumes:  - ./htdocs:/var/www/html　　　・・・　ボリュームの指定  ports:  - 8080:80　　　・・・　ポート番号の指定  container\_name: webdev　　・・・　コンテナ名  restart: always |

　＊htdocsの格納したファイルはコンテナの/var/www/htmlにリンクされる。

　＊docker-composeを実行する前にhtdocsは作成しておくこと。

■htdocsにファイルを格納

・index.html

|  |
| --- |
| <h1>Hello Docker!!</h1> |

表示例(ブラウザで確認する・・・http://パブリックIPv4アドレス:8080)



・test.php

|  |
| --- |
| <?php  phpinfo();  ?> |

表示例(ブラウザで確認する・・・http://パブリックIPv4アドレス:8080/test.php)

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明