

AWS ドキュメント (<https://docs.aws.amazon.com/index.html>) » Amazon EC2 Container Registry (<https://docs.aws.amazon.com/ecr/index.html>) » ユーザーガイド ([https://docs.aws.amazon.com/ja\\_jp/AmazonECR/latest/userguide/index.html](https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/index.html)) » Amazon ECR における Docker の基本

## Amazon ECR における Docker の基本

Docker は、Linux コンテナに基づいた分散アプリケーションの構築、実行、テスト、デプロイを可能にするテクノロジーです。Amazon ECR はマネージド型 AWS Docker レジストリサービスです。お客様は一般的な Docker CLI を使用してイメージをプッシュ、プル、管理できます。Amazon ECR 製品の詳細、主なお客様導入事例、FAQ については、[Amazon Elastic Container Registry 製品の詳細ページ \(http://aws.amazon.com/ecr\)](http://aws.amazon.com/ecr) を参照してください。

このガイドのドキュメントは、読者が Docker の概念と機能を基本的に理解していることを前提としています。Docker の詳細については、「[Docker とは \(http://aws.amazon.com/docker/\)](http://aws.amazon.com/docker/)」および「[Docker の概要 \(https://docs.docker.com/engine/docker-overview/\)](https://docs.docker.com/engine/docker-overview/)」を参照してください。

### トピック

- [Docker のインストール \(https://docs.aws.amazon.com/ja\\_jp/AmazonECR/latest/userguide/docker-basics.html#install\\_docker\)](https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/docker-basics.html#install_docker)
- [Docker イメージの作成 \(https://docs.aws.amazon.com/ja\\_jp/AmazonECR/latest/userguide/docker-basics.html#docker-basics-create-image\)](https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/docker-basics.html#docker-basics-create-image)
- [\(オプション\) イメージの Amazon Elastic Container Registry へのプッシュ \(https://docs.aws.amazon.com/ja\\_jp/AmazonECR/latest/userguide/docker-basics.html#use-ecr\)](https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/docker-basics.html#use-ecr)
- [次のステップ \(https://docs.aws.amazon.com/ja\\_jp/AmazonECR/latest/userguide/docker-basics.html#docker\\_next\\_steps\)](https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/docker-basics.html#docker_next_steps)

## Docker のインストール

### 注記

Docker をインストール済みの場合は、この手順をスキップして「[Docker イメージの作成 \(https://docs.aws.amazon.com/ja\\_jp/AmazonECR/latest/userguide/docker-basics.html#docker-basics-create-image\)](https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/docker-basics.html#docker-basics-create-image)」に進んでください。

Docker は、Ubuntu のような最新の Linux ディストリビューションから Mac OSX や Windows まで、さまざまなオペレーティングシステムで使用できます。特定のオペレーティングシステムに Docker をインストールする方法の詳細については、[Docker インストールガイド \(https://docs.docker.com/engine/installation/#installation\)](https://docs.docker.com/engine/installation/#installation) を参照してください。

Docker を使用するには、ローカルの開発システムすら必要ありません。Amazon EC2 をすでに使用している場合は、Amazon Linux インスタンスを起動し、Docker をインストールして使用し始めることができます。

## Amazon Linux インスタンスに Docker をインストールするには

1. Amazon Linux AMI でインスタンスを起動します。詳細については、*Linux インスタンス用 Amazon EC2 ユーザーガイド* の「[インスタンスの起動](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html) (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html>)」を参照してください。
2. インスタンスに接続します。詳細については、『*Linux インスタンス用 Amazon EC2 ユーザーガイド*』の「[Linux インスタンスへの接続](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html) (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html>)」を参照してください。
3. インスタンスでインストールされているパッケージとパッケージキャッシュを更新します。

```
sudo yum update -y
```

4. 最新の Docker Community Edition パッケージをインストールします。

```
sudo yum install -y docker
```

5. Docker サービスを開始します。

```
sudo service docker start
```

6. `ec2-user` を `docker` グループに追加すると、`sudo` を使用せずに Docker コマンドを実行できます。

```
sudo usermod -a -G docker ec2-user
```

7. ログアウトし、再びログインして、新しい `docker` グループアクセス権限を取得します。これは、現在の SSH ターミナルウィンドウを閉じて、新しいウィンドウでインスタンスに再接続することで達成できます。新しい SSH セッションには適切な `docker` グループ権限があります。

8. `ec2-user` が `sudo` を使用せずに Docker コマンドを実行できることを確認します。

```
docker info
```

### 注記

場合によっては、Docker デーモンにアクセスするための `ec2-user` に対するアクセス権限を提供するため、インスタンスを再起動する必要があります。次のエラーが表示された場合は、インスタンスを再起動してください:

```
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
```

## Docker イメージの作成

このセクションでは、シンプルなウェブアプリケーションの Docker イメージを作成し、ローカルシステムまたは EC2 インスタンスでテストしてから、コンテナレジストリ (Amazon ECR や Docker Hub など) にプッシュして、ECS タスク定義で使えるようにします。

### シンプルなウェブアプリケーションの Docker イメージを作成するには

1. Dockerfile という名前のファイルを作成します。Dockerfile は、Docker イメージに使用する基本イメージと、そのイメージにインストールして実行するものを記述するマニフェストです。Dockerfile の詳細については、「[Dockerfile リファレンス \(https://docs.docker.com/engine/reference/builder/\)](https://docs.docker.com/engine/reference/builder/)」を参照してください。

```
touch Dockerfile
```

2. 前の手順で作成した Dockerfile を編集し、以下のコンテンツを追加します。

```
FROM ubuntu:12.04

# Install dependencies
RUN apt-get update -y
RUN apt-get install -y apache2

# Install apache and write hello world message
RUN echo "Hello World!" > /var/www/index.html

# Configure apache
RUN a2enmod rewrite
RUN chown -R www-data:www-data /var/www
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

この Dockerfile は Ubuntu 12.04 イメージを使用します。RUN の手順により、パッケージキャッシュが更新され、ウェブサーバー用のいくつかのソフトウェアがインストールされてから、「Hello World!」のコンテンツがウェブサーバーのドキュメントルートに書き込まれます。EXPOSE 命令はコンテナ上のポート 80 を公開し、CMD 命令はウェブサーバーを起動します。

3. Dockerfile から Docker イメージを作成します。

### 注記

Docker の一部のバージョンでは、下に示す相対パスの代わりに、次のコマンドで Dockerfile への完全パスが必要になる場合があります。

```
docker build -t hello-world .
```

4. **docker images** を実行して、イメージが正しく作成されたことを確認します。

```
docker images --filter reference=hello-world
```

出力:

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 258MB	latest	e9ffedc8c286	4 minutes ago

5. 新しく構築されたイメージを実行します。-p 80:80 オプションは、コンテナ上の公開されたポート 80 をホストシステム上のポート 80 にマッピングします。**docker run** の詳細については、「[Docker run リファレンス](https://docs.docker.com/engine/reference/run/) (<https://docs.docker.com/engine/reference/run/>)」を参照してください。

```
docker run -p 80:80 hello-world
```

### 注記

Apache ウェブサーバーからの出力はターミナルウィンドウに表示されます。"Could not reliably determine the server's fully qualified domain name" メッセージは無視できます。

6. ブラウザーを開き、Docker を実行している、コンテナのホストサーバーを参照します。
- EC2 インスタンスを使用している場合、これはサーバーの **[Public DNS]** 値であり、SSH でインスタンスに接続するとき使用するアドレスと同じです。インスタンスのセキュリティグループでポート 80 上の受信トラフィックを許可していることを確認します。
  - Docker をローカルに実行している場合は、ブラウザで <http://localhost/> (<http://localhost/>) を参照します。
  - Windows または Mac コンピューターで **docker-machine** を使用している場合は、**docker-machine ip** コマンドを使用して Docker のホスト VirtualBox VM の IP アドレスを見つけ、**machine-name** を、使用中の Docker マシンの名前に置き換えます。

```
docker-machine ip machine-name
```

ウェブページに「Hello, World!」がstatement.

7. **[Ctrl + C]** キーを押して、Docker コンテナを停止します。

## (オプション) イメージの Amazon Elastic Container Registry へのプッシュ

Amazon ECR はマネージド型 AWS Docker レジストリサービスです。お客様は一般的な Docker CLI を使用してイメージをプッシュ、プル、管理できます。Amazon ECR 製品の詳

細、主なお客様導入事例、FAQ については、[Amazon Elastic Container Registry](https://aws.amazon.com/ecr) 製品の詳細ページ (<http://aws.amazon.com/ecr>) を参照してください。

## 注記

このセクションでは AWS CLI が必要です。AWS CLI がシステムにインストールされていない場合は、*AWS Command Line Interface ユーザーガイド* の「[AWS コマンドラインインターフェースのインストール](https://docs.aws.amazon.com/cli/latest/userguide/installing.html) (<https://docs.aws.amazon.com/cli/latest/userguide/installing.html>)」を参照してください。

## イメージにタグを付け、Amazon ECR にプッシュするには

1. hello-world イメージを保存する Amazon ECR リポジトリを作成します。出力の repositoryUri に注目してください。

```
aws ecr create-repository --repository-name hello-world
```

出力:

```
{
  "repository": {
    "registryId": "aws_account_id",
    "repositoryName": "hello-world",
    "repositoryArn": "arn:aws:ecr:us-east-1:aws_account_id:repository/hello-world",
    "createdAt": 1505337806.0,
    "repositoryUri": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world"
  }
}
```

2. 前のステップの repositoryUri の値で hello-world イメージにタグを付けます。

```
docker tag hello-world aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world
```

3. `aws ecr get-login --no-include-email` コマンドを実行して、レジストリ用の `docker login` 認証コマンド文字列を取得します。

## 注記

`get-login` コマンドは AWS CLI のバージョン 1.9.15 以降で利用できます。ただし、Docker の最新バージョン (17.06 以降) を使用している場合は、バージョン 1.11.91 以降をお勧めします。AWS CLI のバージョンは、`aws --version` コマンドを使用して確認できます。Docker のバージョン 17.06 以降を使用している場合は、`get-login` の後に `--no-include-email` オプションを含めます。Unknown options: --no-include-email エラーが返された場合は、AWS CLI の最新バージョンをインストールします。詳細については、*AWS Command Line Interface ユーザーガイド* の「[AWS コマンドラインインターフェースのインストール](#)」

(<https://docs.aws.amazon.com/cli/latest/userguide/installing.html>)」を参照してください。

```
aws ecr get-login --no-include-email
```

4. 前のステップで返された **docker login** コマンドを実行します。このコマンドは、12 時間有効な認証トークンを提供します。

### 重要

**docker login** コマンドを実行すると、システムの他のユーザーに対して、プロセスリスト表示 (**ps -e**) でコマンド文字列が表示されます。**docker login** コマンドには認証情報が含まれているため、システムの他のユーザーがこのようにして認証情報を表示し、それを使ってレポジトリへのプッシュおよびプル権限を取得するリスクがあります。安全なシステムを使用していない場合は、このリスクを考慮してください。-p **password** オプションを省略してインタラクティブにログインし、求められたときにパスワードを入力することを検討してください。

5. 前のステップの値 `repositoryUri` を使用して、Amazon ECR にイメージをプッシュします。

```
docker push aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world
```

## 次のステップ

Amazon ECR イメージの試用を終了したら、レポジトリを削除し、イメージストレージに対して課金されないようにします。

### 注記

このセクションでは AWS CLI が必要です。AWS CLI がシステムにインストールされていない場合は、*AWS Command Line Interface ユーザーガイド* の「[AWS コマンドラインインターフェースのインストール](https://docs.aws.amazon.com/cli/latest/userguide/installing.html) (<https://docs.aws.amazon.com/cli/latest/userguide/installing.html>)」を参照してください。

```
aws ecr delete-repository --repository-name hello-world --force
```

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.