



Amazon Elastic Container Registry

Amazon Elastic Container Registry Public

AWS Black Belt Online Seminar

Solution Architect
浅野佑貴
2021-Dec



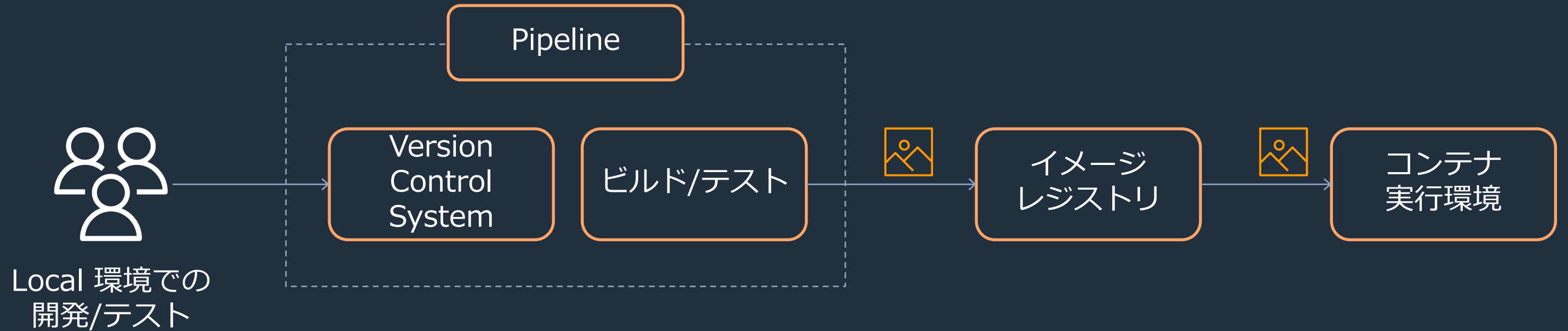
Agenda

- Amazon Elastic Container Registry(Amazon ECR)の概要
 - コンテナワークロードの基本的なワークフロー
 - Amazon ECR とは
- Amazon ECR / Amazon ECR Public の使い方
- 主要な機能

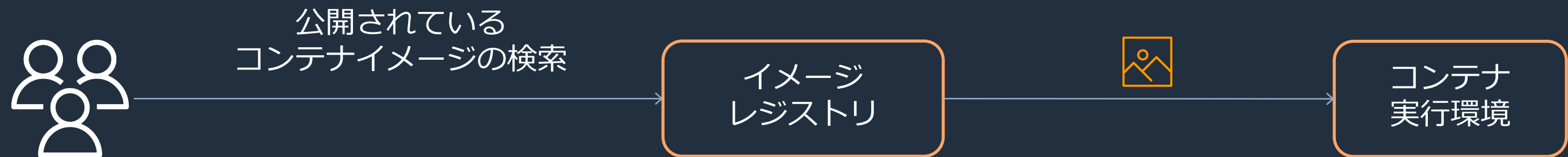
Amazon Elastic Container Registry (Amazon ECR)の概要

コンテナワークロードの基本的なワークフロー

プライベートレジストリ



パブリックレジストリ

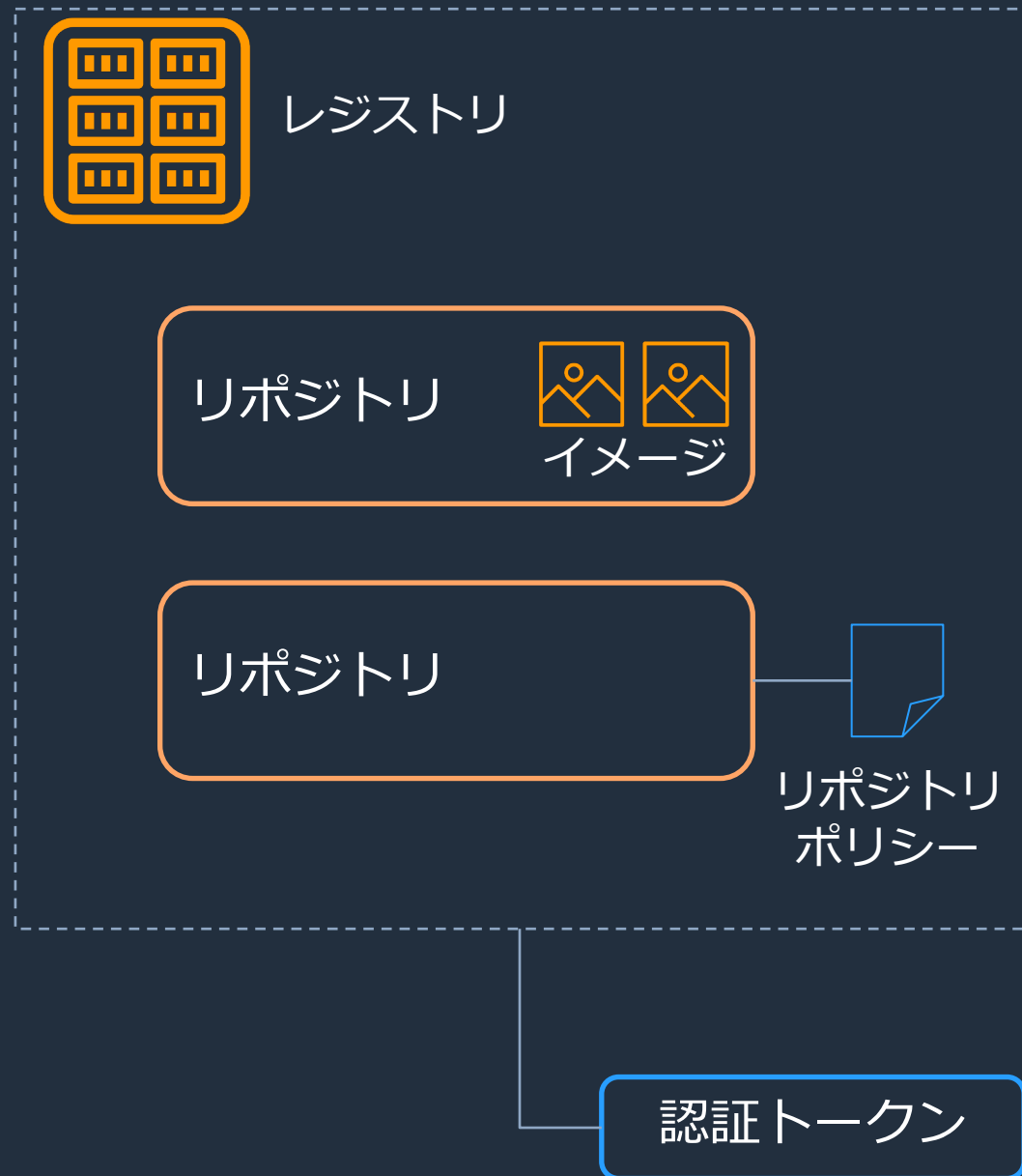


Amazon ECRとは



- コンテナイメージレジストリのマネージドサービス
- Docker CLI コマンドを利用することが可能
- コンテナイメージ/アーティファクトはAmazon S3に保存される
- サポートする形式
 - Docker イメージ(v1,v2)
 - Open Container Initiative (OCI) イメージ
 - OCI 互換 artifacts

基本コンセプト



- レジストリ
 - AWS アカウントごとに用意される
 - イメージを保存するためのリポジトリを複数作成可能
 - Docker クライアントを認証して利用
- リポジトリ
 - Docker イメージ、OCI イメージ、OCI 互換アーティファクトを保存
- リポジトリポリシー
 - リポジトリとリポジトリ内のイメージへのアクセス権を制御
- 認証トークン
 - IAM 認証情報を利用して、レジストリに対して認証をする

サポートする2つのアクセスタイプ

- プライベートレジストリ
 - 利用する(pull・push等)には IAM ベース認証が必須
- パブリックレジストリ
 - Amazon ECR Public
 - 認証無しでもイメージのpullが可能
 - 公開されているイメージを検索可能
- Amazon ECR / Amazon ECR Publicそれぞれでサポートされている機能が異なる

ECR > リポジトリ > リポジトリを作成

リポジトリを作成

一般設定

可視性設定 情報
リポジトリの可視性設定を選択します。

プライベート
アクセスは IAM およびリポジトリポリシーのアクセス許可によって管理されます。

パブリック
イメージについて、パブリックに表示およびアクセス可能。

リポジトリ名
簡潔な名前を指定します。デベロッパーが名前でもリポジトリの内容を識別できる必要があります。

925496135215.dkr.ecr.ap-northeast-1.amazonaws.com/

最大文字数 256 のうち0 (最小文字数 2)。 The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, and forward slashes.

タグのイミュタビリティ 情報
タグのイミュタビリティを有効にすると、同じタグを使用した後続イメージのプッシュによるイメージタグが上書きを防ぎます。タグのイミュタビリティを無効にするとイメージタグを上書きできるようになります。

無効

ℹ リポジトリが作成されると、リポジトリの可視性設定を変更することはできません。

Amazon ECR Amazon ECR Public の使い方

プライベートレジストリの実用例



開発者

1. 認証トークンの取得とDocker クライアントを認証

```
$ aws ecr get-login-password | ¥  
docker login --username AWS --password-stdin ¥  
123456789012.dkr.ecr.<region>.amazonaws.com
```

2. イメージのビルド

```
$ docker build . . . .
```

3. イメージにタグ付け

```
$ docker tag <image> 123456789012.dkr.ecr.<region>.amazonaws.com/<image>:<tag>
```

4. イメージのpush

```
$ docker push 123456789012.dkr.ecr.<region>.amazonaws.com/<image>:<tag>
```



レジストリ

リポジトリ



認証されていない利用者

```
$ docker pull 123456789012.dkr.ecr.<region>.amazonaws.com/<image>:<tag>
```



パブリックレジストリの利用例



開発者

1. 認証トークンの取得とDocker クライアントを認証

```
$ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws
```

2. イメージのビルド

```
$ docker build . . . .
```

3. イメージにタグ付け

```
$ docker tag <image> public.ecr.aws/<alias>/<image>:<tag>
```

4. イメージのpush

```
$ docker push public.ecr.aws/<alias>/<image>:<tag>
```



レジストリ

リポジトリ



認証されていない利用者

```
$ docker pull public.ecr.aws/<alias>/<image>:<tag>
```

Amazon ECR Docker Credential Helper

- Docker Deamon向けのcredential helper
- Dockerクライアントからより簡単にECRを利用することが可能
- プライベートレジストリ/パブリックレジストリ共に利用可能

```
$ cat ./docker/config.json
{
  "credHelpers": {
    "123456789012.dkr.ecr.ap-northeast-1.amazonaws.com": "ecr-login",
    "public.ecr.aws": "ecr-login"
  }
}
< snip >
$ docker pull 123456789012.dkr.ecr.<region>.amazonaws.com/<image>:<tags>
```

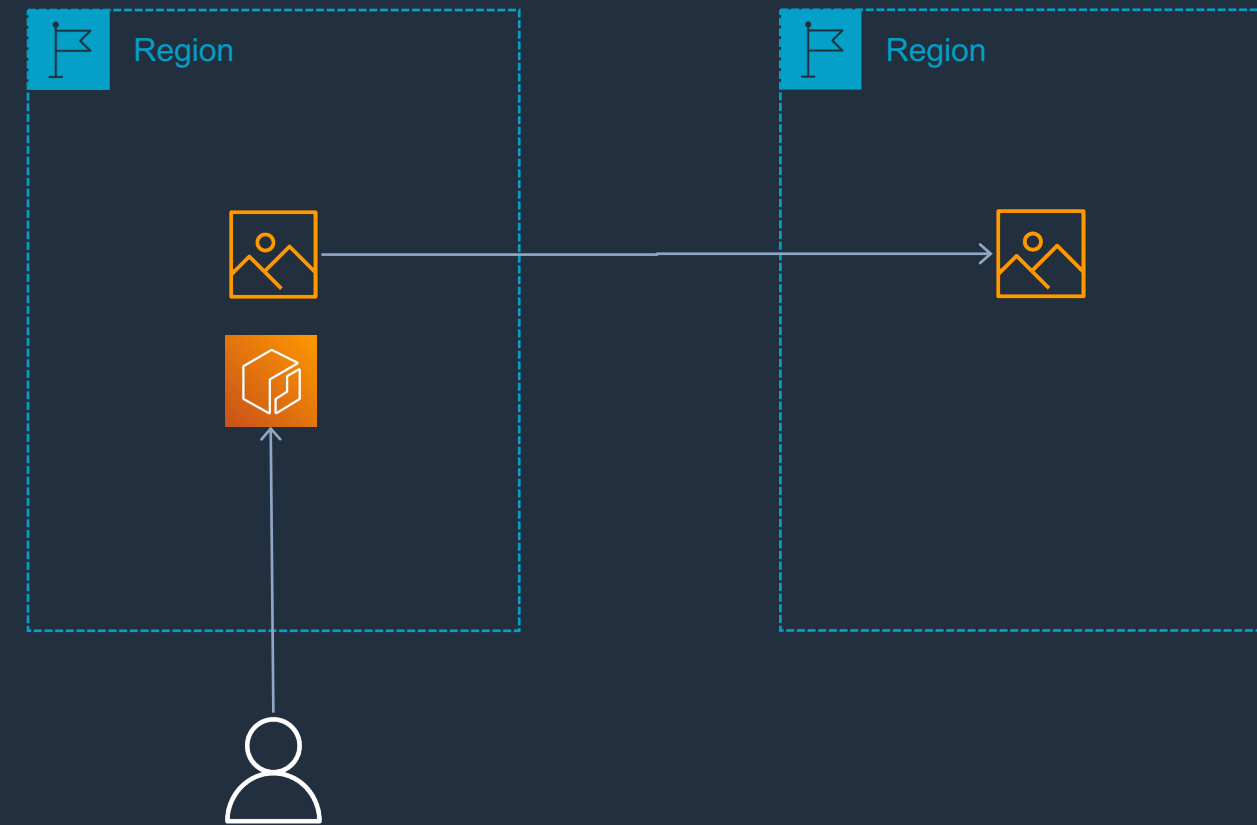
<https://github.com/awslabs/amazon-ecr-credential-helper>

主要な機能

イメージのクロスリージョンレプリケーション

プライベートレジストリ

- コンテナイメージを別リージョン・別AWSアカウントにレプリケーションすることが可能
- イメージをpushした際に自動でレプリケーションされる
- 脆弱性の検出結果はレプリケーションされない



<https://docs.aws.amazon.com/AmazonECR/latest/userguide/replication.html>

<https://aws.amazon.com/jp/blogs/news/cross-region-replication-in-amazon-ecr-has-landed-jp/>

イメージの脆弱性スキャン

プライベートレジストリ

- ECR に pushされたコンテナイメージの脆弱性の検出をする
- Basic scanning
 - 検出対象：OSパッケージ
 - スキャン実行：on push / マニュアル実行
 - 追加料金なしで利用可能
- Enhanced scanning
 - 検出対象：OSパッケージ、プログラミングライブラリ
 - スキャン実行：on push / 継続的
 - Amazon Inspectorとの統合

Scan type
Select the scanning type that will be used for this registry. [Enhanced scanning has additional pricing](#)

Basic scanning
Basic scanning allows manual scans and scan on push of images in this registry. This is a free service.

Enhanced scanning
Enhanced scanning with Amazon Inspector provides automated continuous scanning. Inspector identifies vulnerabilities in both operating system and programming language (such as Python, Java, Ruby etc.) packages in real time.

Continuous scanning filters
Select which repositories will continuously have images scanned for vulnerabilities. Filters are wildcard (*) match on repository name.

Continuously scan all repositories

Add filter

Scan on push filters
Select which repositories will have images scanned for vulnerabilities on image push. Filters are wildcard (*) match on repository name.

Scan on push all repositories

Add filter

Pull Through Cache

プライベートレジストリ

- パブリックレジストリのイメージを ECR Private でキャッシュ
- リモートレジストリと自動同期
 - 24時間以内に同期される
- アップストリームのパブリックレジストリとして ECR Public と Quay.io をサポート

Source Info
The public registry to use as the source for a pull through cache rule.

Public registry
Choose one of the preconfigured public registries to use as the source for the pull through cache rule.

ECR Public
ECR Public
Quay
public.ecr.aws

Destination Info
The Amazon ECR repository namespace to use as the destination for a pull through cache rule.

Amazon ECR repository namespace
Specify the repository namespace to use when caching images from the source registry.

ecr-public-1

https://docs.aws.amazon.com/ja_jp/AmazonECR/latest/userguide/pull-through-cache.html

Amazon ECR Public Gallery

パブリックレジストリ

- Amazon ECR public で公開されているコンテナイメージを検索
- Docker 公式イメージも利用可能
- AWS App Runner サービスを直接起動することも可能

The screenshot displays the Amazon ECR Public Gallery interface. At the top, there is a search bar with the text 'node' entered. Below the search bar, the title 'Amazon ECR Public Gallery' is shown, followed by the subtitle 'Share and deploy container images, publicly and privately'. The main content area is divided into two columns. The left column contains a 'Filters' section with options for 'Verification' (Verified account checked), 'Operating Systems' (Linux checked, Windows unchecked), and 'Architectures' (ARM, ARM 64, x86, x86-64). The right column shows a list of repositories under the heading 'Repositories' (Showing 1 - 10 results of 10). The first repository is 'projectcalico/node' by Amazon EKS Anywhere, with 23 Downloads and architecture tags for Linux and x86-64. The second repository is 'library/node' by Docker, with 118K+ Downloads and architecture tags for Linux, x86-64, x86, ARM, and ARM 64. Below this, a detailed view of the 'hello-app-runner' repository is shown, featuring a profile picture of a person wearing a yellow beanie and sunglasses. The repository is by the AWS Container Services DA Team, has 588K+ Downloads, and is described as an 'Example container for AWS App Runner'. It includes a 'Launch with App Runner' button and a 'Report an Issue' link. The repository was updated 5 months ago. At the bottom of the repository view, there are tabs for 'About', 'Usage', and 'Image tags'.

<https://docs.aws.amazon.com/AmazonECR/latest/public/public-gallery.html>

<https://aws.amazon.com/jp/blogs/news/docker-official-images-now-available-on-amazon-elastic-container-registry-public/>