

AWS CI/CD for Amazon ECS

ハンズオン

～ Cloud9, Docker, Code Services を用いた開発効率向上～

Amazon Web Services Japan K.K.

2020/04/28

本ハンズオンで学ぶこと

ハンズオン①

AWS 上での Docker 管理環境の構築方法

- Cloud9 を使ったクラウド開発環境
- ECS/ECR を使ったコンテナ環境／コンテナイメージ管理
- AWS Fargate を使ったクラスターレスコンテナ運用

ハンズオン②

AWS Code Servicesを利用したCI/CD環境構築方法

- CodeCommitを使ったソースコード管理
- CodeBuildを使ったDockerイメージの自動ビルド
- CodeDeployを使ったECSへのBlue/Green Deployment
- CodePipelineを使ったCI/CDパイプラインの構築



事前準備

サービス概要／用語の理解

- Amazon Elastic Container Service

<https://www.slideshare.net/AmazonWebServicesJapan/20190731-black-belt-online-seminar-amazon-ecs-deep-dive-162160987>

- AWS CodeCommit/CodeBuild/CodePipeline

<https://www.slideshare.net/AmazonWebServicesJapan/aws-black-belt-online-seminar-2017-aws-code-services-codecommit-codebuild>

- AWS Cloud9

<https://www.slideshare.net/AmazonWebServicesJapan/20180613-aws-black-belt-online-seminar-aws-cloud9>



ハンズオンコンテンツ

サンプルソースは以下から取得してください。

- サンプルソース：https://pages.awscloud.com/rs/112-TZM-766/images/cicd_handson_sample.zip

zipファイルはダウンロード後に解凍してください。



以降のハンズオンはすべて
N.Virginiaリージョンで実施します



Amazon ECS



Amazon Elastic Container Service

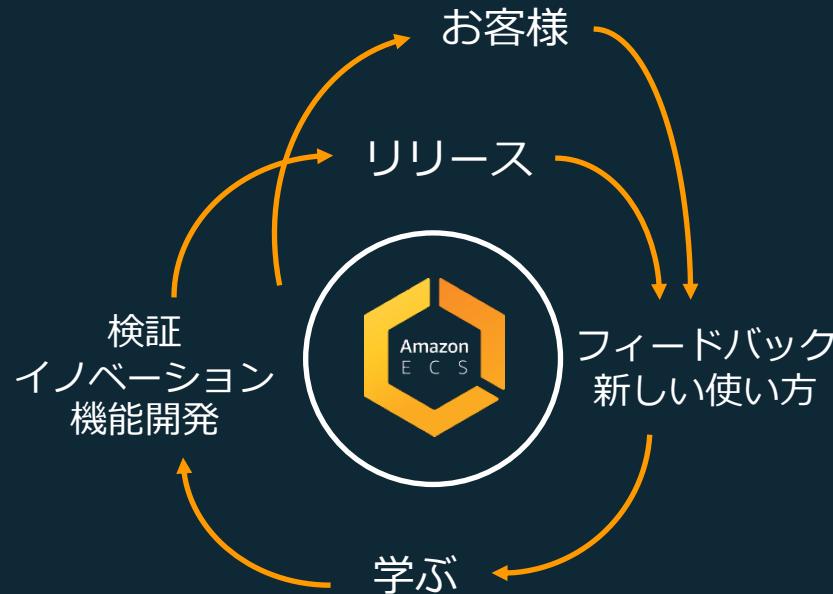


450+%
年間アクティブユーザの成長
(2016年と比較)



数億コンテナ
が毎週起動
数百万もの
インスタンス上で

お客様がAmazon ECS成長の鍵



50+

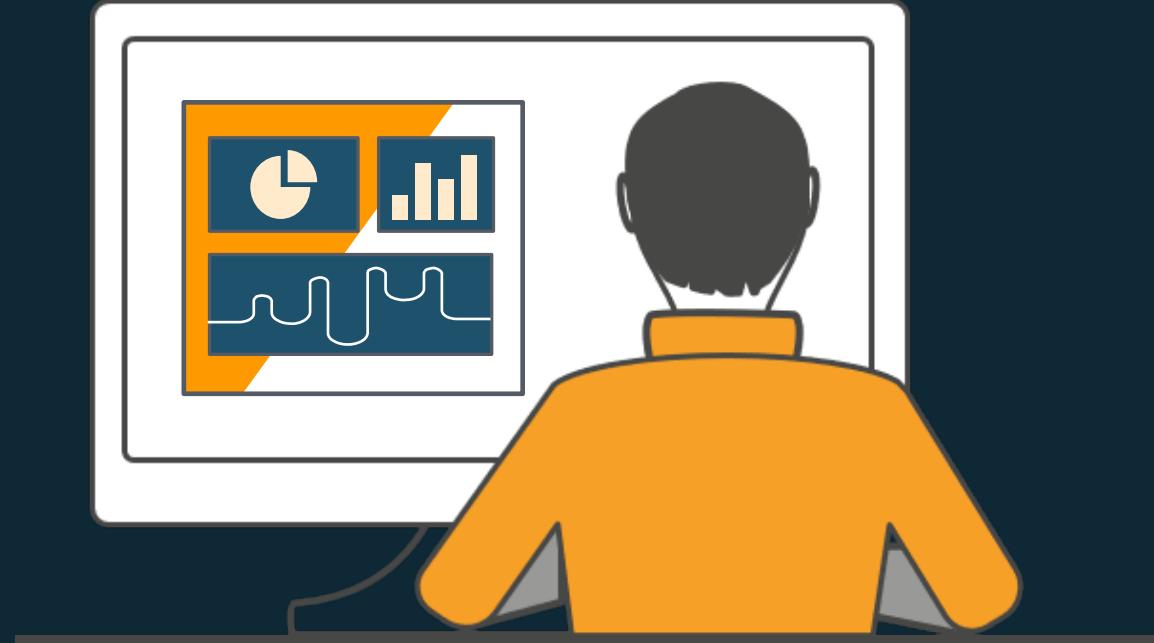
2015年のGA以来
リリースした機能の数

AWS上の本番環境のコンテナを支える



-  AWS VPCネットワークモード
-  タスク配置
-  他のAWSサービスとの深い連携
-  ECS CLI
-  グローバル展開
-  強力なスケジューラ
-  オートスケーリング
-  CloudWatch メトリクス
-  ロードバランサ

アプリケーションの開発に集中したい



計算リソースの使い方を根本的に変える



AWS Fargate



インスタンス
管理不要



タスク
ネイティブAPI



リソース
ベースの価格

=



簡素で、使いやすく、
強力な
新しいリソース消費モデル

タスクに割り当てるCPUとメモリの設定



柔軟な設定の選択肢

- **50** のCPU/メモリ設定から

CPU	Memory
256 (.25 vCPU)	512MB, 1GB, 2GB
512 (.5 vCPU)	1GB to 4GB (1GB 刻み)
1024 (1 vCPU)	2GB to 8GB (1GB 刻み)
2048 (2 vCPU)	4GB to 16GB (1GB 刻み)
4096 (4 vCPU)	8GB to 30GB (1GB 刻み)

ECSでFargateタスクに指定できる主なもの

タスクの機能として

- コンテナイメージ
- CPU, メモリ
- IAMロール
- VPCサブネット
- セキュリティグループ
- CloudWatch Logs

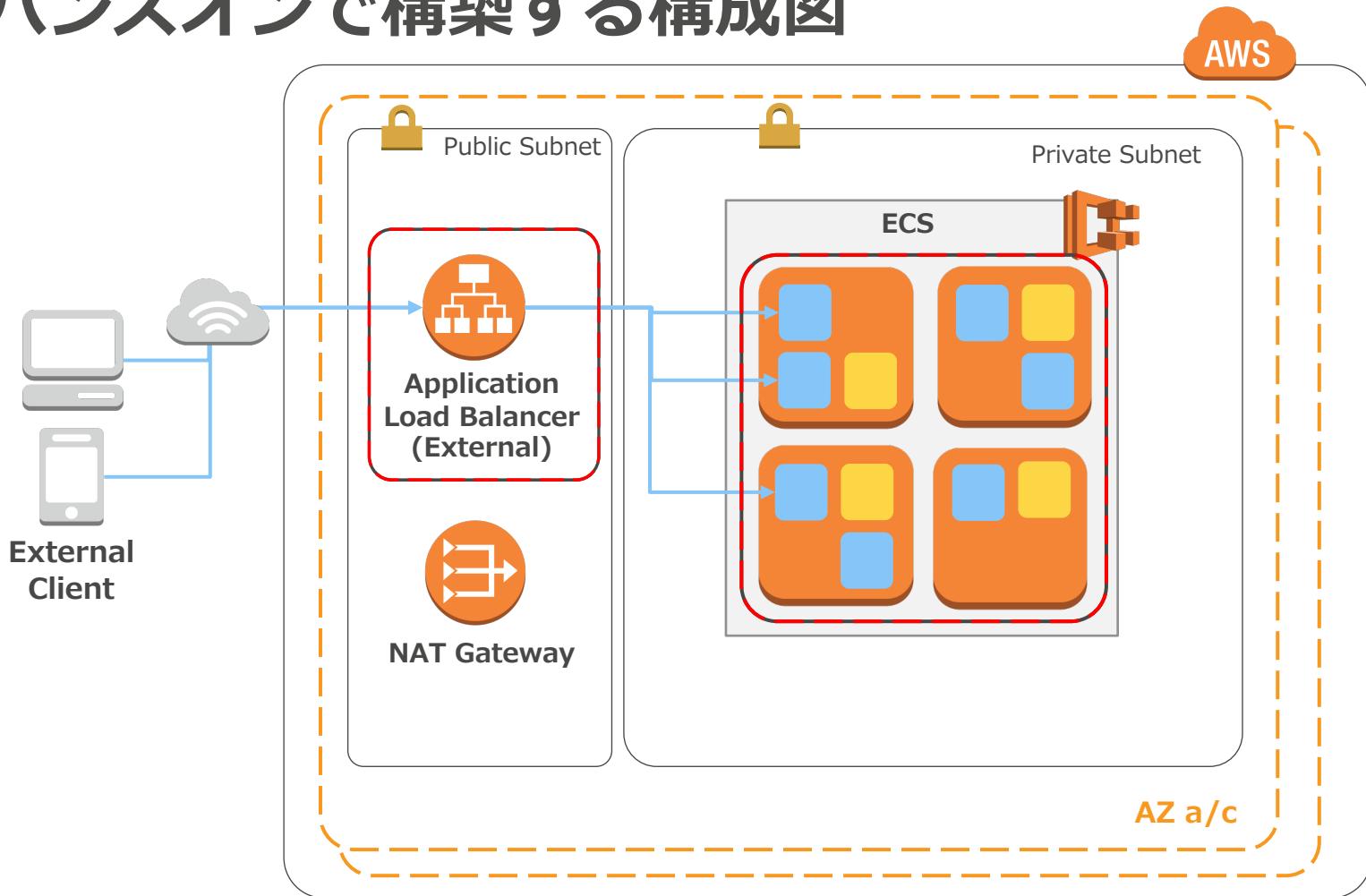
サービスの機能として

- ELB (ALB/NLB)
- オートスケーリング

ハンズオン①

ECS環境の構築

本ハンズオンで構築する構成図



ハンズオン1の手順

1. VPC作成
2. ALB作成
3. ECSクラスター作成
4. Dockerアプリ構築
 - ECRレジストリ作成
 - Cloud9環境の構築
5. ECSタスク・サービス作成
6. ECSサービスAuto Scaling設定

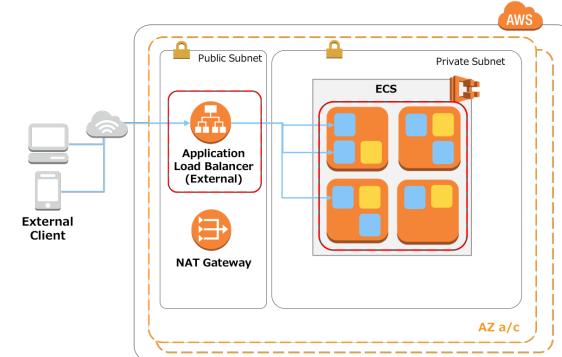
VPC作成

CloudFormationの実行手順の説明 (1)

本日のハンズオンに必要となるネットワークを構築する。

- 主題の ECS/Cloud9/CICD に集中するため、今回は CloudFormation で以下を自動構築する。

- VPC [10.1.0.0/16]
- Internet Gateway & EIP
- NAT Gateway & EIP * 2
- Subnet
 - Public Subnet [10.1.0.0/24, 10.1.2.0/24]
 - Private Subnet [10.1.1.0/24, 10.1.3.0/24]
- Route Table



CloudFormationの実行手順の説明 (2)

1. <https://console.aws.amazon.com/console> を開く
2. N.Virginia Region を選択する
3. Services -> CloudFormation を開く
4. Create Stack を押す
5. Specify templateのUpload template file を選択
6. Choose fileをクリックし、解凍した cloudformation.ymlを選択し、Nextをクリック
7. Stack name を "vpc-for-handson-cicd" と入力
8. 他は特に入力せず Next → Next -> Create Stack

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready Use a sample template Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

Upload a template file

Choose file *cloudformation.yml*
JSON or YAML formatted file

S3 URL: <https://s3-external-1.amazonaws.com/cf-templates-wftlfr40py6m-us-east-1/2020104Flq-cloudformation.yml>

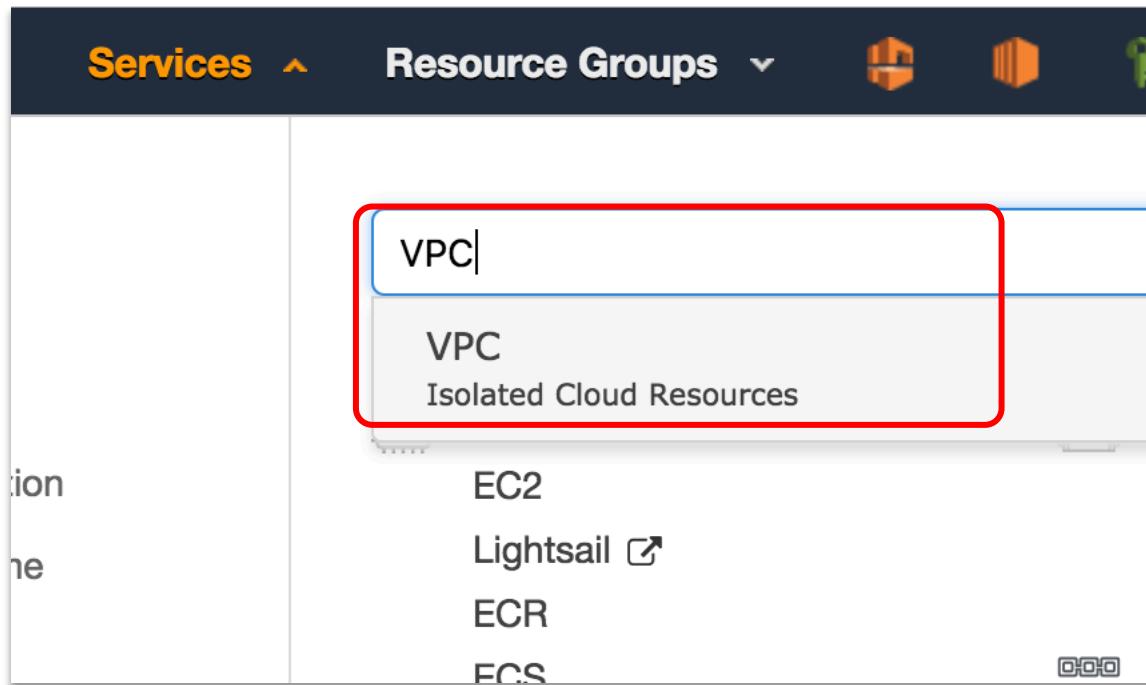
View in Designer

Cancel **Next**



ALB作成

Services -> VPC ^



ALB用セキュリティグループ作成(1)

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Security' section, the 'Security Groups' item is highlighted with a red box. A red arrow points from this box up to the 'Create security group' button at the top of the main content area. The main content area shows the 'Create security group' wizard. It includes fields for 'Security group name*' (set to 'ALB security group'), 'Description*' (set to 'Internet facing ALB'), and 'VPC' (a dropdown menu showing 'vpc-01beaab861ce7f014'). Below the dropdown is a modal dialog titled 'Filter by attributes' with columns for 'VPC ID', 'Name tag', and 'Owner'. It lists two VPCs: 'vpc-01beaab861ce7f014' (selected and highlighted in orange) and 'vpc-0500a57f'. A red box highlights the 'Owner' column for the selected VPC. A blue 'Create' button is visible at the bottom right of the modal. At the bottom of the page, a callout box with an orange background contains the text: 'CloudFormationで作成されたVPC(handson-vpc)を選択 Cloud9の作成時に選択するのでvpc idを保存しておく'.

VPC Dashboard

Filter by VPC:

Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Create security group

Filter by tags and attributes

Name

Security Groups > Create security group

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group fill in the fields below.

Security group name* ALB security group

Description* Internet facing ALB

VPC vpc-01beaab861ce7f014

* Required

Filter by attributes

VPC ID	Name tag	Owner
vpc-01beaab861ce7f014	handson-vpc	handson-vpc
vpc-0500a57f	default	default

CloudFormationで作成されたVPC(handson-vpc)を選択
Cloud9の作成時に選択するのでvpc idを保存しておく

aws

ALB用セキュリティグループ作成(2)

VPC Dashboard

Filter by VPC:

Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Create security group

Actions ▾

Filter by tags and attributes or search by keyword

Name	Group ID	Group Name	VPC ID	Type	Description
sg-0258b65e4901...	default	vpc-01beaab861c...	EC2-VPC	default VPC secur..	
sg-08c9327814375ec74	ALB security group	vpc-01beaab861c...	EC2-VPC	Internet facing ALB	
sg-ef8c4fa1	default	vpc-0500a57f	EC2-VPC	default VPC secur..	

作成したALB用SGを選択

HTTPAnywhere(0.0.0.0/0)に開放し、Save Rules

Security Group: sg-08c9327814375ec74

Inbound Rules

Type: HTTP | Protocol: TCP | Port Range: 80 | Source: Anywhere | Description: e.g. SSH for Admin D

Add Rule

Note: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

* Required

Cancel Save rules

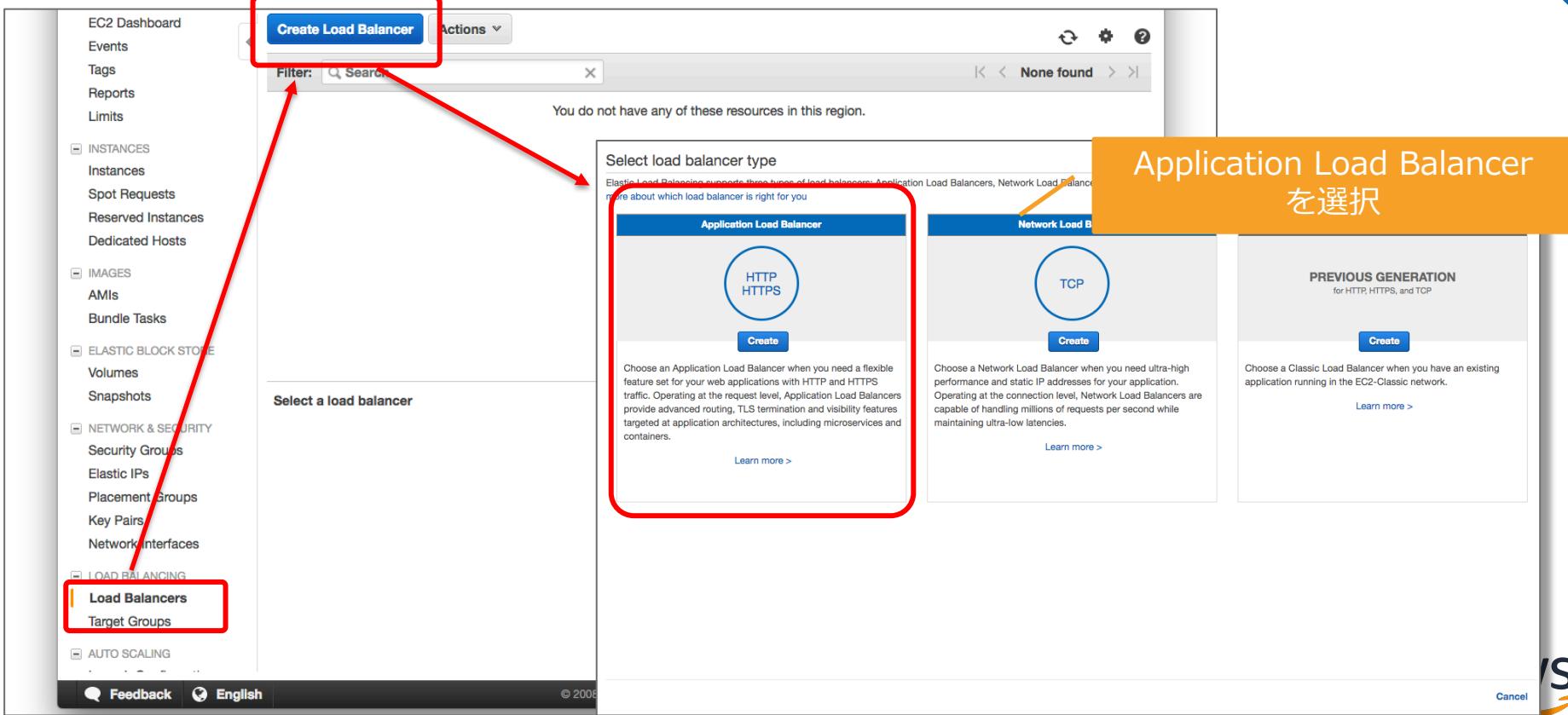
This security group has no rules



ALB(Application Load Balancer)作成

コンソール
EC2

Services -> EC2 ^



基本設定

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name: (php-sample)

Scheme: Internet-facing internal

IP address type:

Nameは php-sample

Internet-facingを選択

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener

Listenerはデフォルト

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC:

Availability Zones:

- us-east-1a: subnet-079864f4068582ea9 (Public subnet)
IPv4 address: Assigned by AWS
- us-east-1b: subnet-0f355931bb1c082a8 (Public subnet 2)
IPv4 address: Assigned by AWS

handsonで作成したvpcを選択

★よくある間違いポイント

サブネットにはus-east-1aのPublic subnetとus-east-1bのPublic subnet 2を選択
Cloud9作成のためにsubnet idを記録しておく

Tags

Cancel

Next: Configure Security Settings

セキュリティ設定(Nextをクリック)

The screenshot shows a step-by-step configuration wizard for a load balancer. The current step is "Step 2: Configure Security Settings". A warning message is displayed: "⚠ Improve your load balancer's security. Your load balancer is not using any secure listener. If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings." At the bottom, there are buttons for "Cancel", "Previous", and "Next: Configure Security Groups", with the "Next" button being highlighted by a red box.

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 2: Configure Security Settings

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings.

Cancel Previous Next: Configure Security Groups



セキュリティグループ設定

[1. Configure Load Balancer](#)[2. Configure Security Settings](#)[3. Configure Security Groups](#)[4. Configure Routing](#)[5. Register Targets](#)[6. Review](#)

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group: Create a new security group

Select an existing security group

Filter [VPC security groups](#)

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-08854e7c10ef43eae	ALB security group	Internet facing ALB	Copy to new
<input type="checkbox"/> sg-0f76a8c692156303c	default	default VPC security group	Copy to new



よくある間違いポイント

先程作成したALB用のSGと
defaultのSGを 両方チェック

[Cancel](#)[Previous](#)[Next: Configure Routing](#)

ルーティング設定(スキップ)

[1. Configure Load Balancer](#)[2. Configure Security Settings](#)[3. Configure Security Groups](#)[4. Configure Routing](#)[5. Register Targets](#)[6. Review](#)

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

Target group

Target group [i](#)

Name [i](#)

Target type
 Instance
 IP
 Lambda function

Protocol [i](#)

Port [i](#)

適当な名前を指定
(このターゲットグループは
あとの手順で削除します)

Health checks

Protocol [i](#)

Path [i](#)

▶ Advanced health check settings

[Cancel](#)[Previous](#)[Next: Register Targets](#)

ターゲットトレジスター(スキップ)

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone	
No instances available.							

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

<input checked="" type="checkbox"/>	Add to registered	on port 80					
<input type="text"/> Search Instances X							
<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR

Cancel Previous Next: Review



ALB作成

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 6: Review
Please review the load balancer details before continuing

Load balancer

- Name: php-sample
- Scheme: internet-facing
- Listeners: Port:80 - Protocol:HTTP
- IP address type: ipv4
 - VPC: vpc-06f24252b9c8d0967 (handson-vpc)
 - Subnets: subnet-024227273adc51ebc (Public subnet), subnet-008ea04a2afd7075e (Public subnet 2)
 - Tags

Security groups

- Security groups: sg-076a8c692156303c, sg-08854e7c10ef43eae

Routing

- Target group: New target group
- Target group name: dummy
 - Port: 80
 - Target type: instance
 - Protocol: HTTP
- Health check protocol: HTTP
 - Path: /
 - Health check port: traffic port
 - Healthy threshold: 5
 - Unhealthy threshold: 2
 - Timeout: 5
 - Interval: 30

Create Load Balancer Actions

Name	DNS name	State	VPC ID	Availability Zones	Type
php-sample	php-sample-1903026157.us-east-1.elb.amazonaws.com	provisioning	vpc-06f24252b9c8d0967	us-east-1b, us-east-1a	application

Load balancer: php-sample

Description Listeners Monitoring Tags

Basic Configuration

Name: php-sample	Creation time: September 27, 2018 at 2:24:42 PM UTC+9
ARN: arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/php-sample/a78ab0b9be355cf0	Hosted zone: Z355XD0TRQ7X7K
DNS name: php-sample-1903026157.us-east-1.elb.amazonaws.com	State: provisioning
name: (A Record)	VPC: vpc-06f24252b9c8d0967
Scheme: internet-facing	IP address type: ipv4
Type: application	AWS WAF Web ACL:
Availability Zones: subnet-008ea04a2afd7075e - us-east-1b, subnet-024227273adc51ebc - us-east-1a	Edit availability zones

Edit

Cancel Previous Create

アプリケーションのエンドポイントとなるDNS名をメモしておく



ALBリスナー削除

リスナーおよびターゲットはECSサービス構成時に再度作成しますので、ALB作成時に自動構成されたものは削除します

Create Load Balancer Actions

search : arn:aws:elasticloadbalancing:us-east-1:... Add filter

Name	DNS name	State	VPC ID
php-sample	php-sample-1903026157.us...	provisioning	vpc-06f24252b9c8d0967

Load balancer: php-sample

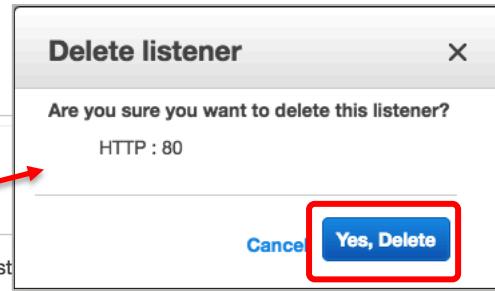
Description **Listeners** Monitoring Tags

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listed listeners and listener rules.

Add listener Edit Delete

Listener ID	Security policy	SSL Certificate	Rules
HTTP : 80	N/A	N/A	Default: forwarding to dummy View/edit rules

HTTP : 80
arn...37885240e0ed436f



ターゲットグループ削除

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Launch Templates
 - Spot Requests
 - Reserved Instances
 - Dedicated Hosts
 - Scheduled Instances
- IMAGES
- ELASTIC BLOCK STORE
- NETWORK & SECURITY
- LOAD BALANCING
 - Load Balancers
 - Target Groups
- AUTO SCALING
 - Launch Configurations
 - Auto Scaling Groups
- SYSTEMS MANAGER SERVICES
- Run Command

The screenshot shows the AWS EC2 Target Groups page. A target group named "dummy" is selected. A context menu is open over the target group name, with the "Delete" option highlighted. A red arrow points from the "dummy" target group name to the "Delete" option in the menu. Another red arrow points from the "Delete" button in the confirmation dialog to the "Yes" button.

Create target group

Actions

- Edit health check
- Register and deregister targets
- Edit attributes
- Delete**

Target group: dummy

Description Targets Health checks Monitoring Tags

Basic Configuration

Name	dummy
ARN	arn:aws:elasticloadbalancing:us-east-1:
Protocol	HTTP
Port	80
Target type	instance
VPC	vpc-06f24252b9c8d0967
Load balancer	

Delete target group

Are you sure you want to delete this target group?
dummy

Cancel Yes

リスナーおよびターゲットはECSサービス構成時に再度作成しますので、ALB作成時に自動構成されたものは削除します



Dockerアプリ開発



ECRリポジトリ作成

Services -> Elastic Container Registry ^

ECR
コンソール

Compute

Amazon Elastic Container Registry

Easily store, manage, and deploy container images

Amazon Elastic Container Registry (ECR) is a fully-managed container registry that makes it easy for developers to store, manage, and deploy container images.

Create a repository

Get Started



ECRリポジトリ作成

ECR > Repositories > Create repository

Create repository

Repository configuration

Repository name

.dkr.ecr.us-east-1.amazonaws.com/

任意のリポジトリ名を指定

A namespace can be included with your repository name (e.g. namespace/repo-name).

Tag immutability

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

Disabled

Scan on push

Enable scan on push to have each image automatically scanned after being pushed to a repository. If disabled, each image scan must be manually started to get scan results.

Disabled

Cancel

Create repository



ECRリポジトリ作成

Successfully created repository

View push commands

Push commands for concatenate-parts

macOS / Linux Windows

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry.
Use the AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-REDACTED
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t concatenate-parts .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag concatenate-parts:latest dkr.ecr.us-east-1.amazonaws.com(concatenate-parts:latest)
```

Close NS

AWS Cloud9環境の構築



AWS Cloud9環境の構築

Cloud9



AWS サービス

cloud9

Cloud9
コードの記述、実行、デバッグのためのクラウド IDE

コンピューティング

- EC2
- Lightsail
- Elastic Container Service
- EKS
- Lambda
- Batch
- Elastic Beanstalk

ストレージ

- S3
- EFS
- Glacier
- Storage Gateway

データベース

- RDS
- DynamoDB
- ElastiCache
- Neptune
- Amazon Redshift

Developer Tools

AWS Cloud9

a cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

How it works

Create an AWS Cloud9 development environment on a new Amazon EC2 instance or connect it to your own Linux server through SSH. Once you've created an AWS Cloud9 environment, you will have immediate access to a rich code editor, integrated debugger, and built-in terminal with pre-configured AWS CLI – all within your browser.

役に立つヒント

コストの管理

AWS Budgets を使用して、AWS のコスト、使用量、および予約をモニタリングします。今すぐ開始

[Create environment] をクリック

New AWS Cloud9 environment

Create environment

Getting started

Before you start	2 min read
Create a environment	3 min read
Working with environments	15 min read
Working with the IDE	10 min read
Working with AWS Lambda	5 min read



AWS Cloud9環境の構築

AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Name environment

Environment name and description

Name

The name needs to be unique per user. You can update it at any time in your environment settings.

hansdon-yourname

Limit: 60 characters

Description - Optional

This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Write a short description for your environment

[Next step]をクリック

Limit: 200 characters

Cancel

Next step

[Network settings(advanced)]をクリック

AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Configure settings

Environment settings

Environment type

Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

Create a new instance for environment (EC2)

Launch a new instance in this region to run your new environment.

Connect and run in remote server (SSH)

Display instructions to connect remotely over SSH and run your new environment.

Instance type

t2.micro (1 GiB RAM + 1 vCPU)

Free-tier eligible. Ideal for educational users and exploration.

t2.small (2 GiB RAM + 1 vCPU)

Recommended for small-sized web projects.

m4.large (8 GiB RAM + 2 vCPU)

Recommended for production and general-purpose development.

Other instance type

Select an instance type.

t2.nano

Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation setting of half an hour of no activity to maximize savings.

After 30 minutes (default)

IAM role

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

▶ Network settings (advanced)

※ docker build 時などでメモリが不足することがあるので、t3.small 以上が望ましい

Cancel

Previous step

Next step

AWS Cloud9環境の構築

ハンズオンで作成したvpc idを選択
(defaultがある場合、defaultじゃない方)

Network settings (advanced)

Network (VPC)
vpc-t... ▾ Create new VPC

Subnet
Select a range of IP addresses in your VPC to isolate EC2 resources from each other.
subnet-... is-east-1a ▾ Create new subnet

Cancel Previous step Next step

よくある間違いポイント

保存しておいたPublic subnetのいずれかを選択
(Services -> VPC -> Subnetで確認)

[Next step]をクリック

Review

Environment name and settings

Name
hansdon-yourname

Description
No description provided

Environment type
EC2

Instance type
t2.micro

Subnet
subnet-...

Cost-saving
After 30 min

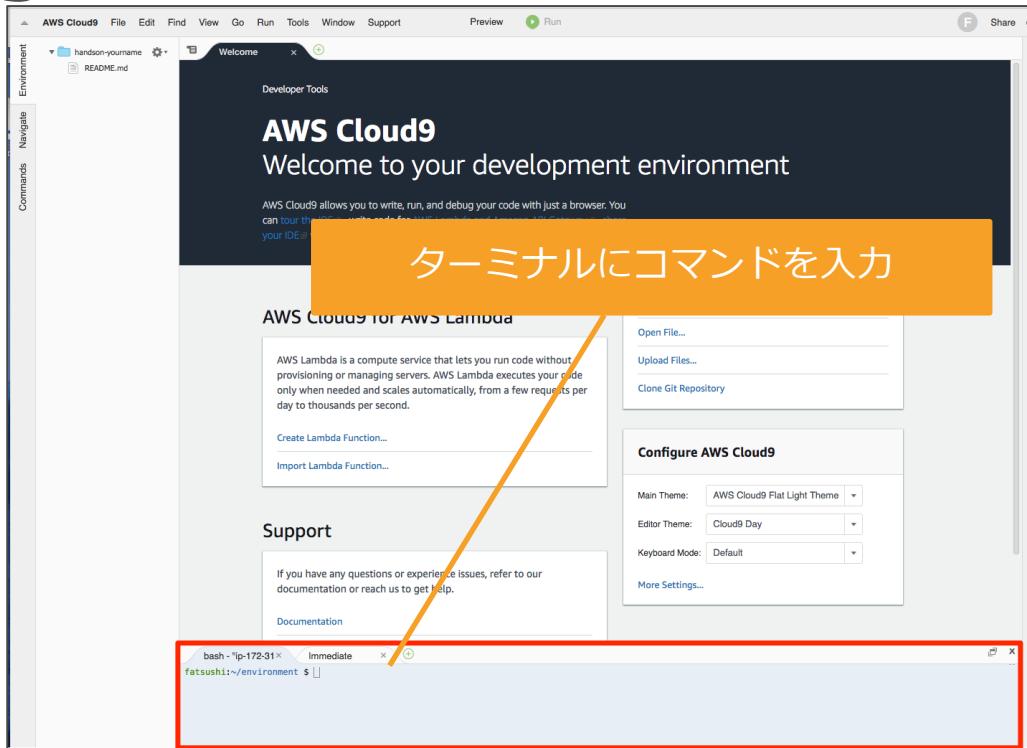
IAM role
AWSServiceRoleForAWSCloud9 (generated)

We recommend the following best practices for using your AWS Cloud9 environment

- Use source control and backup your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular updates of software on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- Turn on AWS CloudTrail in your AWS account to track activity in your environment. Learn more
- Only share your environment with trusted users. Sharing your environment may put your AWS access credentials at risk. Learn more

Cancel Previous step Create environment

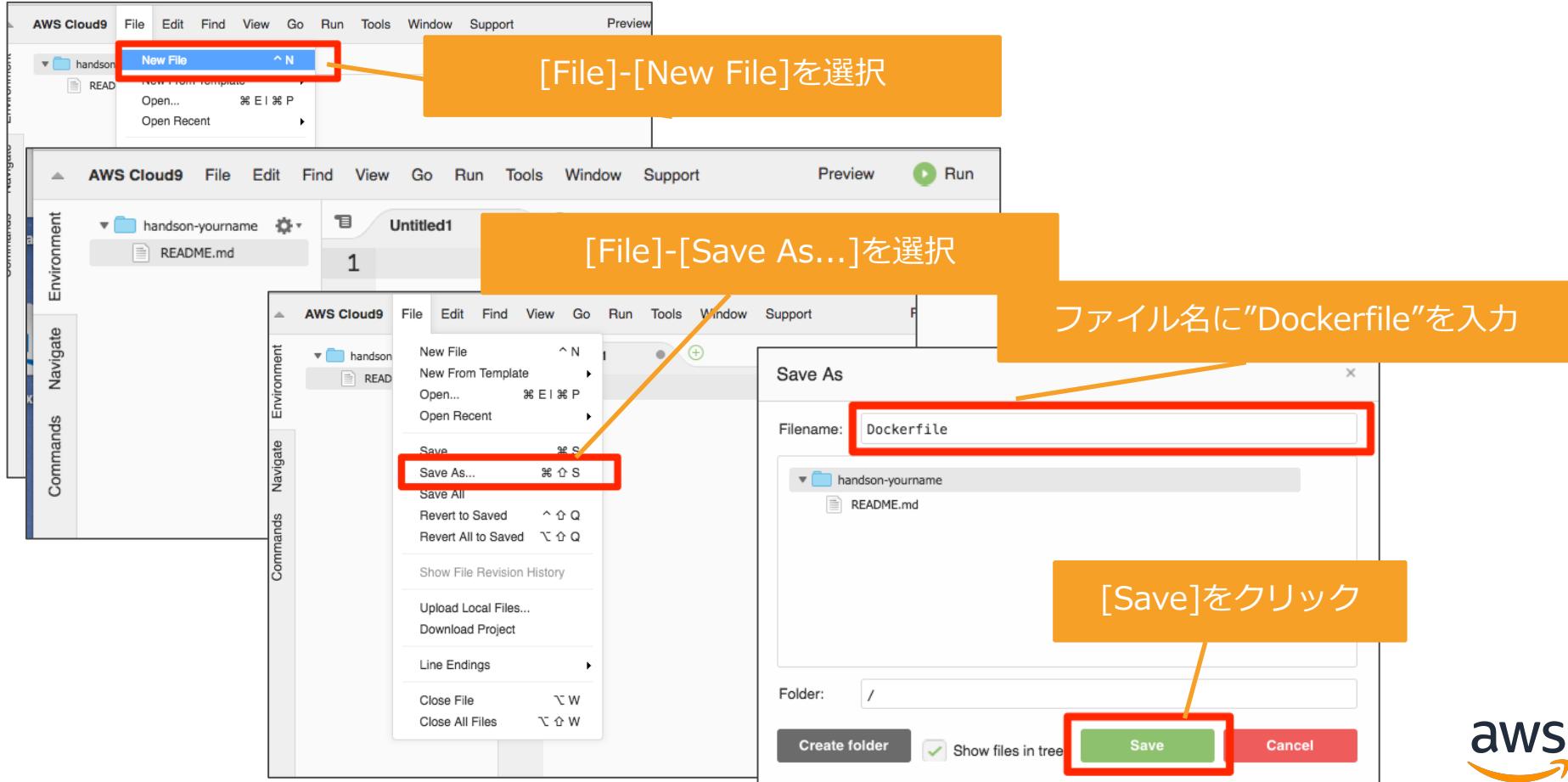
git環境の設定



```
$ git config --global user.name "yourname"
$ git config --global user.email yourname@abc.com
$ git config --global credential.helper '!aws codecommit credential-helper $@'
$ git config --global credential.UseHttpPath true
```



Dockerfileの作成



Dockerfile作成(PHPサンプルアプリ)

1. 前ページで作成したDockerfileに以下の内容を入力して保存

```
FROM php:7.4.0-apache  
COPY src /var/www/html/
```

Dockerfile

Docker Hubの公式イメージから
PHP/apacheのDockerイメージを取得
https://hub.docker.com/_/php/

2. handson-xxxx配下に、srcディレクトリとindex.phpファイルを作成

```
<!DOCTYPE html>  
<html lang="ja">  
  <head>  
    <title>PHP Sample</title>  
  </head>  
  <body>  
    <?php echo gethostname(); ?>  
  </body>  
</html>
```

src/index.php

ローカルのsrcディレクトリ配下を
コンテナ内の/var/www/html
ディレクトリにコピー

ホスト名を表示するPHPアプリ
(余裕のある方は好きなコードを書いてください)

※お急ぎの方はZipファイル内のサンプルコードをご利用ください。



Dockerfile作成(PHPサンプルアプリ) – 例

The screenshot shows the AWS Cloud9 IDE interface. On the left, the file tree displays a project structure with a .c9 environment folder containing a Dockerfile, index.php, and README.md files. The main workspace has two tabs open: 'Dockerfile' and 'index.php'. The 'Dockerfile' tab contains the following code:

```
1 FROM php:7.4.0-apache
2 COPY src/ /var/www/html/
3
```

The 'index.php' tab contains the following PHP sample code:

```
1 <html>
2 ..<head>
3 ....<title>PHP Sample</title>
4 ..</head>
5 ..<body>
6 ....<?php echo gethostname(); ?>
7 ..</body>
8 </html>
```

At the bottom, status bars indicate '3:1 Dockerfile Spaces: 4' and '8:8 PHP Spaces: 4'.



Dockerイメージ作成～コンテナ起動

Cloud9ターミナルで作業

1. Dockerイメージ作成

```
$ docker build -t php-sample . #メモしておいたコマンド
$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
php-sample      latest    bc47e3ede49a    3 minutes ago  390 MB
```

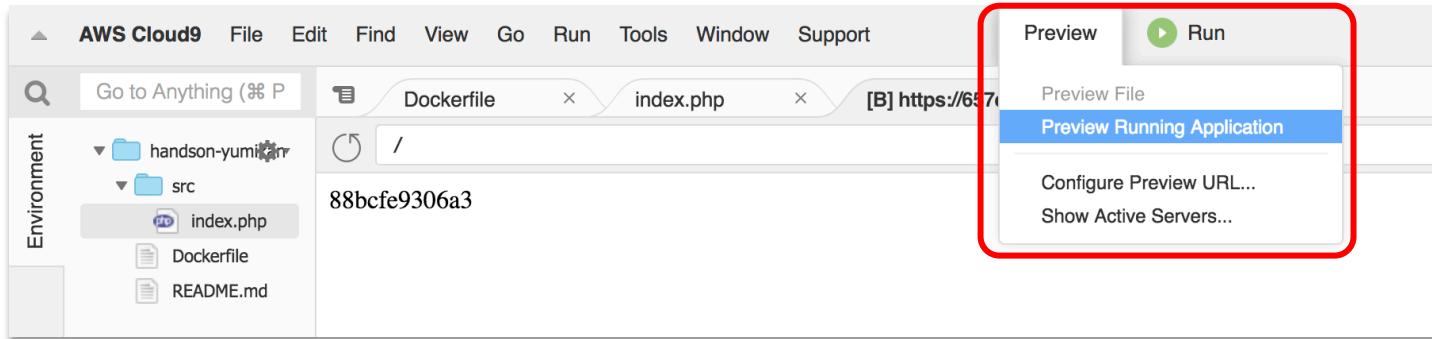
2. Dockerコンテナ起動

```
$ docker run --rm -p 8080:80 -d php-sample:latest #ホストPort8080でコンテナport80にアクセス
$ docker ps
CONTAINER ID  IMAGE          PORTS          NAMES
67776a2787dd  php-sample:latest  ~ ~ ~  0.0.0.0:8080->80/tcp  modest_wozniak
```



Dockerイメージ作成～コンテナ動作確認と停止

- [Preview]メニューの[Preview Running Application]を実行しPHPのページが表示されることを確認



- Dockerコンテナ停止

```
$ docker stop 67776a2787dd # `docker ps`で表示されたコンテナIDを指定
```

ECRへのDockerイメージPush

ECR 作成時にメモしたコマンドを実行する

1. ECRへのログイン

```
$ $(aws ecr get-login --no-include-email)  
...  
Login Succeeded # ログインに成功することを確認する
```

先ほどメモしておいたコマンド

2. タグ付け

```
$ docker tag php-sample:latest XXXXXXXXXX.dkr.ecr.us-east-1.amazonaws.com/php-sample:latest
```

3. ECRへのイメージPush

```
$ docker push XXXXXXXXXX.dkr.ecr.us-east-1.amazonaws.com/php-sample:latest  
...  
latest: digest: sha256:.... size: 3450 # イメージ Push に成功することを確認する
```

先ほどメモしておいたコマンド



ECRコンソールでのイメージ確認

Services -> ECR ^

ECS
コンソール

ECR > Repositories

Repositories (1)

Find Repositories

Repository name	URI	Created at
php-sample	dkr.ecr.us-east-1.amazonaws.com/php-sample	03/17/19, 11:11:22 PM

このあとのタスク定義で利用するためリポジトリURIをメモしておく

Amazon Container Services

Amazon ECS

Clusters

Task definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

Images

Permissions

Lifecycle Policy

Tags

ECR > Repositories > php-sample

php-sample

View push commands

Images (1)

Find Images

Image tag	Image URI	Pushed at	Digest	Size (MB)
latest	dkr.ecr.us-east-1.amazonaws.com/php-sample:latest	03/17/19, 11:42:02 PM	sha256:54495c061...	133.37

View push commands

Latestのタグがついたイメージがあることを確認

The screenshot shows the AWS ECR console interface. In the top section, a repository named 'php-sample' is listed with its URI: 'dkr.ecr.us-east-1.amazonaws.com/php-sample'. A callout box with the text 'このあとのタスク定義で利用するためリポジトリURIをメモしておく' (Memo the repository URI for use in the task definition) points to this URI. In the bottom section, the 'Images' tab for the 'php-sample' repository is shown, displaying one image entry with the tag 'latest' and the same URI. A red box highlights the 'latest' tag entry, and a callout box with the text 'Latestのタグがついたイメージがあることを確認' (Confirm that an image with the latest tag exists) points to it. The left sidebar shows navigation links for Amazon Container Services, Amazon ECS, Amazon EKS, and Amazon ECR, with 'Images' under ECR being the active tab.

aws

ハンズオン②

AWS Fargate環境の構築

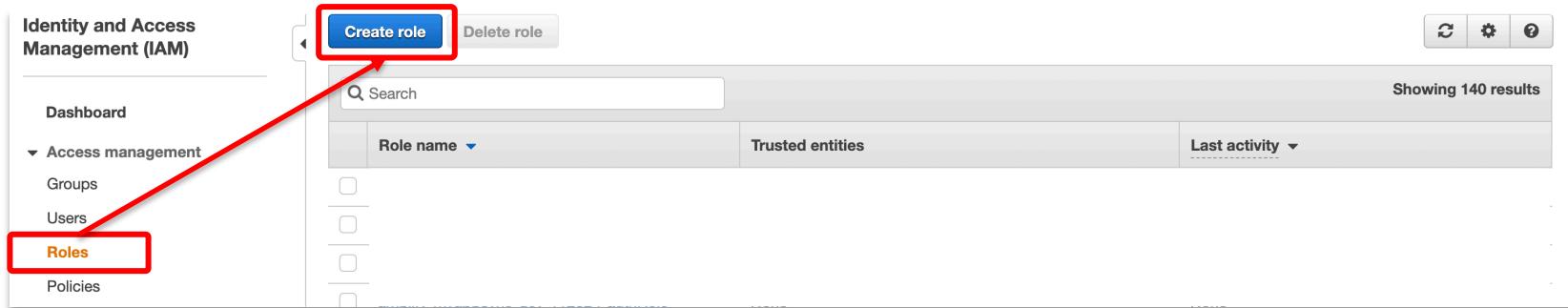


IAMロール設定(1)

コンソール IAM

CodeDeployが利用するIAMロールを作成します。

Services -> IAM へ



IAMロール設定(2)

Create role

Select type of trusted entity

1 2 3 4

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web Identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

API Gateway	Comprehend	ElastiCache	Lambda	SMS
AWS Backup	Config	Elastic Beanstalk	Lex	SNS
AWS Support	Connect	Elastic Container Service	License Manager	SWF
Amplify	DMS	Elastic Transcoder	Machine Learning	SageMaker
AppSync	Data Lifecycle Manager	Elastic Load Balancing	Macie	Security Hub
Application Auto Scaling	Data Pipeline	Forecast	MediaConvert	Service Catalog
Application Discovery Service	DataSync	Glue	OpsWorks	Step Functions
Batch	DeepLens	Greengrass	Personalize	Storage Gateway
CloudFormation	Directory Service	GuardDuty	RAM	Textract
CloudHSM	DynamoDB	Inspector	RDS	Transfer
CloudTrail	EC2	IoT	Redshift	Trusted Advisor
CloudWatch Application Insights	EC2 - Fleet	IoT Things Graph	Rekognition	VPC
CloudWatch Events	EC2 Auto Scaling	KMS	RoboMaker	WorkLink
CodeBuild	EKS	Kinesis	S3	WorkMail
	EMR			

CodeDeploy

Select your use case

CodeDeploy
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

CodeDeploy - ECS
Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

CodeDeploy for Lambda
Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.

* Required

Cancel

Next: Permissions

Roleタイプで
AWS Service >
CodeDeploy >
CodeDeploy – ECS を選択してから
Next: Permissionを選択



IAMロール設定(3)

Create role

Attached permissions policies

The type of role that you selected requires the following policy.

Policy name	Used as	Description
AWSCodeDeployRoleForECS	Permissions policy (1)	Provides CodeDeploy service wide access to ...

* Required

AWSCodeDeployRoleForECS
ポリシーが表示されていることを確認しNext: Tags

Cancel Previous Next: Tags



IAMロール設定(4)

Create role

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text"/>	<input type="text"/>	Remove

[Add new key](#)

You can add 50 more tags.

Cancel Previous **Next: Review**

□ ロール名はCodeDeployRoleforECSにして
Create role

Create role

Review

Provide the required information below and review this role before you create it.

Role name **CodeDeployRoleforECS**

Role description Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

Trusted entities AWS service: codedeploy.amazonaws.com

Policies **AWSCodeDeployRoleForECS**

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

Cancel Previous **Create role**



Fargateクラスター作成(1) Services->ECS

ECS
コンソール

The screenshot shows the AWS ECS console interface. On the left, a sidebar lists various services: Amazon ECR, Amazon EKS, Amazon ECR Repositories, AWS Marketplace, Discover software, and Subscriptions. The 'Clusters' option is selected and highlighted with a red box. A red arrow points from this box to the 'Create Cluster' button, which is also highlighted with a red box. The main area displays the 'Clusters' page for the 'handson-cluster'. It shows 0 Services, 0 Running tasks, and 0 Pending tasks under the EC2 section. Below this, it shows 1 Services, 2 Running tasks, and 0 Pending tasks under the FARGATE section. The CPU Utilization is listed as 0.0. To the right, a modal window titled 'Select cluster template' is open. It contains three options: 'Networking only', 'EC2 Linux + Networking', and 'EC2 Windows + Networking'. The 'Networking only' option is highlighted with a red box and has a red arrow pointing to its 'Powered by AWS Fargate' text. At the bottom of the modal, there is a 'Next step' button highlighted with a red box.

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

Create Cluster

View [list](#) [card](#)

handson-cluster >

FARGATE

	Services	Running tasks	Pending tasks	CPU Util
EC2	0	0	0	0.0
	1	2	0	0.0
	Services	Running tasks	Pending tasks	CPU Util

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only

Resources to be created:

- Cluster
- VPC (optional)
- Subnets (optional)

Powered by AWS Fargate

EC2 Linux + Networking

Resources to be created:

- Cluster
- VPC
- Subnets

Auto Scaling group with Linux AMI

EC2 Windows + Networking

Resources to be created:

- Cluster
- VPC
- Subnets

Auto Scaling group with Windows AMI

*Required

Cancel **Next step**

Fargateクラスター作成(2)

Configure cluster

Cluster name* fargate-cluster i

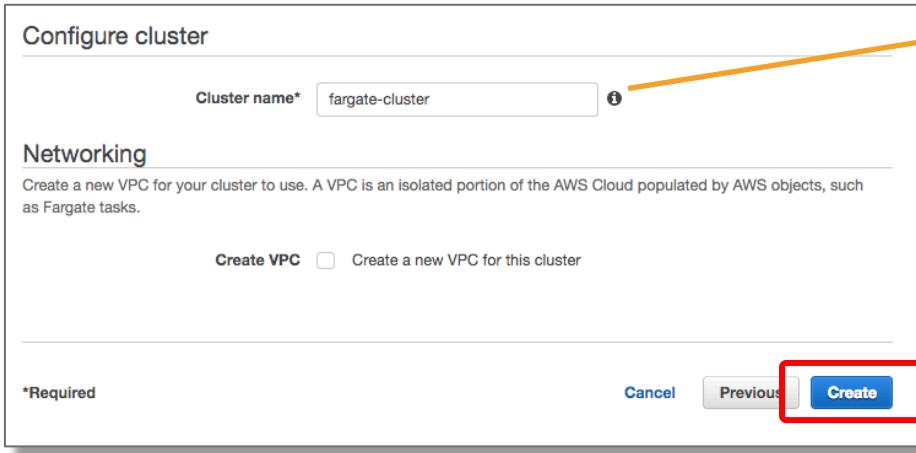
Networking

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

Create VPC Create a new VPC for this cluster

*Required

Cancel Previous Create



クラスター名を指定
(fargate-cluster)



Fargateクラスターの確認

Launch status

Your container instances are launching, and it may take a few minutes until they are in the running state and ready to access. Usage hours on your new container instances start immediately and continue to accrue until you stop or terminate them.

The screenshot shows the AWS CloudWatch Metrics console with a red box highlighting the 'View Cluster' button. A red arrow points from this button to a larger red box containing the cluster details. The cluster is named 'fargate-cluster' and is currently 'ACTIVE'. It has 0 registered container instances, 0 pending tasks, 0 running tasks, 0 active services, and 0 draining services. The table below shows no results for the service list.

ECS status - 1 of 1 complete fargate-cluster

ECS cluster
ECS Cluster fargate-cluster successfully created

Cluster : fargate-cluster

Status ACTIVE

Registered container instances 0

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 0 EC2

Active service count 0 Fargate, 0 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks ECS Instances Metrics Scheduled Tasks

Create Update Delete Last updated on January 24, 2018 11:05:13 AM (0m ago) [Filter in this page](#) [Launch type ALL](#)

<input type="checkbox"/>	Service Name	Status	Task Definiti...	Desired tas...	Running tas...	Launch type	Platform ver...
No results							



Fargate用タスク定義作成(1)

Services -> ECS ^

ECS
コンソール

Amazon ECS

Clusters

Task Definitions

Repositories

Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use. [Learn more](#)

Create new Task Definition Create new revision Actions

Last updated on June 7, 2017 3:42:29 PM (0m ago) [Edit](#) [Help](#)

Status: **ACTIVE** INACTIVE

Filter in this page

Task Definition

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE

Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2

Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

*Required

Cancel **Next step**

Feedback English

タスク定義作成(2)

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name* ⓘ

Requires Compatibilities* FARGATE

Task Role ⓘ

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#) ⓘ

Network Mode ⓘ

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Task execution IAM role

This role is required by Fargate tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the `ecsTaskExecutionRole` already, we can create one for you.

Task execution role ⓘ

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (GB) ⓘ

The valid memory range for 0.25 vCPU is: 0.5GB - 2GB.

Task CPU (vCPU) ⓘ

1GB shared CPU for 0.5 GB memory is: 0.25 vCPU

Container Definitions

Add container

Container Name	Image	Hard/Soft memory...	CPU Units	Essential
----------------	-------	---------------------	-----------	-----------

タスク名を指定
(php-sample-fargate)

Task Role は None

タスクに割り当てるメモリ／CPUを
0.5GB/0.25vCPUに指定



タスク定義作成(3)

Add container

▼ Standard

Container name*

Image*

Custom image format: [registry-url]/[namespace]/[image]:[tag]

Private repository authentication*

Memory Limits (MiB) Soft limit
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation' parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings Container port

コンテナ名を指定(後続の手順の関係上今回はphp-sample-fargateに固定)

利用するDockerイメージを<ECRのリポジトリURI>:<タグ>の形式で指定(今回のタグはlatest)
リポジトリは以前の手順と同じものを指定する

コンテナに割り当てられるMemoryのリミットを定義
今回はSoftLimitで128MBを指定

コンテナportを80に設定



タスク定義作成(3)

ENVIRONMENT

CPU units: 256

GPUs: [empty]

Essential:

Entry point: comma delimited: sh,-c

Command: comma delimited: echo,hello world

Working directory: /usr/app

Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value into containers at run-time.

Key: Add key Value Add value

* Required Cancel Add

各コンテナに対してのCPU予約を定義
今回は1/4Core予約(1024 CPU units = 1Coreなので256を指定)



タスク定義作成(4)

Container Definitions

Add container

Container N...	Image	Hard/Soft m...	CPU Units	GPU	Essential	
php-sam...	33509371799...	--/128	256		true	

Service Integration

AWS App Mesh is a service mesh based on the Envoy proxy that makes it easy to monitor and control microservices. App Mesh standardizes how your microservices communicate, giving you end-to-end visibility and helping to ensure high-availability for your applications. [Learn more](#)

Enable App Mesh integration

Proxy Configuration

The configuration details for the App Mesh proxy. [Learn More](#)

Enable Proxy Configuration

Volumes

Add volume

Configure via JSON

Tags

Key	Value
<input type="text" value="Add key"/>	<input type="text" value="Add value"/>

*Required

Cancel Previous **Create**



Fargateでのサービス作成(1)

Amazon ECS

Clusters

Task Definitions

Repositories

Task Definitions > php-sample-fargate > 1

Task Definition: php-sample-fargate:1

View detailed information for your task definition. To modify the task definition, you need to create a new revision and then make the required changes to the task definition

Create new revision **Actions** ▾

Builder **JSON**

Create Service

Run Task

Update Service

Deregister

Name: **php-sample-fargate**

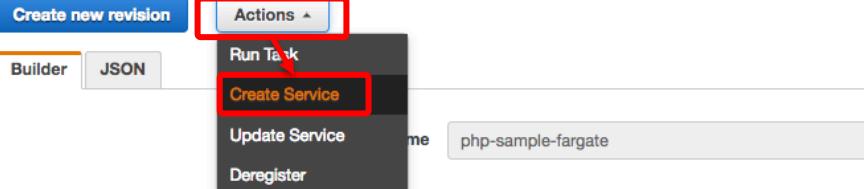
Task Role: **None**

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#)

Network Mode: **awsvpc**

Compatibilities: EC2, FARGATE

Requires compatibilities: FARGATE



Fargateでのサービス作成(2)

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type FARGATE EC2

Task Definition Family
php-sample-fargate

Revision
1

Platform version LATEST

Cluster fatgate-cluster

Service name php-sample-fargate

Service type* REPLICA

Number of tasks 1

Minimum healthy percent 50

Maximum percent 200

*Required

Cancel

Next step

Launch TypeはFARGATEを選択

先ほど作成したタスク定義とクラスターを指定

Service name は php-sample-fargate

初期起動タスク数を指定
今回は1タスクを起動する



Fargateでのサービス作成(3)

Deployments

Choose a deployment option for the service.

Deployment type* Rolling update ⓘ

Blue/green deployment (powered by AWS CodeDeploy) ⓘ

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with [default settings](#) for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

Deployment configuration*

CodeDeployDefault.ECSAllAtOnce

The deployment configuration specifies how traffic is shifted to the updated Amazon ECS task set. [Learn more](#)

Service role for CodeDeploy*

CodeDeployRoleforECS

The IAM role the service uses to make API requests to authorized AWS services. Create a service role for CodeDeploy in the IAM console.
[Learn more](#)

Deployment Typeは
Blue/green deploymentを選択

Service Roleは先ほど作成した
CodeDeployRoleforECSを選択

ⓘ Tagging requires that you opt in to the new ARN and resource ID format.

The IAM user/role has not opted in to the new ARN format. Opt-in to the new format to use this feature. [Manage your opt-in settings](#).

*Required

Cancel

Next step



Fargateでのサービス作成(3)

Configure network

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC*

vpc-06f24252b9c8d0967 (10.1.0.0/16)



Subnets*

subnet-0d41b8b7213b9aa6d
(10.1.1.0/24) | Private subnet - us-east-1a
assign ipv6 on creation: Disabled



subnet-0a93ec50251299df8
(10.1.3.0/24) | Private subnet 2 - us-east-1b
assign ipv6 on creation: Disabled



Security groups*

sg-0f76a8c692156303c

Edit



Auto-assign public IP

DISABLED



作成したVPCを選択
(10.1.0.0/16)

よくある間違いポイント

VPC内のPrivateSubnetを
2つ選択

よくある間違いポイント

DefaultのSecurityGroupを
選択(Editして変更)

Auto-assign public IPを
DISABLEDに変更



Fargateでのサービス作成(4)

Load balancing

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the [Amazon EC2 console](#).

Load balancer type*

Application Load Balancer
Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.

Network Load Balancer
A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.

Service IAM role

Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. [Learn more](#).

Load balancer name

Container to load balance

Container name : port

Container to load balance

php-sample-fargate : 80

Production listener port*

Production listener protocol*

Test listener
An optional test listener is used to test the new application revision before routing traffic to it.

Application Load Balancerを選択

作成したALB名を選択

ポート番号に80を入力



Fargate上でのサービス作成(5)

Container to load balance

php-sample-fargate : 80

Production listener port* 80

Production listener protocol* HTTP

Test listener
An optional test listener is used to test the new application revision before routing traffic to it.

Additional configuration

To facilitate blue/green deployments with AWS CodeDeploy, you need two target groups. Each target group binds to a separate task set in the deployment. [Learn more](#)

Target group 1 name* tg-fargat-php-sample-

Target group 1 protocol* HTTP

Target type* ip

Path pattern* /
Path pattern: The first path pattern for a listener is the default path (/), which accepts all traffic that does not match another rule. You can later add additional patterns and priority values to this listener for other services.

Health check path* /index.php
Additional health check options can be configured in the ECS console after you create your service.

Target group 2 name* tg-fargat-php-sample-

Target group 2 protocol* HTTP

Target type* ip

Path pattern* /
Path pattern: The first path pattern for a listener is the default path (/), which accepts all traffic that does not match another rule. You can later add additional patterns and priority values to this listener for other services.

Health check path* /index.php
Additional health check options can be configured in the ECS console after you create your service.

Test Listenerのチェックを外す

今回は/index.phpというページを公開するため、このページをヘルスチェックパスに設定

上記同様、/index.phpをヘルスチェックパスに設定



Fargate上でのサービス作成(6)

App Mesh

To use your service with App Mesh, you must

- Ensure your task definition is configured properly.
- Set up your service to use Service Discovery.

Service discovery (optional)

Service discovery uses Amazon Route 53 to create a namespace for your service, which allows it to be discoverable via DNS.

Enable service discovery integration

*Required Cancel Previous **Next step**

Set Auto Scaling (optional)

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your Service Auto Scaling configuration at any time to meet the needs of your application.

Service Auto Scaling Do not adjust the service's desired count
 Configure Service Auto Scaling to adjust your service's desired count

*Required Cancel Previous **Next step**

Review

Cluster: handson-cluster **Edit**

Launch type: EC2

Task Definition: php-sample

Service name: php-sample

Number of tasks: 1

Minimum healthy percent: 50

Maximum percent: 200

Configure network

VPC Id: vpc-a1594cd9 **Edit**

Subnets: subnet-c650648d, subnet-3d107f12

Selected security groups: sg-647b7310

Auto assign IP: DISABLED

Container Name: php-sample

Container Port: 80

ELB Name: php-sample

Target Group: ecs-hands-on-php-sample

Health Check Path: /index.php

Listener Port: 80

Path-pattern: /

Set Auto Scaling (optional)

not configured

Cancel Previous **Create Service**

Fargate上でのサービス作成(7)

Clusters > fargate-cluster > Service: php-sample-fargate

Service : php-sample-fargate

Update

Delete

Cluster fargate-cluster

Desired count 1

Status ACTIVE

Pending count 1

Task definition php-sample-fargate:1

Running count 0

Service type REPLICA

Launch type FARGATE

Platform version LATEST(1.2.0)

Service role aws-service-role/ecs.amazonaws.com/AWSServiceRoleForECS

Details Tasks Events Auto Scaling Deployments Metrics Logs

Last updated

Task status: Running Stopped

Filter in this page

< 1-1 > Page size 50 ▾

Task	Task Definition	Last status	Desired status	Group	Launch type	Platform version
34f8bb39-8986-489... php-sample-fargate:1	php-sample-fargate:1	PROVISIONING	RUNNING	service:php-sample-...	FARGATE	1.2.0

サービス作成が完了し、タスクが
1つ立ち上ることが確認



Fargate上でのサービス作成(8)

Services -> EC2 へ

コンソール
EC2

The screenshot shows the AWS EC2 Target Groups page. On the left, a sidebar menu highlights the 'Target Groups' option under the 'LOAD BALANCING' section. The main content area displays a table of registered targets for a target group named 'tg-fargat-php-sample-fargate-1'. A red box highlights the 'Targets' tab in the navigation bar, and another red box highlights the status 'healthy' for the first target. An orange callout box contains the following text:

ALBのターゲットグループ配下にFargateモードで起動したタスク(に割り当てられたENI)がアタッチされ、StatusがHealthyになっていることを確認

Name	Port	Protocol	Target type	Load Balancer	VPC ID	Monitoring
tg-fargat-php-sample-fargate-1	80	HTTP	ip	php-sample	vpc-003bf4740bb2e466d	Enabled
tg-fargat-php-sample-fargate-2	80	HTTP	ip		vpc-003bf4740bb2e466d	Disabled

Target group: tg-fargat-php-sample-fargate-1

Description Targets Health checks Monitoring Tag

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

IP address	Port	Availability Zone	Status
10.1.1.171	80	us-east-1a	healthy ⓘ

Availability Zones

Availability Zone	Target count	Healthy?
us-east-1a	1	Yes

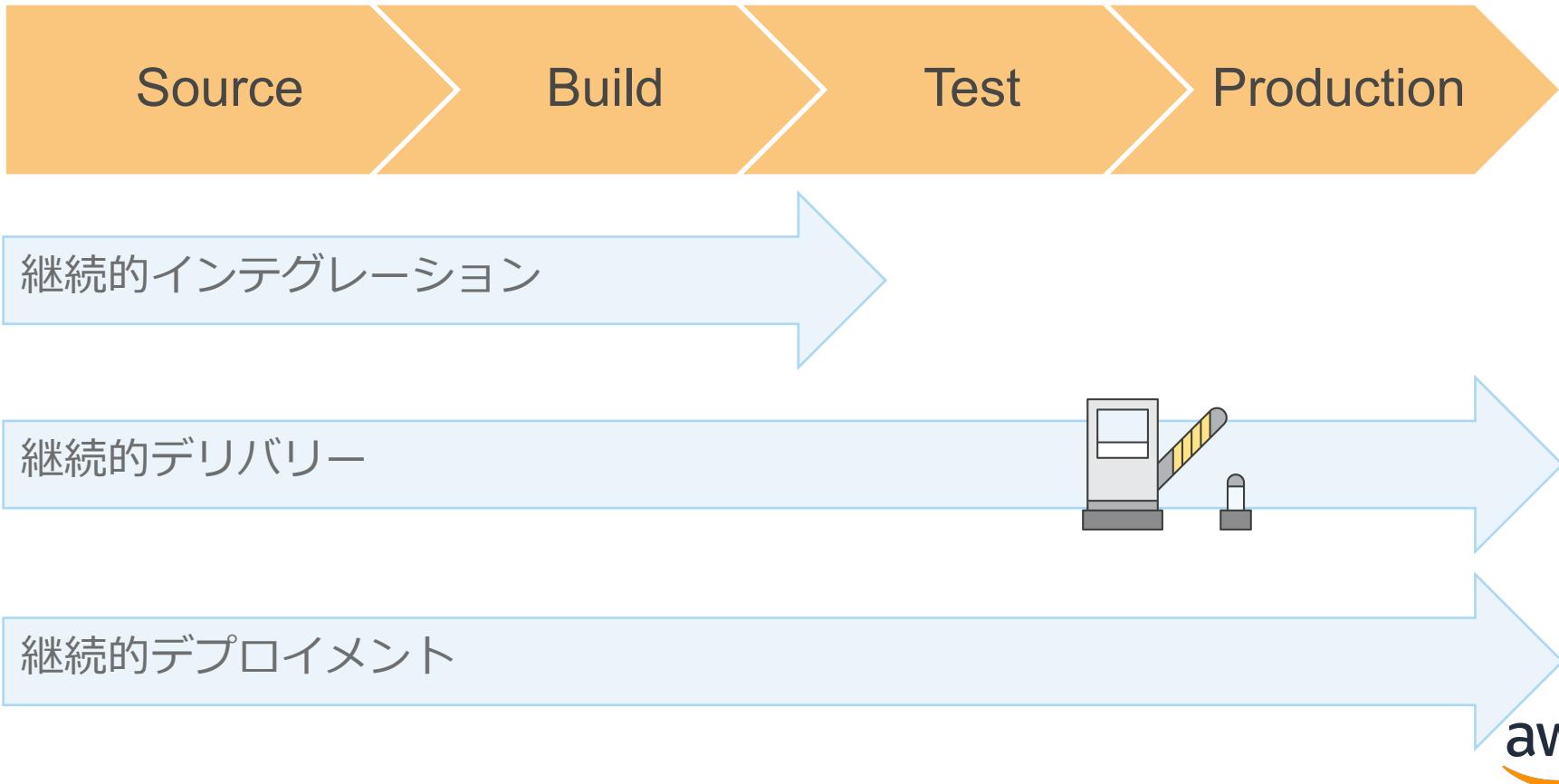
サービス作成(9)



AWS CodeServices



CI/CDパイプライン



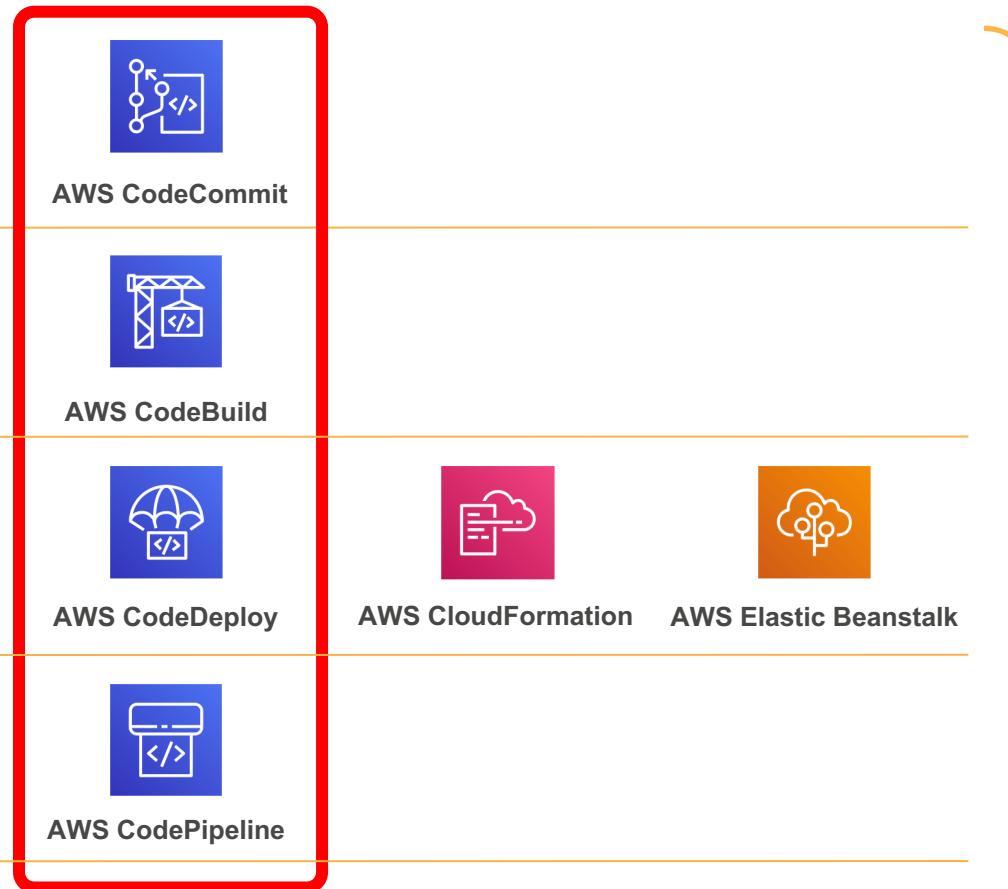
CI/CDパイプラインを実現するAWSサービス

ソースコードの
バージョン管理

ビルト自動化

デプロイ自動化

ワークフロー管理



AWS Code Services



AWS CodeCommit

- ✚ セキュア、スケーラブルなGit互換のリポジトリサービス
 - ✚ スタンダードなGit Toolからアクセス可能
 - ✚ PushなどのイベントをトリガーにSNS/Lambdaを呼び出し可能
-



AWS CodeBuild

- ✚ スケーラビリティに優れたビルドサービス
 - ✚ ソースのコンパイル、テスト、パッケージ生成をサポート
 - ✚ Dockerイメージの作成も可能
-



AWS CodeDeploy

- ✚ S3またはGitHub上のコードをあらゆるインスタンスにデプロイ
 - ✚ デプロイを安全に実行するための様々な機能を提供
 - ✚ In-place(ローリング) およびBlue/Greenのデプロイをサポート
-

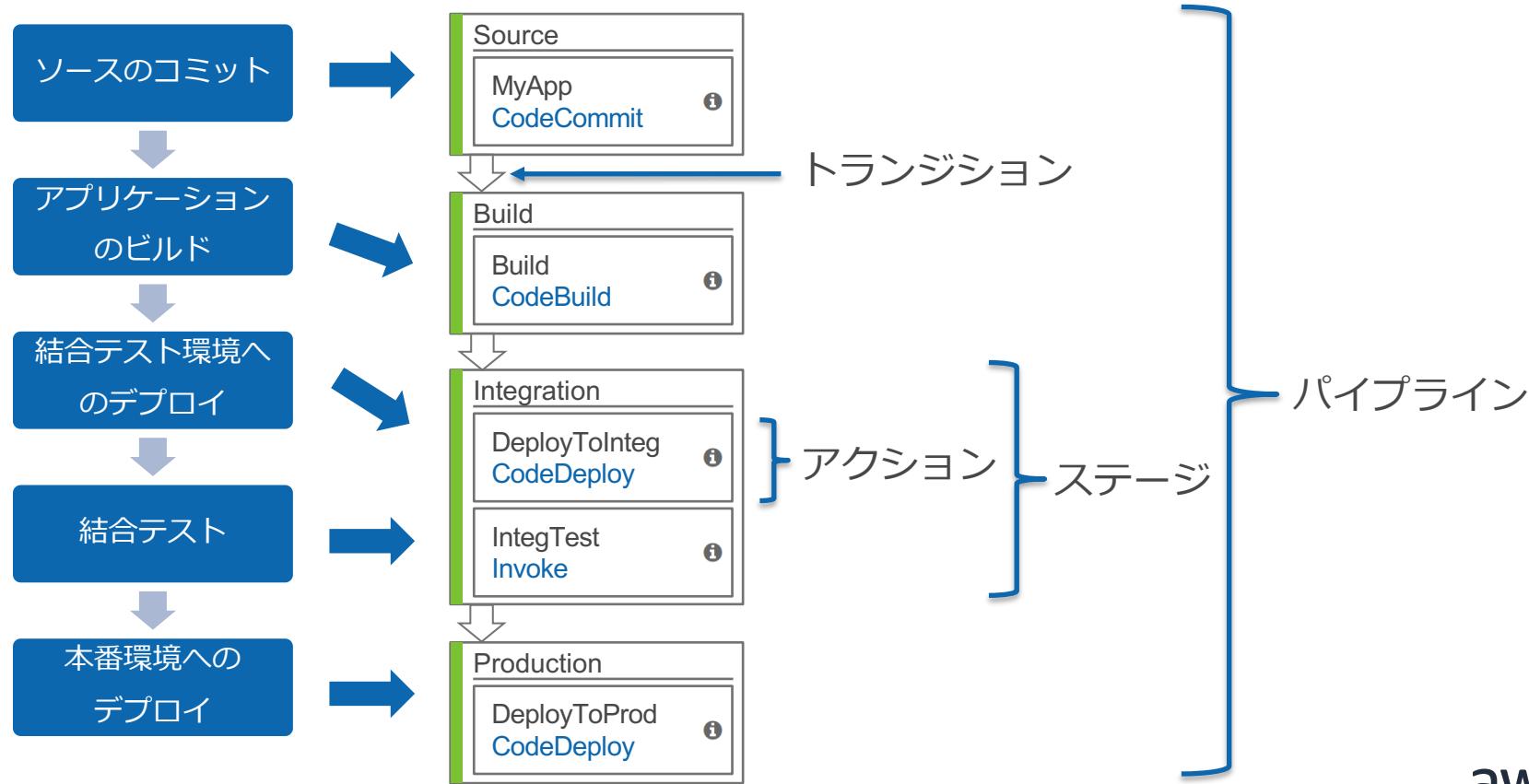


AWS CodePipeline

- ✚ リリースプロセスのモデル化と見える化を実現
 - ✚ カスタムアクションによる柔軟なパイプライン作成が可能
 - ✚ 様々なAWSサービスや3rdパーティ製品との統合をサポート
-



CodePipelineでのリリースプロセスモデル化

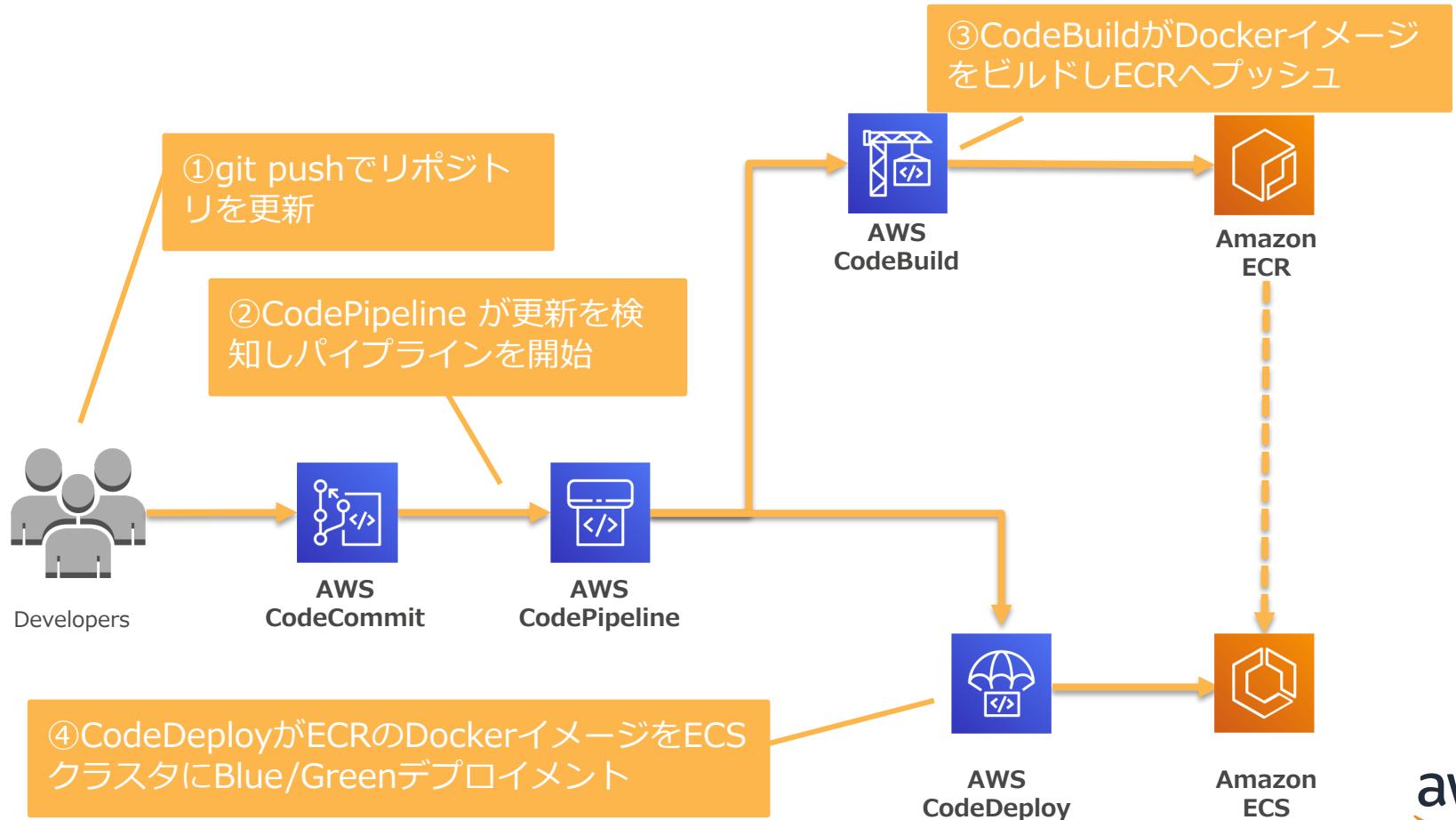


ハンズオン③

AWS Code Servicesを
利用したCI/CDパイプラインの構築



本ハンズオンで作成する継続的デプロイメント構成



CodeCommit設定

CodeCommitリポジトリ

Services -> CodeCommit -> Create repository ^

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name

php-sample

100 characters maximum. Other limits apply.

Description - optional

1,000 characters maximum

Tags

Add

Enable Amazon CodeGuru Reviewer for Java - optional

Get recommendations to improve the quality of the Java code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

Cancel

Create

CodeCommitリポジトリのクローン(1)

Cloud9

CodeCommitでの操作後、Cloud9 terminalへ

The screenshot shows the AWS CodeCommit interface. The top navigation bar includes 'Developer Tools > CodeCommit > Repositories > php-sample'. The main content area displays the repository 'php-sample'. On the right, there's a dropdown menu with three options: 'Clone URL', 'Clone HTTPS', and 'Clone SSH'. The 'Clone URL' and 'Clone HTTPS' options are highlighted with a red box and connected by a red arrow to an orange callout box containing the instruction. Below this, a warning message states: '⚠ You are signed in using federated access or temporary credentials. The only supported connection method for these sign-in types is to use the credential manager included with the AWS CLI, as documented below. To configure a connection using SSH or Git credentials over HTTPS, sign in as an IAM user.' A red arrow points from this message to the 'HTTPS' tab in the bottom navigation bar. The bottom section shows a terminal window with the command '\$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/php-sample' being run, followed by the output 'Cloning into 'php-sample'...' and 'warning: You appear to have cloned an empty repository.'

Clone URL ▾

Clone HTTPS

Clone SSH

⚠ You are signed in using federated access or temporary credentials. The only supported connection method for these sign-in types is to use the credential manager included with the AWS CLI, as documented below. To configure a connection using SSH or Git credentials over HTTPS, sign in as an IAM user.

HTTPS | SSH

#リポジトリのクローン

```
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/php-sample
Cloning into 'php-sample'...
warning: You appear to have cloned an empty repository.
```

Git client, you can install one from Git downloads page [View Git downloads page](#)



コンテンツ準備

リポジトリ配下に以下のファイルを作成

- Dockerfile と src/index.php は以前のものを php-sample/ ヘコピー
- buildspec.yml を新規作成
 - CodeBuildのビルドプロジェクト定義ファイル
 - 内容は次ページ
- appspec.yml を新規作成
- taskdef.json を新規作成

```
php-sample/
├── Dockerfile
├── buildspec.yml
├── appspec.yml
└── taskdef.json
    └── src
        └── index.php
```

Cloud9 上のローカルリポジトリディレクトリ構成



buildspec.yml

```
version: 0.2
phases:
  pre_build:
    commands:
      - $(aws ecr get-login --region $AWS_DEFAULT_REGION --no-include-email)
      - REPOSITORY_URI=01234567890.dkr.ecr.us-east-1.amazonaws.com/php-sample
      - IMAGE_TAG=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)
```

pre_buildフェーズではECRレポジトリへのログインをおこない、ビルドIDのPrefixをDockerイメージのタグに指定

```
build:
  commands:
    - docker build -t $REPOSITORY_URI:latest .
    - docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
```

buildフェーズでは環境変数で指定するレポジトリURIを取得し、Dockerイメージのビルドを実施

```
post_build:
  commands:
    - docker push $REPOSITORY_URI:latest
    - docker push $REPOSITORY_URI:$IMAGE_TAG
    - printf '{"Version":"1.0","ImageURI":"%s"}' $REPOSITORY_URI:$IMAGE_TAG > imageDetail.json
```

```
artifacts:
  files: imageDetail.json
```

post_buildフェーズではビルドしたDockerイメージをECRレポジトリにPush

buildspec.yml の拡大版は次ページを参照

CodePipelineで後続のフェーズにタグ情報を渡すため、アウトプットアーティファクトを指定



buildspec.yml の修正

```
version: 0.2
phases:
  pre_build:
    commands:
      - $(aws ecr get-login --region $AWS_DEFAULT_REGION --no-include-email)
      - REPOSITORY_URI=${Your account ID}.dkr.ecr.us-east-1.amazonaws.com/php-sample
      - IMAGE_TAG=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)

  build:
    commands:
      - docker build -t $REPOSITORY_URI:latest .
      - docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG

  post_build:
    commands:
      - docker push $REPOSITORY_URI:latest
      - docker push $REPOSITORY_URI:$IMAGE_TAG
      - printf '{"Version":"1.0","ImageURI":"%s"}' $REPOSITORY_URI:$IMAGE_TAG > imageDetail.json

artifacts:
  files: imageDetail.json
```

アカウントIDを自身のものに修正してください

※お急ぎの方はZipファイル内のサンプルコードをご利用ください。



appspec.yml

```
version: 0.0
Resources:
- TargetService:
  Type: AWS::ECS::Service
  Properties:
    TaskDefinition: "<TASK_DEFINITION>"
    LoadBalancerInfo:
      ContainerName: "php-sample-fargate"
      ContainerPort: "80"
```

AppSpecファイルはCodeDeployがデプロイを管理するために使用するファイルです

TASK_DEFINITIONプレースホルダーはPipelineが走ると自動的に置き換わります

今回の対象はECSなのでECSタスク定義、ロードバランサー情報などを含める必要があります

修正せずそのままお使いください

※お急ぎの方はZipファイル内のサンプルコードをご利用ください。



taskdef.json

ECS -> Task Definition -> php-sample-fargate -> 最新のRevisionを表示

The screenshot shows the AWS ECS Task Definitions console. On the left, there's a sidebar with links for Amazon ECS (Clusters, Task Definitions, Account Settings), Amazon EKS (Clusters), Amazon ECR (Repositories), AWS Marketplace (Discover software), and Subscriptions. The 'Task Definitions' link is highlighted with an orange border. The main area shows a breadcrumb path: Task Definitions > php-sample-fargate > 14. Below it, the title 'Task Definition: php-sample-fargate:1' is displayed. A sub-instruction 'View detailed information for your task definition. To modify the task definition...' is present. At the top right, there are 'Create new revision' and 'Actions' buttons. Below them are three tabs: 'Builder' (selected), 'JSON' (highlighted with a red box and arrow), and 'Tags'. The 'JSON' tab contains the task definition configuration in JSON format, which is highlighted with a red box. An orange callout box with the text 'Jsonタブを表示し、内容を全てコピー' (Display the JSON tab and copy all content) points to this red box.

```
{  
    "ipcMode": null,  
    "executionRoleArn": "arn:aws:iam::335093717998:role/ecsTaskExecutionRole",  
    "containerDefinitions": [  
        {  
            "dnsSearchDomains": null,  
            "logConfiguration": {  
                "logDriver": "awslogs",  
                "secretOptions": null,  
                "options": {  
                    "awslogs-group": "/ecs/php-sample-fargate",  
                    "awslogs-region": "us-east-1",  
                    "awslogs-stream-prefix": "ecs"  
                }  
            }  
        }  
    ]  
}
```



taskdef.json の修正

```
...  
  "memoryReservation": 128,  
    "volumesFrom": [],  
    "stopTimeout": null,  
    "image": "<IMAGE1_NAME>,"  
    "startTimeout": null,  
    "dependsOn": null,  
    "disableNetworking": null,  
    "interactive": null,  
    "healthCheck": null,  
  ...
```

taskdef.jsonはFargateで起動されるタスクの情報が記載されたファイルです。今回はPipelineがトリガーされると動的にタスク定義を更新します

先ほどコピーした内容を全て貼り付ける

imageを<IMAGE1_NAME>に変更。このプレースホルダーはBuildアクション後に動的にアップデートされます



リポジトリへのpush

```
#リモートリポジトリへのPush
```

```
$ cd php-sample  
$ git add -A  
$ git commit -m "my first commit"  
$ git push origin master
```

The screenshot shows the AWS CodeCommit interface. On the left, there's a sidebar with navigation links: Source (CodeCommit), Getting started, Repositories, Code, Pull requests, **Commits**, Branches, Git tags, Settings, Build (CodeBuild), Deploy (CodeDeploy), and Pipeline (CodePipeline). The main area displays the 'Commits' page for the 'php-sample' repository. The commit history table has columns: Commit ID, Commit message, Commit date, Author, and Actions. One commit is listed: '714bda4b' with the message 'my first commit', dated 'Just now', and author '"yourname"'. An orange arrow points from this commit row to an orange callout box at the bottom right.

CodeCommitコンソールから
Commit履歴が見えることを確認

https://docs.aws.amazon.com/ja_jp/codecommit/latest/userguide/troubleshooting-ch.html#troubleshooting-macoshttps



CodePipeline設定

パイプライン作成(1)

Services -> CodePipeline -> Create pipeline ^

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Choose pipeline settings

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
php-sample-pipeline

No more than 100 characters

Service role
 New service role
Create a service role in your account Existing service role
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-us-east-1-php-sample-pipeline

Type your service role name

Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

▶ Advanced settings

Cancel **Next**

php-sample-pipeline と入力



パイプライン作成(2)

Step 1 Developer Tools > CodePipeline > Pipelines > Create new pipeline

Choose pipeline settings

Step 2

Add source stage

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

Add source stage

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

Repository name
Choose a repository that you have already created where you have pushed your source code.

php-sample

Branch name
Choose a branch of the repository

master

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

Amazon CloudWatch Events

AWS CodePipeline
Configure AWS CodePipeline to check periodically for changes

SourceプロバイダーにCodeCommitの
レポジトリとmasterブランチを指定

Cancel Previous Next

aws

パイプライン作成(3)

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add build stage

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build tools, and more.

AWS CodeBuild

Region
US East - (N. Virginia)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

Search input field or Create project []

Cancel Previous Skip build stage Next

BuildプロバイダーにCodeBuildを指定



パイプライン作成(4)

Project configuration

Project name
php-sample-build

A project name must be unique across all AWS accounts in the number of the capital characters -

Environment

Environment image
 Managed image Use an image managed by AWS CodeBuild Custom image Specify a Docker image

Operating system
Ubuntu

The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See Docker Images Provided by CodeBuild for details.

Runtime(s)
Standard

Image
aws/codebuild/standard:1.0

Image version
aws/codebuild/standard:1.0-1.8.0

Environment type
Linux

Privileged
 Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Service role
 New service role Create a service role in your account Existing service role Choose an existing service role from your account

Role name
codebuild-php-sample-build-service-role

Type your service role name

Buildプロジェクト名を指定
php-sample-build

Build環境を指定

- Environment Image : Managed Image
- Operating System : Ubuntu
- Runtime: Standard
- Image: aws/codebuild/standard:1.0
- Image version: aws/codebuild/standard: 1.0-1.8.0
- Privileged の項目にチェックをつける
- Role name に codebuild-php-sample-build-service-role を入力

その他はデフォルトのままにし Continue to Codepipeline をクリック、CodePipeline へ戻り、Nextを選択



Successfully created php-sample-build in CodeBuild.

Cancel

Previous

Skip build stage

Next

パイプライン作成(5)

Add deploy stage

Deploy - optional

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Region

AWS CodeDeploy application name
Choose one of your existing applications, or create a new one in AWS CodeDeploy.

AWS CodeDeploy deployment group
Choose one of your existing deployment groups, or create a new one in AWS CodeDeploy.

Amazon ECS task definition
Choose the input artifact where your Amazon ECS task definition file is stored. If other than the default file path, specify the path and filename of your task definition file.

The default path is taskdef.json.

AWS CodeDeploy AppSpec file
Choose the input artifact where your AWS CodeDeploy AppSpec file is stored. If other than the default file path, specify the path and filename of your AppSpec file.

Dynamically update task definition image - optional
You can provide an input artifact and a placeholder name for the container definition image that will be used to dynamically update a task definition. You can specify multiple input artifacts and placeholders.

Input artifact with image details

Placeholder text in the task definition

DeployプロバイダーにAmazon ECS (Blue/Green)を指定

Application name に AppECS-xxx を指定

Deployment groupに DgpECS-xxx を指定

Task DefinitionとAppSpec Fileのinput artifactにBuildArtifactを選択、これは後ほど修正します

Reviewページ確認後、
Create pipelineを選択

Cancel

Previous

Create pipeline



CodeBuildのIAM Role編集 (1)

Services -> IAM へ

コンソール
IAM

The screenshot shows the AWS IAM Roles page. A red box highlights the 'Roles' button in the left sidebar. The main area has a search bar with 'codebuild' typed into it. Below the search bar, a dropdown menu shows 'Role name' and the selected role 'codebuild-php-sample-build-service-role'. A red box highlights this selected role.

Summary

Role ARN	arn:aws:iam::[REDACTED]:role/service-role/codebuild-php-sample-build-service-role
Role description	Edit
Instance Profile ARNs	[REDACTED]
Path	/service-role/
Creation time	2019-03-18 02:21 UTC+0900
Maximum CLI/API session duration	1 hour Edit

Permissions

- Permissions policies (1 policy applied)
- [Attach policies](#)

Policy name: CodeBuildBasePolicy-php-sample-build-us-east-1

今回定義したビルド定義(buildspec.yml)では、ビルド内でECRに対してDockerイメージをPushしているため、CodeBuildに割り当てたIAMロール
(codebuild-php-sample-build-service-role)に対してECRの操作権限を付与する必要があります

IAM Roleより、codebuildを検索し、先ほど自動作成されたロール codebuild-php-sample-build-service-role を選択

Attach policiesを選択



CodeBuildのIAM Role編集 (2)

Attach Policy

Select one or more policies to attach. Each role can have up to 10 policies attached.

Filter: Policy Type ▾ container Showing 9 results

	Policy Name	Attached Entities	Creation Time	Edited Time
<input type="checkbox"/>	AmazonEC2ContainerServiceAuto...	1	2016-05-13 08:25 UTC+0900	2016-05-13 08:25 UTC+0900
<input type="checkbox"/>	AmazonEC2ContainerServiceforE...	1	2015-03-20 03:45 UTC+0900	2017-05-18 08:09 UTC+0900
<input type="checkbox"/>	AmazonEC2ContainerServiceRole	1	2015-04-10 01:14 UTC+0900	2016-08-11 22:08 UTC+0900
<input checked="" type="checkbox"/>	AmazonEC2ContainerRegistryFull...	0	2015-12-22 02:06 UTC+0900	2015-12-22 02:06 UTC+0900
<input type="checkbox"/>	AmazonEC2ContainerRegistryPow...	0	2015-12-22 02:05 UTC+0900	2016-10-12 07:28 UTC+0900
<input type="checkbox"/>	AmazonEC2ContainerRegistryRea...	0	2015-12-22 02:04 UTC+0900	2016-10-12 07:08 UTC+0900
<input type="checkbox"/>	AmazonEC2ContainerServiceEven...	0	2017-05-31 01:51 UTC+0900	2017-05-31 01:51 UTC+0900
<input type="checkbox"/>	AmazonEC2ContainerServiceFullA...	0	2015-04-25 01:54 UTC+0900	2017-06-08 09:18 UTC+0900
<input type="checkbox"/>	AWSelasticBeanstalkMulticontain...	0	2016-02-09 08:15 UTC+0900	2016-06-07 08:45 UTC+0900

AmazonEC2ContainerRegistryPowerUser
をアタッチ

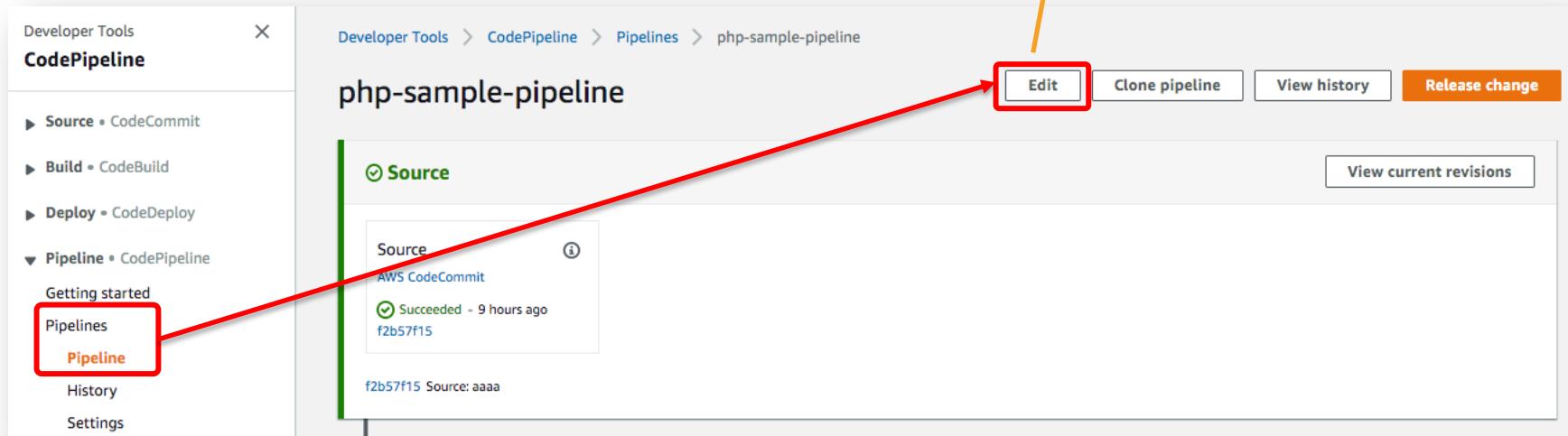
Cancel Attach Policy

CodePipeline の設定編集 (1)

Services -> CodePipeline へ

Code Pipeline
コンソール

先ほど作成したphp-sample-pipelineを選択し、
Edit



CodePipeline の設定編集 (2)

Editing: php-sample-pipeline

Delete

Cancel

Save

Edit: Source

Edit stage

Source

?

AWS CodeCommit

+ Add stage

Edit: Build

Edit stage

Build

?

AWS CodeBuild

+ Add stage

Edit: Deploy

Edit stage

Deploy

?

Amazon ECS (Blue/Green)

+ Add stage

Deployステージを編集

Cancel

Delete

Done

Edit: Deploy

+ Add action group

Deploy

Amazon ECS (Blue/Green)



+ Add action



CodePipeline の設定編集 (3)

Input artifacts

Choose an input artifact for this action. [Learn more](#)

BuildArtifact

Add

No more than 100 characters

Input artifactsで Addを押し、
SourceArtifactを追加する

Input artifacts

Choose an input artifact for this action. [Learn more](#)

BuildArtifact

SourceArtifact

Remove

Add

No more than 100 characters



CodePipeline の設定編集 (4)

AWS CodeDeploy application name
Choose one of your existing applications, or create a new one in AWS CodeDeploy.

AppECS-fargate-cluster-php-sample-fargate

AWS CodeDeploy deployment group
Choose one of your existing deployment groups, or create a new one in AWS CodeDeploy.

DgpECS-fargate-cluster-php-sample-fargate

Amazon ECS task definition
Choose the input artifact where your Amazon ECS task definition file is stored. If other than the default file path, specify the path and filename of your task definition file.

SourceArtifact taskdef.json

The default path is taskdef.json.

AWS CodeDeploy AppSpec file
Choose the input artifact where your AWS CodeDeploy AppSpec file is stored. If other than the default file path, specify the path and filename of your AppSpec file.

SourceArtifact appspec.yaml

Dynamically update task definition image - *optional*
You can provide an input artifact and a placeholder name for the container definition image that will be used to dynamically update a task definition. You can specify multiple input artifacts and placeholders.

Input artifact with image details

BuildArtifact

Placeholder text in the task definition

IMAGE1_NAME Remove

Cancel

Developer Tools > CodePipeline > Pipelines > php-sample-pipeline > Edit php-sample-pipeline

Editing: php-sample-pipeline

ECS task definitionのinput artifactを **SourceArtifact**とする

CodeDeploy AppSpec file のinput artifactを **SourceArtifact**とする

Task definitionのimageを動的に変更するための情報が記載されたinput artifactを **BuildArtifact**とする

Image placeholderを IMAGE1_NAME とする

Pipeline編集のページでSaveする



CodeDeploy の設定編集 (1)

CodeDeploy 選択

The screenshot shows the AWS CodeDeploy console interface. On the left, a sidebar menu under 'Developer Tools' lists 'Source' (CodeCommit), 'Build' (CodeBuild), 'Deploy' (CodeDeploy), 'Getting started', 'Deployments', 'Applications', and 'Pipeline' (CodePipeline). The 'Applications' section is highlighted with a red box, and 'Application' is selected. In the main content area, the path 'Developer Tools > CodeDeploy > Applications > AppECS-fargate-cluster-php-sample-fargate' is shown. The application name 'AppECS-fargate-cluster-php-sample-fargate' is also highlighted with a red box. Below it, the 'Deployment groups' tab is selected, showing a table with one row. The row contains the deployment group name 'DgpECS-fargate-cluster-php-sample-fargate'. A red box highlights this row. To the right of the table, there are buttons for 'View details', 'Edit' (highlighted with a red box), and 'Create deployment group'. An orange callout box points to the 'Edit' button with the text: 'CodeDeploy > Application > AppECS-fargate-cluster-php-sample-fargate を選択し、Deployment groupをEdit'.

CodeDeploy > Application > AppECS-fargate-cluster-php-sample-fargate を選択し、Deployment groupをEdit

Name	Status	Last attempted deployment	Last successful deployment	Trigger count
DgpECS-fargate-cluster-php-sample-fargate	-	-	-	0

CodeDeploy の設定編集 (2)

Deployment settings

Traffic rerouting
Choose whether traffic reroutes to the replacement environment immediately or waits for you to start the rerouting process

Reroute traffic immediately
 Specify when to reroute traffic

Deployment Configuration
CodeDeployDefault.ECSAllAtOnce

Original revision termination
Specify how long CodeDeploy waits before it terminates the original task set. After termination starts, you cannot rollback manually or automatically

Days Hours Minutes

0	0	5
---	---	---

▶ Advanced - optional

Cancel **Save changes**

Deployment settingsで Original revision termination を
0 Days, 0 Hours, 5 Minutesに変更



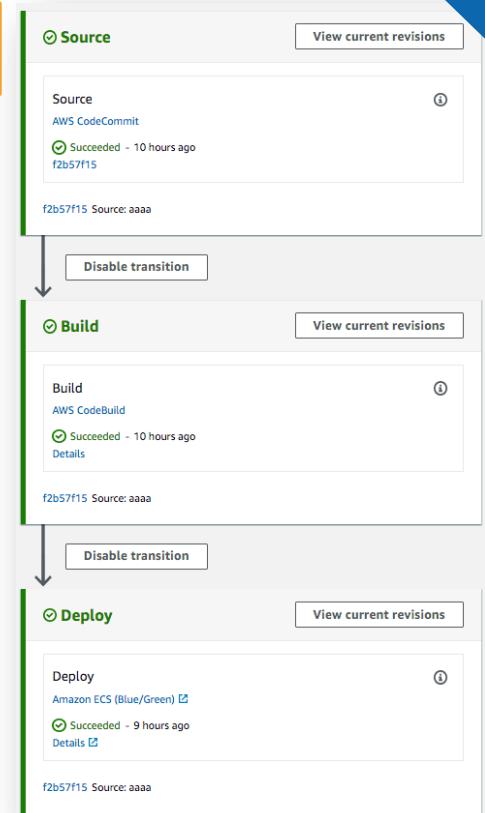
パイプライン実行

CodePipeline 選択

CodePipelineからphp-sample-pipelineを選択し、Release changeをクリック

The screenshot shows the AWS CodePipeline console interface. On the left, there's a sidebar with 'Developer Tools' and 'CodePipeline' sections. Under 'CodePipeline', 'Pipeline' is selected. In the main area, the pipeline named 'php-sample-pipeline' is displayed. It has three stages: 'Source', 'Build', and 'Deploy'. The 'Source' stage is currently active, showing a successful run from 'AWS CodeCommit' with revision 'f2b57f15' completed 10 hours ago. Below the stage details, it says 'f2b57f15 Source: aaaa'. At the top of the pipeline view, there are buttons for 'Edit', 'Clone pipeline', 'View history', and 'Release change'. The 'Release change' button is highlighted with a red box.

パイプラインのリリースが始まります。パイプラインが正常に動作しリリースが完了することを確認して下さい。
パイプラインの完了には10分程度かかります。



Code Pipeline
コンソール

デプロイ状況の確認 (1)

CodeDeploy -> Deployments へ

The screenshot shows the AWS CodeDeploy console under the 'Developer Tools' section. The left sidebar has a 'Deployments' section expanded, showing 'Getting started', 'Deployments' (which is selected), 'Applications', and 'Deployment configurations'. The main area shows a table titled 'Deployment history' with columns: Deployment ID, Status, Deployment type, Compute platform, Application, Deployment group, Revision location, and Initiating event. A deployment named 'd-3FAJ6T5QB' is listed with the status 'In progress'. This row is highlighted with a red box.

Deployment ID	Status	Deployment type	Compute platform	Application	Deployment group	Revision location	Initiating event
d-3FAJ6T5QB	In progress	Blue/green	Amazon ECS	AppECS-fargate-cluster-php-sample-fargate	DgECS-fargate-cluster-php-sample-fargate	d144de3a...	user

Developer Tools > CodeDeploy > Deployments > d-3FAJ6T5QB

d-3FAJ6T5QB

[Stop deployment](#) [Stop and roll back deployment](#) [Terminate original task set](#)

This screenshot shows the detailed view for deployment 'd-3FAJ6T5QB'. It includes two main sections: 'Deployment status' and 'Traffic shifting progress'. The 'Deployment status' section lists four steps: Step 1 (Deploying replacement task set, Completed, Succeeded), Step 2 (Rerouting production traffic to replacement task set, 100% traffic shifted, Succeeded), Step 3 (Wait 5 minutes 0 seconds, Waiting, In progress), and Step 4 (Terminate original task set, Not started, In progress). The 'Traffic shifting progress' section shows a 0% value for the 'Original' task set and a 100% value for the 'Replacement' task set, indicating the traffic has been successfully shifted.

Step	Action	Status	Progress
Step 1:	Deploying replacement task set	Completed	Succeeded
Step 2:	Rerouting production traffic to replacement task set	100% traffic shifted	Succeeded
Step 3:	Wait 5 minutes 0 seconds	Waiting	In progress
Step 4:	Terminate original task set	Not started	In progress

Traffic shifting progress

Task Set	Progress
Original	0%
Replacement	100%

Original task set not serving traffic Replacement task set serving traffic

Deploymentの進行状況を
CodeDeployコンソールより確認
できます。

Original Taskが削除されるまで
5分間待ちます。



Code Deploy
コンソール

デプロイ状況の確認 (2)

Services -> ECS -> Cluster -> Service へ

ECS
コンソール

Clusters > fargate-cluster > Service: php-sample-fargate

Service : php-sample-fargate

Cluster	fargate-cluster
Status	ACTIVE
Task definition	php-sample-fargate:15
Service type	REPLICAS
Launch type	FARGATE
Platform version	1.3.0
Service role	AWSServiceRoleForECS

Update

Delete

Task Placement

Step 1: No strategies
Constraint: No constraints

Blue/green deployment

Deployment ID	d-3FAJ6T5QB
Type	Blue/green
Started by	AWS CodeDeploy
Status	Succeeded

CodeDeploy deployment group: DgxECS-fargate-cluster-php-sample-fargate

Start time: 2019-07-22 02:56:48
End time: 2019-07-22 03:04:04
Deployment history: DgxECS-fargate-cluster-php-sample-fargate

Task set ID	Environment	Task set status	Traffic	Desired count	Running count	Pending count
ecs-svc/92233704730797827...	-	PRIMARY	100%	1	1	0

ServiceのDeploymentsタブよりBlue/Green Deploymentの状況を確認することもできます

Eventsタブでは、タスクの起動や、Drainingの履歴を確認することができます

Details	Tasks	Events	Auto Scaling	Deployments	Metrics	Tags	Logs
Last updated on July 22, 2019 3:07:32 PM (0 ago) Filter in this page							

Event Id	Event Time	Message
18cde49d-3c42-432b-90b9-4ed17e02120b8	2019-07-22 15:04:43 +0900	service php-sample-fargate has reached a steady state.
c03b309d-e9b2-438f-857c-08bfad549993	2019-07-22 15:04:33 +0900	(service php-sample-fargate, taskSet ecs-svc/9223370473118860118) updated state to STEADY_STATE.
d781b5c6-1963-a5a2-b4f1-117cc6b71ac9	2019-07-22 15:04:23 +0900	(service php-sample-fargate, taskSet ecs-svc/9223370473118860118) has stopped 1 running tasks: task 243ec628-daae-4294-9511-9d6e0d352d0.
7cb625c2-b95a-44fc-9695-20710a95f50	2019-07-22 15:04:14 +0900	(service php-sample-fargate, taskSet ecs-svc/9223370473118860118) has begun draining connections on 1 tasks.
2862a0ce-57ed-480a-a904-906ca6d190ac	2019-07-22 15:04:14 +0900	service php-sample-fargate deregistered 1 targets in target-group tg-fargat-php-sample-fargate-2
e0041b86-0e79-450d-af73-e236fbfa163a	2019-07-22 15:04:14 +0900	service php-sample-fargate updated computedDesiredCount for taskSet ecs-svc/9223370473118860118 to 0.
054a15e5-4054-45cd-960a-8dfb4f459771	2019-07-22 15:04:14 +0900	service php-sample-fargate has reached a steady state.
da8bd64c-2079-4094-a0a0-39ff56686e8	2019-07-22 14:57:46 +0900	(service php-sample-fargate, taskSet ecs-svc/9223370473079782723) updated state to STEADY_STATE.
03c5d5d-8417-432b-a80e-78fd1d102aba	2019-07-22 14:57:34 +0900	(service php-sample-fargate, taskSet ecs-svc/9223370473079782723) registered 1 targets in target-group tg-fargat-php-sample-fargate-1
881c55f9-bc09-6f6d-a2dd-65202c204d01	2019-07-22 14:57:02 +0900	(service php-sample-fargate, taskSet ecs-svc/9223370473079782723) has started 1 tasks: task 764da32c-b517-4882-bced-761c46b01b9d.
dd580ebd-adce-4a40-971f-36be60a63a09	2019-07-22 14:57:01 +0900	service php-sample-fargate updated computedDesiredCount for taskSet ecs-svc/9223370473079782723 to 1.

aws

継続的デプロイメント パイプライン動作確認

リポジトリへのpush

Services -> Cloud9 へ

```
#アプリのバージョンアップ(index.phpを適当に書き変える)  
$ vim src/index.php
```

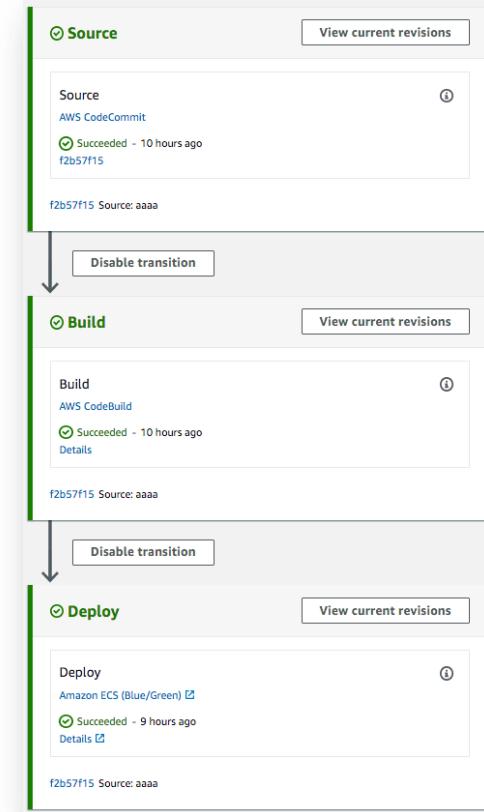
#リモートリポジトリへのPush

```
$ git add -A  
$ git commit -m "pipeline test"  
$ git push origin master
```

継続的デプロイメント動作確認

以下のポイントを確認する

- CodePipelineのパイプラインが自動的に開始され、正常に完了すること
- CodeBuildで新しいBuild runが開始されていること
- CodeDeployで新しいDeploymentが開始されていること
- ECSで新しいタスク定義でサービス更新が行われていること
- サイトにアクセスし、新しいアプリケーションがデプロイされていること



後片付け



作成したリソース削除

- CodePipelineパイプライン
- CodeDeployアプリケーション
- CodeBuildビルドプロジェクト
- CodeCommitレポジトリ
- S3バケット
- 各種IAMロール
- 各種CloudWatchログ
- ECRレポジトリ
- ECSクラスター
- ECSタスク定義
- ALBターゲットグループ
- ALB
- Cloud9環境
- CloudFormation



CodePipeline の削除

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar menu is open under 'Developer Tools' for 'CodePipeline'. The 'Pipelines' option is selected and highlighted with a red box. A red arrow points from this selection to the 'Delete pipeline' button in the main content area. The main content area displays a table of pipelines. One pipeline, named 'php-sample-pipeline', is selected and highlighted with a blue box. This pipeline has a status of 'Succeeded' and was last executed 20 minutes ago. The 'Delete pipeline' button is also highlighted with a red box.

Name	Most recent execution	Latest source revisions	Last executed
php-sample-pipeline	Succeeded	Source - 0c4482d5: pipeline test	20 minutes ago

作成したPipelineを選択し、Deleteを選択します。

CodeDeploy の削除

The screenshot shows the AWS CodeDeploy console interface. On the left, the navigation pane is visible with the 'Applications' option selected, indicated by a red box. In the main content area, the 'Applications' page lists an application named 'AppECS-fargate-cluster-php-sample-fargate'. A red box highlights this application name. To the right, a detailed view of the application 'sample-fargate' is shown, also with its name highlighted by a red box. At the top right of this detail view, there is a 'Delete application' button, which is also highlighted with a red box.

作成したApplicationを選択し、Deleteを選択します。

Name	Status	Last attempted deployment	Last successful deployment	Trigger count
DgpECS-fargate-cluster-php-sample-fargate	Succeeded	Apr 12, 2020 1:05 PM (UTC+9:00)	Apr 12, 2020 1:05 PM (UTC+9:00)	0

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

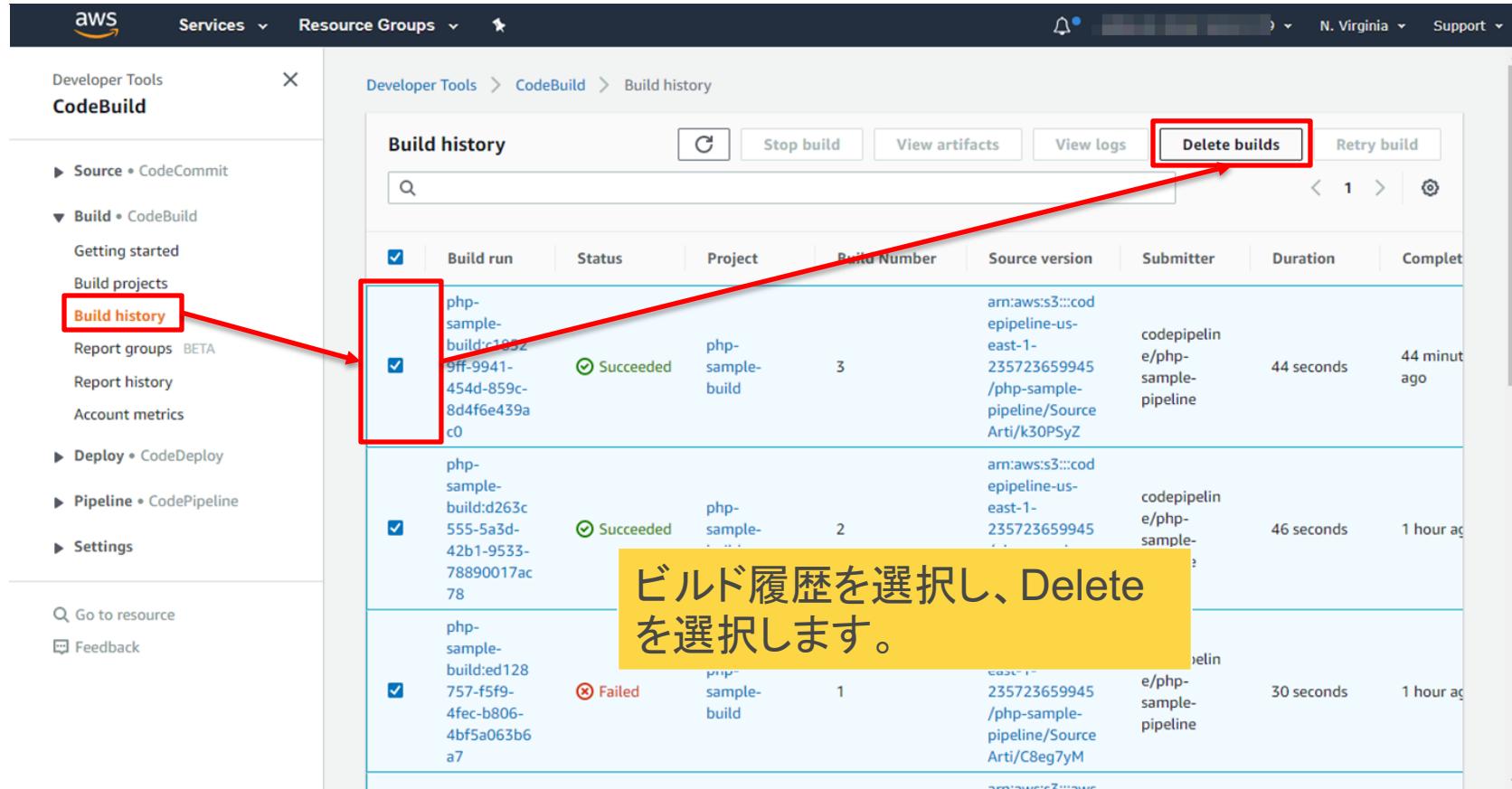
CodeBuild の削除

The screenshot shows the AWS CodeBuild 'Build projects' page. On the left sidebar, under 'Build • CodeBuild', the 'Build projects' link is highlighted with a red box. A red arrow points from this link to the blue circular icon next to the project name 'php-sample-build' in the main table. Another red arrow points from this icon to the 'Delete build project' button at the top of the page.

Name	Source provider	Repository	Description
php-sample-build	AWS CodePipeline	-	-

作成したProjectを選択し、Deleteを選択します。

CodeBuild の削除



Developer Tools > CodeBuild > Build history

Build history

Stop build View artifacts View logs Delete builds Retry build

Build run	Status	Project	Build Number	Source version	Submitter	Duration	Completion
php-sample-build:c1032 9ff-9941- 454d-859c- 8d4f6e439a c0	Succeeded	php-sample-build	3	arn:aws:s3:::cod epipeline-us- east-1- 235723659945 /php-sample- pipeline/Source Arti/k30PSyZ	codepipeline/php- sample-pipeline	44 seconds	44 minutes ago
php-sample-build:d263c 555-5a3d- 42b1-9533- 78890017ac 78	Succeeded	php-sample-build	2	arn:aws:s3:::cod epipeline-us- east-1- 235723659945	codepipeline/php- sample-pipeline	46 seconds	1 hour ago
php-sample-build:ed128 757-f5f9- 4fec-b806- 4bf5a063b6 a7	Failed	php-sample-build	1	arn:aws:s3:::cod epipeline-us- east-1- 235723659945 /php-sample- pipeline/Source Arti/C8eg7yM	codepipeline/php- sample-pipeline	30 seconds	1 hour ago

ビルド履歴を選択し、Deleteを選択します。

CodeCommit の削除

The screenshot shows the AWS CodeCommit service in the AWS Management Console. On the left, the navigation pane is open, showing the 'Source' section with 'CodeCommit'. Under 'Repositories', the 'Repositories' link is highlighted with a red box. In the main content area, the 'Repositories' tab is selected. A red box highlights the 'Delete repository' button in the top right corner of the toolbar. Another red box highlights the small blue circular icon next to the repository name 'php-sample' in the list. A large yellow callout box in the bottom right corner contains the Japanese text: '作成したRepositoryを選択し、Deleteを選択します。' (Select the created Repository and choose Delete). The repository list table has columns for Name, Description, Last modified, and Clone URL (with options for HTTPS, SSH, and HTTPS (GRC)).

Name	Description	Last modified	Clone URL
php-sample	-	47 minutes ago	<input type="button" value="HTTPS"/> <input type="button" value="SSH"/> <input type="button" value="HTTPS (GRC)"/>

S3 bucket の削除

The screenshot shows the AWS S3 service dashboard. On the left sidebar, the 'Buckets' link is highlighted with a red box. In the main content area, a list of 12 buckets is displayed. A yellow callout box points to the first bucket in the list, which has a blue circular selection icon and the identifier 'codepipeline-us-east-1-235723659945'. Above this bucket, another red box highlights the 'Delete' button in the top right corner of the table header. A large red arrow points from the 'Delete' button towards the selected bucket.

Amazon S3

Buckets

Batch Operations

Access analyzer for S3

Block public access (account settings)

Feature spotlight

Buckets (12)

Find bucket by name

Name Region Access Bucket created

codepipeline-us-east-1-235723659945 US East (N. Virginia) us-east-1 Objects can be public 2020-04-09T06:57:29.000Z

Copy ARN Empty Delete Create bucket

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

自動作成したBucketを選択し、Deleteを選択します。

S3 bucket の削除

The screenshot shows the 'Delete bucket' confirmation dialog in the AWS S3 console. At the top, there is a message box with a red border containing an error icon and the text: 'This bucket is not empty' followed by a detailed message: 'Buckets must be empty before they can be deleted. To delete all objects in the bucket, use the [empty bucket configuration](#)'. A red rectangle highlights the 'empty bucket configuration' link. Below this, the bucket name 'codepipeline-us-east-1-235723659945' is displayed. A text input field contains the bucket name. At the bottom, there are 'Cancel' and 'Delete bucket' buttons.

Bucketが空ではないので、空にする必要があります。リンクをクリックします。

S3 bucket の削除

The screenshot shows the AWS S3 'Empty bucket' confirmation dialog. At the top, there is a warning message: '⚠️ • Emptying the bucket cannot be undone.
• Objects added to the bucket while the empty bucket action is in progress will be deleted.' Below this, a section titled 'Delete all objects in bucket codepipeline-us-east-1-235723659945' contains a text input field with the bucket name 'codepipeline-us-east-1-235723659945'. A red box highlights this input field, and a red arrow points from it to the 'Empty' button, which is also highlighted with a red box. To the left of the 'Empty' button is a 'Cancel' link.

確認のためにbucket Nameを
入力し、Emptyをクリックします。

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

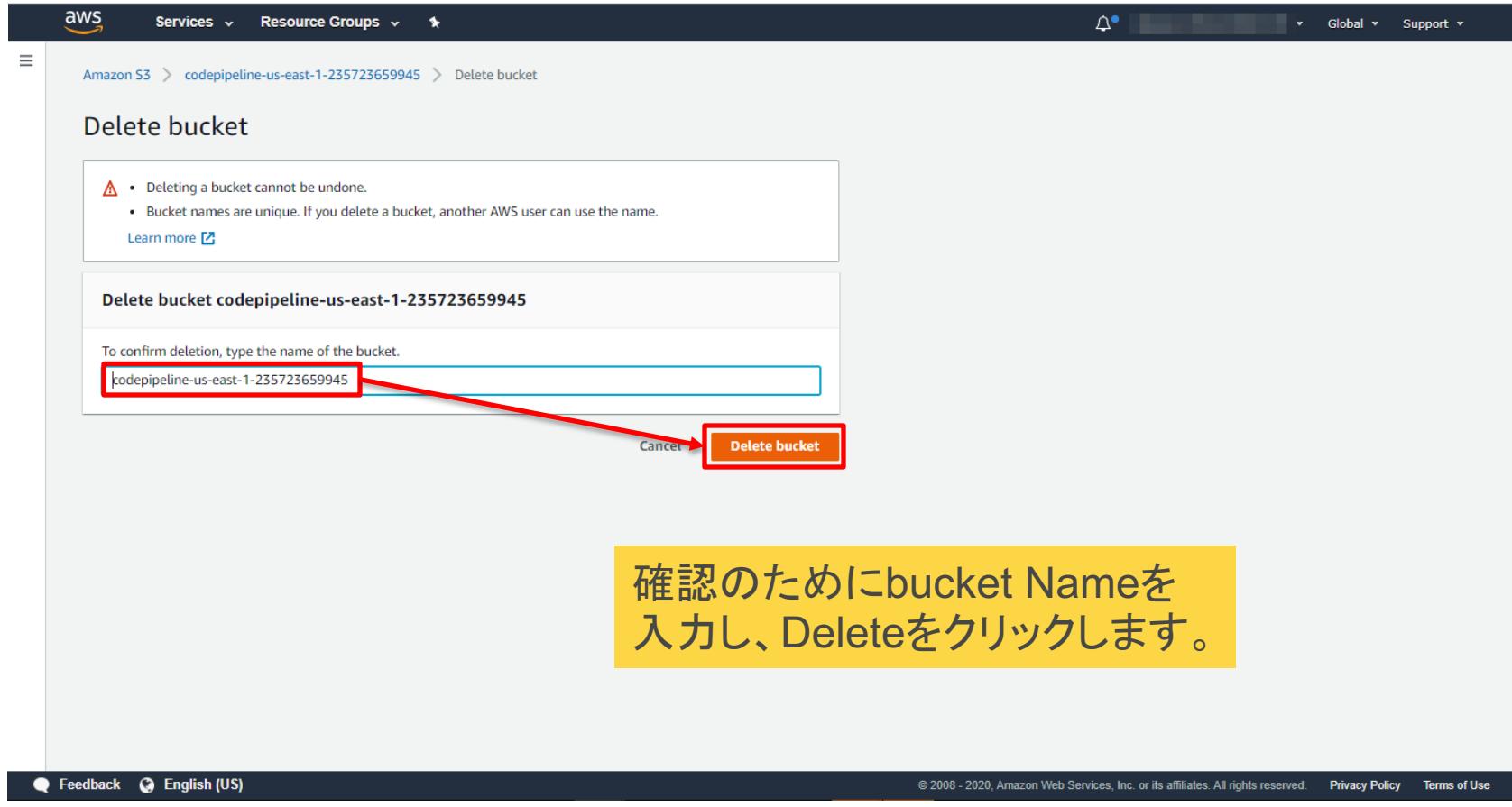
S3 bucket の削除

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with 'Amazon S3' selected. The main area displays a table titled 'Buckets (12)' with columns for Name, Region, Access, and Bucket created. A yellow callout box points to the first row, which has a blue circular selection icon and the name 'codepipeline-us-east-1-235723659945'. A red arrow points from this row to the 'Delete' button at the top right of the table. The 'Delete' button is highlighted with a red box. Other buttons in the header include 'Copy ARN', 'Empty', and 'Create bucket'. A search bar at the top says 'Find bucket by name'. The bottom of the page includes standard AWS footer links like 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and the AWS logo.

再度、自動作成したbucketを選択し、Deleteを選択します。



S3 bucket の削除



AWS Services Resource Groups Global Support

Amazon S3 > codepipeline-us-east-1-235723659945 > Delete bucket

Delete bucket

⚠ • Deleting a bucket cannot be undone.
• Bucket names are unique. If you delete a bucket, another AWS user can use the name.

[Learn more](#)

Delete bucket codepipeline-us-east-1-235723659945

To confirm deletion, type the name of the bucket.

 Cancelling Delete bucket

確認のためにbucket Nameを
入力し、Deleteをクリックします。

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

ECR の削除

The screenshot shows the AWS ECR service page. On the left sidebar, under 'Amazon Container Services', 'Amazon ECR' is selected, and 'Repositories' is highlighted with a red box and arrow. In the main content area, the 'Repositories' section displays one repository named 'php-sample'. A red box and arrow point to the blue circular checkbox next to the repository name. Another red box and arrow point to the 'Delete' button in the top right corner of the repository card. Below the table, a yellow callout box contains the Japanese text: '作成したRepositoryを選択し、Deleteを選択します。' (Select the created Repository and choose Delete). The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, 'N. Virginia' region, and 'Support' dropdown.

Repository name	URI	Created at	Tag immutability	Scan on push
php-sample	363654005279.dkr.ecr.us-east-1.amazonaws.com/php-sample	04/08/20, 03:08:58 AM	Disabled	Disabled

作成したRepositoryを選択し、Deleteを選択します。

ECS Task definitions の削除

The screenshot shows the AWS ECS Task Definitions page for a cluster named 'php-sample'. The left sidebar highlights 'Task Definitions'. A red box and arrow point to the 'Deregister' button in the 'Actions' dropdown menu for the selected task definition 'php-sample:1'. A yellow callout box at the bottom contains Japanese text and a bulleted list.

Task Definition Name : php-sample

Select a revision for more details

Last updated on April 9, 2020 6:06:04 PM (0m ago)

1-1 Page size 50

Task Definition Name: php-sample:1

Status: Active

Actions

- Run Task
- Create Service
- Update Service
- Deregister
- Edit tags

作成したTask定義を選択し、登録解除します。

- php-sample-fargate

ECS Clusters の削除

The screenshot shows the AWS ECS Clusters management interface. On the left, a sidebar menu has 'Clusters' selected, indicated by a red box and arrow. The main content area shows two clusters: 'Cluster : handson-cluster' and 'Cluster : fargate-cluster'. The 'handson-cluster' card displays its ARN, status (ACTIVE), and various task counts. The 'fargate-cluster' card is highlighted with a yellow box containing Japanese text and a bullet point. The top right of the main content area has 'Update Cluster' and 'Delete Cluster' buttons, with 'Delete Cluster' also highlighted by a red box and arrow.

Clusters

Clusters

handson-cluster

Cluster : handson-cluster

Get a detailed view of the resources on your cluster.

Cluster ARN arn:aws:ecs:us-east-1:363654005279:cluster/handson-cluster

Status ACTIVE

Registered container instances 1

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 0 EC2

Active service count 0 Fargate, 1 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks

Create Update

Filter in this page

作成したClusterを選択し、Deleteします。

- fargate-cluster

Service Name	Status	Service ty...	Task Defin...	Desired ta...	Running t...	Launch ty...	Platform v...
php-sample	ACTIVE	REPLICA	php-sample:1	0	0	EC2	--

ALB の削除

The screenshot shows the AWS CloudFormation console interface. On the left, a navigation sidebar lists various AWS services under categories like IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, LOAD BALANCING, and AUTO SCALING. The 'LOAD BALANCING' section is expanded, and the 'Load Balancers' item is highlighted with a red box. In the main content area, a table lists a single load balancer named 'php-sample'. A red box highlights the 'Delete' option in the context menu that appears when clicking on the row. A large yellow callout box contains the Japanese text: '作成したALBを選択し、Deleteを選択します。' (Select the created ALB and select Delete). The bottom of the screen shows the standard AWS footer with links for Feedback, English (US), Copyright notice (2008-2020), Privacy Policy, Terms of Use, and the AWS logo.

作成したALBを選択し、
Deleteを選択します。

New EC2 Experience
Tell us what you think

Services ▾ Resource Groups ▾

Create Load Balancer Actions ▾

Filter by tags and attributes

Name: php-sample

State: active

VPC ID: vpc-08878c6c13dd0c5ab

Availability Zones: us-east-1a, us-east-1b

Type: application

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

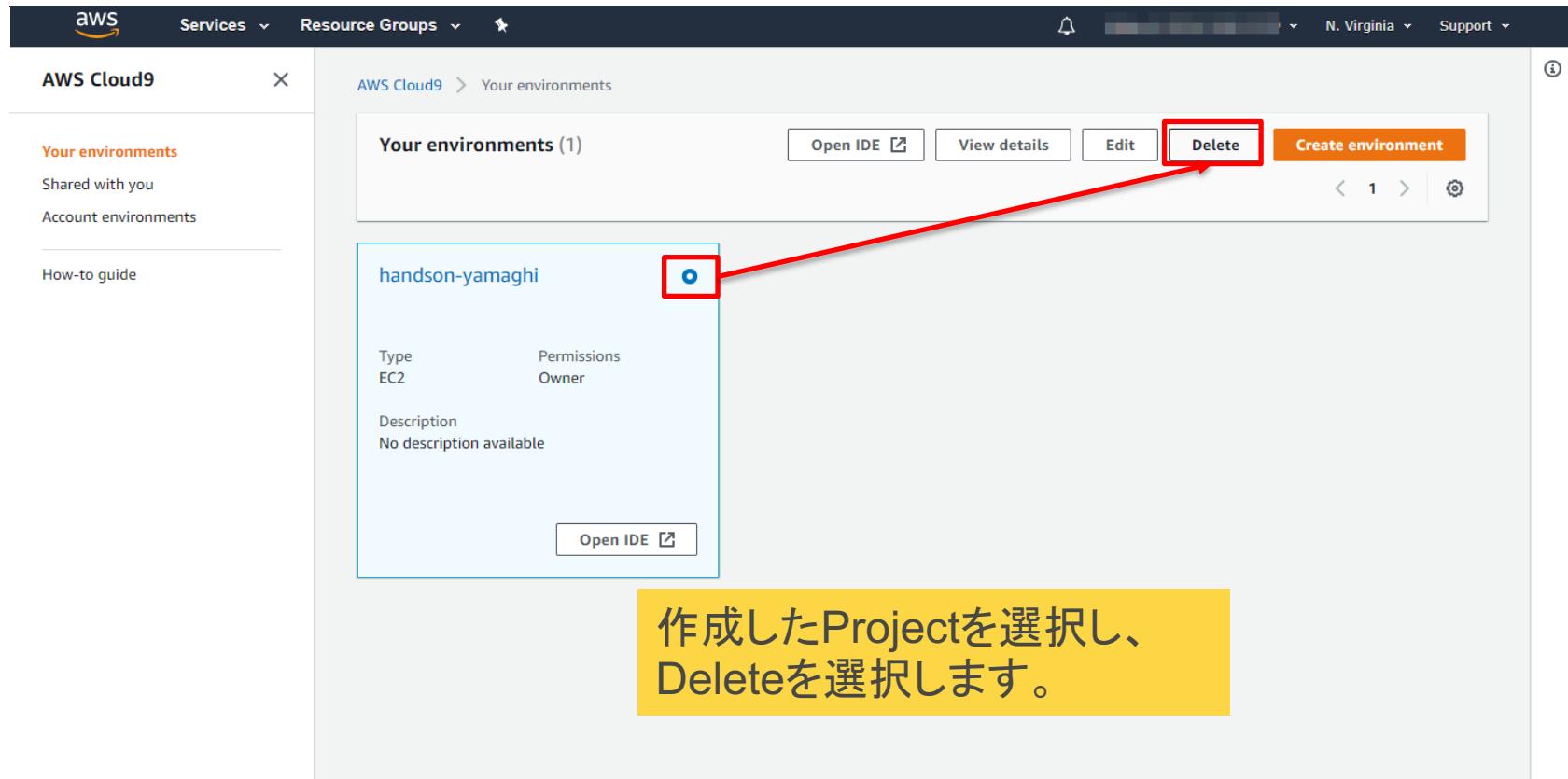
ALB Target Groups の削除

The screenshot shows the AWS Lambda console interface. On the left, a sidebar lists various services: EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing (with Load Balancers and Target Groups selected), and Auto Scaling. A red box highlights the 'Target Groups' link under the Load Balancing section. The main content area displays a table of target groups. Two target groups are listed: 'tg-fargat-php-sample-fargate-1' and 'tg-fargat-php-sample-fargate-2'. A red box surrounds the first target group. A context menu is open over the second target group, with 'Delete' highlighted by another red box. The menu also includes options like 'Edit health check', 'Register and deregister instance / ip targets', and 'Edit attributes'. Below the table, a message states 'Target groups: tg-fargat-php-sample-fargate-1, tg-fargat-php-sample-fargate-2'. At the bottom, tabs for 'Description', 'Targets' (which is active and highlighted in orange), 'Health checks', 'Monitoring', and 'Tags' are visible. A yellow callout box contains the Japanese text: '作成したTarget Groupsを選択し、Deleteを選択します。' (Select the created Target Groups and select Delete).

Name	Port	Protocol	Target type	Load Balancer	VPC ID	Monitoring
tg-fargat-php-sample-fargate-1	80	HTTP	ip	lb-003b4740bb2e466d	vpc-003b4740bb2e466d	
tg-fargat-php-sample-fargate-2	80	HTTP	ip	lb-003b4740bb2e466d	vpc-003b4740bb2e466d	



Cloud9 の削除



AWS Cloud9

Services ▾ Resource Groups ▾

N. Virginia Support

Your environments (1)

Open IDE View details Edit Delete Create environment

handson-yamaghi

Type EC2 Permissions Owner

Description No description available

Open IDE

作成したProjectを選択し、Deleteを選択します。

CloudFormation の削除

The screenshot shows the AWS CloudFormation service in the AWS Management Console. On the left, a navigation pane lists 'Stacks' (selected), 'Stack details', 'Drifts', 'StackSets', 'Exports', and 'Designer'. Below this is a section for 'CloudFormation registry' with 'Resource types'. At the bottom left is a 'Feedback' link. The main content area is titled 'CloudFormation > Stacks' and shows a table of stacks. The table has columns for 'Stack name', 'Status', 'Created time', and 'Description'. A single stack named 'vpc-for-hanson-cicd' is listed, showing a status of 'CREATE_COMPLETE' and a creation time of '2020-04-08 02:10:21 UTC+0900'. To the right of the table are buttons for 'Delete', 'Update', 'Stack actions', and 'Create stack'. A red box highlights the 'Delete' button, and another red box highlights the stack name 'vpc-for-hanson-cicd' in the table. A large yellow callout box in the bottom right corner contains the Japanese text: '作成したStackを選択し、Deleteを選択します。' (Select the created Stack and choose Delete). The top right of the screen shows the region 'N. Virginia' and a 'Support' link.

Stacks

Stack details

Drifts

StackSets

Exports

Designer

CloudFormation registry

Resource types

Feedback

CloudFormation > Stacks

Stacks (2)

Filter by stack name

Active View nested

< 1 >

Stack name	Status	Created time	Description
vpc-for-hanson-cicd	CREATE_COMPLETE	2020-04-08 02:10:21 UTC+0900	Template of ECS_CICD_Handso...

Delete

Update

Stack actions

Create stack

作成したStackを選択し、Deleteを選択します。

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

IAM Role の削除

The screenshot shows the AWS IAM service interface. On the left, a sidebar menu is open under the 'Access management' section, with 'Roles' selected. A red box highlights the 'Roles' link. In the main content area, a search bar at the top contains the text 'ecsServiceRole'. Below it, a table lists a single role: 'Role name: ecsServiceRole', 'Trusted entities: AWS service: ecs', and 'Last activity: Today'. A red box highlights the 'ecsServiceRole' entry, and another red box highlights the 'Delete role' button in the top right corner of the table header.

作成したRoleを選択し、Deleteを選択します。

- cwe-role-us-east-1-php-sample-pipeline
- AWSCodePipelineServiceRole-us-east-1-php-sample-pipeline
- CodeDeployRoleforECS
- codebuild-php-sample-build-service-role

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



CloudWatch Logs の削除

The screenshot shows the AWS CloudWatch Logs console. On the left sidebar, under the 'Logs' section, 'Log groups' is highlighted with a red box. A red arrow points from this box to a context menu that is open over a specific log group entry. This entry is also highlighted with a red box. The context menu contains the following options: 'Create log group', 'Delete log group' (which is highlighted with a red box), 'Export', 'Export data to Amazon S3', 'View all exports to Amazon S3', 'Subscriptions', 'Stream to AWS Lambda', 'Stream to Amazon Elasticsearch Service', and 'Remove Subscription Filter'. The main table on the right lists 10 log groups, each with columns for 'Log Groups', 'Expire Events After', 'Metric Filters', and 'Subscriptions'. The first log group listed is '/ecs/php-sample-fa'. A yellow callout box with Japanese text is overlaid on the bottom right of the screenshot.

出力されたLogを選択し、
Deleteを選択します。

Log Groups	Expire Events After	Metric Filters	Subscriptions
/ecs/php-sample-fa	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None
[redacted]	Never Expire	0 filters	None

Thank you!

