

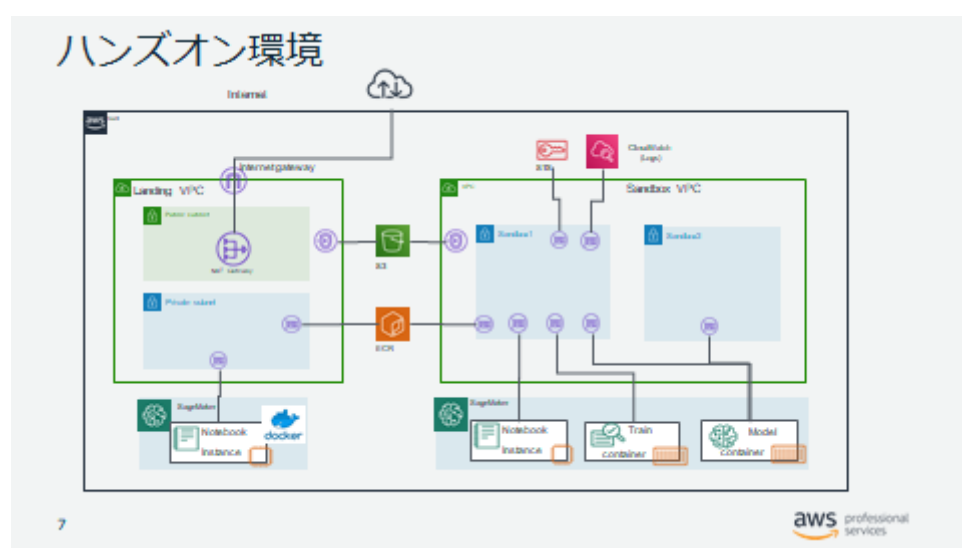
# SageMaker Sandbox handson

Handson

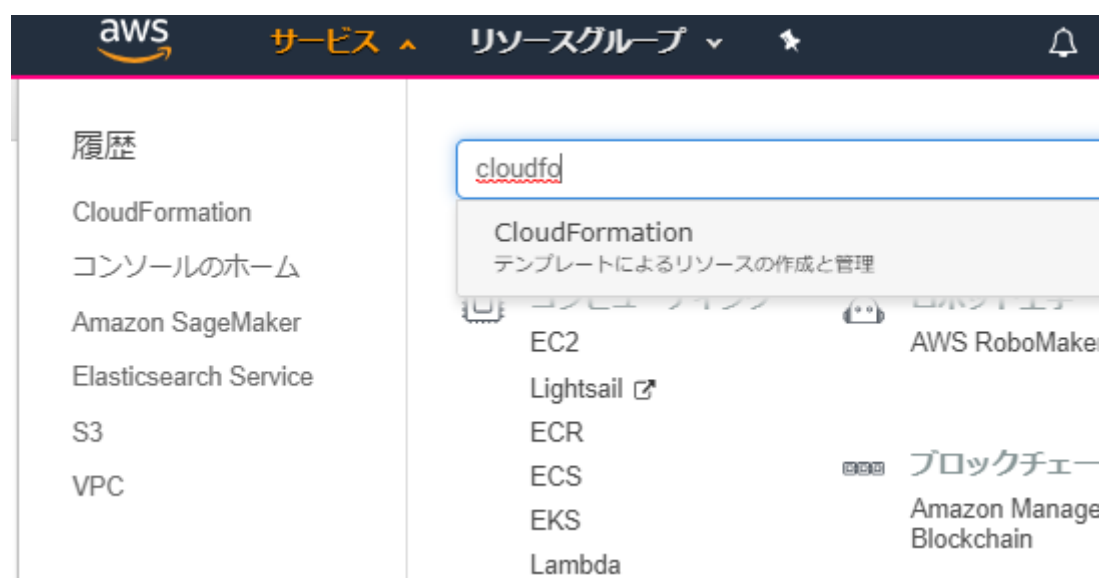
## Step1 CloudFormationによる環境作成

事前に配布した2つのYAMLファイルのCloudFormation テンプレートを実行します、

- sandbox-landing.yaml
- sandbox-sandbox.yaml



管理コンソールからCloudFormationのサービスを選択



Landingのスタック作成



## 1. 「新しいスタックの作成」を選択



## 2. 「Amazon S3 テンプレート URL の指定」で

<https://s3-ap-northeast-1.amazonaws.com/kh-handsondata/cfn/sandbox-landing.yaml> を指定し、「次へ」

## スタックの作成

### テンプレートの選択

詳細の指定  
オプション  
確認

### テンプレートの選択

作成したいスタックの内容が書かれたテンプレートを選択してください。スタックは、単一のユニットとして管理できる関連リソースのグループです。

テンプレートをデザインする

既存のテンプレートの作成や修正は、AWS CloudFormation Designer をご利用ください。[詳細はこちら](#)。

テンプレートのデザイン

### テンプレートの選択

テンプレートは、スタックのリソースとそのプロパティを説明する、JSON/YAML 形式のテキストファイルです。[詳細はこちら](#)。

● サンプルテンプレートの選択

● テンプレートを Amazon S3 にアップロード

ファイルを選択 選択されていません

● Amazon S3 テンプレート URL の指定

[デザイナーで表示](#)

キャンセル

次へ

### 3. スタックの名前に "WS1" を入れて「次へ」

IAM ロール

ロールを選択 (オプション)

ロールの ARN を入力します

### ▼ ロールバックトリガー

ロールバックトリガーを使用すると、スタックの作成および更新中にアプリケーションの状態を AWS CloudFormation でモニタリングし、指定したいいずれかのアラームのしきい値をアプリケーションが超過した場合に、そのオペレーションをロールバックできます。[詳細はこちら](#)

モニタリング時間 ⓘ

分

最小値は 0 です。最大値は 180 です

残りの利用可能なトリガー: 5

	タイプ	ARN (Amazon リソースネーム)	
1	AWS::CloudWatch::Alarm	<input type="text"/>	<a href="#">+</a>

### ▶ アドバンスド

通知オプションやスタックポリシーなど、スタックのオプションを追加設定することができます。[詳細情報](#)。

キャンセル

戻る

次へ

### 4. デフォルトのまま「次へ」



The following resource(s) require capabilities: [AWS::IAM::Role]

このテンプレートには、Identity and Access Management (IAM) リソースが含まれています。これらのリソースを個別に作成し、それぞれに最小限必要な権限を与えるかどうか確認してください。さらに、カスタム名が付けられているか確認してください。カスタム名が、ご利用の AWS アカウント内で一意のものであることを確認してください。[詳細はこちら](#)。を選択してください。

☒ **AWS CloudFormation** によってカスタム名のついた IAM リソースが作成される場合があることを承認します。

スタックのクイック作成 (これに類似したスタックを作成します。詳細の大部分は自動的に入力されます)

キャンセル

戻る

作成

5. 「AWS CloudFormation によってカスタム名のついた IAM リソースが作成される場合があることを承認します。」のチェックをオンにして、「作成」

CloudFormation ▼ スタック

スタックの作成 ▼ アクション ▼ テンプレートのデザイン

フィルター: アクティブ ▼ スタックの名前でフィルター 1 個のスタックを表示中

スタックの名前	作成時間	状況	ドリフトステータス	説明
<input checked="" type="checkbox"/> WS1	2019-08-07 21:03:16 UTC+0900	CREATE_IN_PROGRESS	NOT_CHECKED	Stack to create VPC and Network sett...

概要 出力 リソース イベント テンプレート パラメータ タグ スタックのポリシー 変更セット ロールバックトリガー

フィルター条件: ステータス ▼ 検索イベント

2019-08-07	状況	タイプ	論理 ID	状況の理由
▶ 21:03:16 UTC+0900	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	WS1	User Initiated

## Sandboxのスタック作成

CloudFormation ▼ スタック

スタックの作成 ▼ アクション ▼ テンプレートのデザイン

6. 「新しいスタックの作成」を選択し、「[Amazon S3 テンプレート URL の指定]」で <https://s3-ap-northeast-1.amazonaws.com/kh-handsondata/cfn/sandbox-sandbox.yaml> を指定する

## スタックの作成

## テンプレートの選択

詳細の指定  
オプション  
確認

## テンプレートの選択

作成したいスタックの内容が書かれたテンプレートを選択してください。スタックは、単一のユニットとして管理できる関連リソースのグループです。

テンプレートをデザインする

既存のテンプレートの作成や修正は、AWS CloudFormation Designer をご利用ください。詳細はこちら。

テンプレートのデザイン

## テンプレートの選択

テンプレートは、スタックのリソースとそのプロパティを説明する、JSON/YAML 形式のテキストファイルです。詳細はこちら。

● サンプルテンプレートの選択

● テンプレートを Amazon S3 にアップロード

ファイルを選択 選択されていません

● Amazon S3 テンプレート URL の指定

https://s3-ap-northeast-1.amazonaws.com/kh-handsor デザイナーで表示

キャンセル

次へ

## 7. スタック名を「WS2」にして同様に作成する

## スタックの作成

## テンプレートの選択

詳細の指定  
オプション  
確認

## 詳細の指定

スタック名とパラメータ値を指定します。AWS CloudFormation テンプレートに定義づけられるデフォルトのパラメータ値を使用、または変更することができます。詳細はこちら。

スタックの名前

WS2

## パラメータ

SageMakerInstanceType

ml.t3.xlarge

The type of SageMaker notebook to be provisioned.

キャンセル

戻る

次へ

8. 「AWS CloudFormation によってカスタム名のついた IAM リソースが作成される場合があることを承認します。」にチェックをオンにして、「作成」

**i** The following resource(s) require capabilities: [AWS::IAM::Role]

このテンプレートには、Identity and Access Management (IAM) リソースが含まれています。これらのリソースを個別に作成し、それぞれに最小限必要な権限を与えるかどうか確認してください。さらに、カスタム名が付けられているか確認してください。カスタム名が、ご利用の AWS アカウント内で一意のものであることを確認してください。[詳細はこちら](#)。を選択してください。

☒ AWS CloudFormation によってカスタム名のついた IAM リソースが作成される場合があることを承認します。

スタックのクイック作成 (これに類似したスタックを作成します。詳細の大部分は自動的に入力されます)

キャンセル

戻る

作成

## 9. スタックが CREATE\_COMPLETE になれば完成

CloudFormation ▼ スタック

スタックの作成 ▼ アクション ▼ テンプレートのデザイン

フィルター: アクティブ ▼ スタックの名前でフィルター

	スタックの名前	作成時間	状況	ドリフトステータス	説明
<input checked="" type="checkbox"/>	WS2	2019-08-07 22:55:13 UTC+0900	CREATE_COMPLETE	NOT_CHECKED	Stack to create VPC and Network settings
<input type="checkbox"/>	WS1	2019-08-07 21:03:16 UTC+0900	CREATE_COMPLETE	NOT_CHECKED	Stack to create VPC and Network settings

概要 出力 リソース イベント テンプレート パラメータ タグ スタックのポリシー 変更セット ロールバックトリガー

フィルター条件: ステータス ▼ 検索イベント

2019-08-07	状況	タイプ	論理 ID	状況の理由
▶ 22:59:06 UTC+0900	CREATE_COMPLETE	AWS::CloudFormation::Stack	WS2	
▶ 22:59:03 UTC+0900	CREATE_COMPLETE	AWS::SageMaker::NotebookInstance	NotebookInstance	
▶ 22:57:30 UTC+0900	CREATE_COMPLETE	AWS::EC2::VPCEndpoint	SageMakerNotebookEndpoint	
▶ 22:57:29 UTC+0900	CREATE_COMPLETE	AWS::EC2::VPCEndpoint	SageMakerApiEndpoint	
▶ 22:57:29 UTC+0900	CREATE_COMPLETE	AWS::EC2::VPCEndpoint	LogEndpoint	
▶ 22:57:29 UTC+0900	CREATE_COMPLETE	AWS::EC2::VPCEndpoint	SageMakerRuntimeEndpoint	
▶ 22:57:29 UTC+0900	CREATE_COMPLETE	AWS::EC2::VPCEndpoint	STSEndpoint	
▶ 22:57:28 UTC+0900	CREATE_COMPLETE	AWS::EC2::VPCEndpoint	ECRApiEndpoint	
▶ 22:56:12 UTC+0900	CREATE_COMPLETE	AWS::EC2::SubnetNetworkAclAssociation	PrivateSubnetNetworkAclAssociation1	
▶ 22:56:12 UTC+0900	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PrivateSubnetRouteTableAssociation1	
▶ 22:56:12 UTC+0900	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PrivateSubnetRouteTableAssociation2	
▶ 22:56:12 UTC+0900	CREATE_COMPLETE	AWS::EC2::SubnetNetworkAclAssociation	PrivateSubnetNetworkAclAssociation2	

## 確認

- WS1/WS2 の2つのVPCが作成されている

VPC ダッシュボード

VPC でフィルタリング: VPC の選択

Virtual Private Cloud

VPC

サブネット

ルートテーブル

インターネットゲートウェイ

VPC の作成 アクション ▼

タグや属性によるフィルター、またはキーワードによる検索

Name	VPC ID	状態	IPv4 CIDR	IPv6 CIDR	DHCP オプションセット	メインルートテーブル
<input type="checkbox"/> vpc-98133dff	vpc-98133dff	available	172.31.0.0/16	-	dopt-5a85d13d	rtb-8240c7e4
<input type="checkbox"/> WS1-LandingVPC	vpc-07b68e0f5447b7ee8	available	10.1.0.0/16	-	dopt-5a85d13d	rtb-0be4c3cf160c1bb16
<input type="checkbox"/> WS2-SandboxVPC	vpc-0b6ab2fd074a9b13d	available	10.0.0.0/16	-	dopt-5a85d13d	rtb-01d57d84c7a6a155e

- WS1/WS2 の4つのサブネットが作成されている

VPC ダッシュボード

VPC でフィルタリング:

Q VPC の選択

Virtual Private Cloud

VPC

サブネット

ルートテーブル

インターネットゲートウェイ

Egress Only インターネットゲートウェイ

サブネットの作成 アクション

タグや属性によるフィルター、またはキーワードによる検索

Name	サブネット ID	状態	VPC	IPv4 CIDR	利用可能な IPv4
WS1-LandingPublic	subnet-08fdec1037144708	available	vpc-07b68e0f5447b7ee8   WS1-LandingVPC	10.1.1.0/24	250
WS1-LandingPrivate	subnet-02b3ac04d3a275af7	available	vpc-07b68e0f5447b7ee8   WS1-LandingVPC	10.1.3.0/24	248
WS2-Sandbox1	subnet-02a2d2c201dd7d9d4	available	vpc-0b6ab2fd074a9b13d   WS2-SandboxVPC	10.0.2.0/24	244
WS2-Sandbox2	subnet-05bc55bb94482a9e1	available	vpc-0b6ab2fd074a9b13d   WS2-SandboxVPC	10.0.4.0/24	245
	subnet-57e1bb0c	available	vpc-98133dff	172.31.0.0/20	4091
	subnet-45aa410d	available	vpc-98133dff	172.31.32.0/20	4091
	subnet-7d4fbc56	available	vpc-98133dff	172.31.16.0/20	4091

- endpoint インターフェースが作成されている

VPC ダッシュボード

VPC でフィルタリング:

Q VPC の選択

Virtual Private Cloud

VPC

サブネット

ルートテーブル

インターネットゲートウェイ

Egress Only インターネットゲートウェイ

DHCP オプションセット

Elastic IP

エンドポイント

エンドポイントのサービス

エンドポイントの作成 アクション

タグや属性によるフィルター、またはキーワードによる検索

Name	エンドポイント ID	VPC ID	サービス名	エンドポイントタイプ	ステータス
	vpce-008d1b775f0f66bf2	vpc-07b68e0f5447b7ee8   WS1-LandingV...	com.amazonaws.ap-northeast-1.s3	Gateway	使用可能
	vpce-053f598199715c83b	vpc-07b68e0f5447b7ee8   WS1-LandingV...	com.amazonaws.ap-northeast-1.ecr.api	Interface	使用可能
	vpce-08b2095882d808718	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.s3	Gateway	使用可能
	vpce-0b74491368185e056	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sagemaker.runtime	Interface	使用可能
	vpce-0c914875df0f8bec5	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sagemaker.api	Interface	使用可能
	vpce-0cbf0917a1c84b793	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	aws.sagemaker.ap-northeast-1.notebook	Interface	使用可能
	vpce-0e237cf2091a87953	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.ecr.api	Interface	使用可能
	vpce-0e4f6f22b0a2abe2e	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sts	Interface	使用可能
	vpce-0ef953a108d314ef4	vpc-07b68e0f5447b7ee8   WS1-LandingV...	aws.sagemaker.ap-northeast-1.notebook	Interface	使用可能
	vpce-0f25a98b71ebd4d4d	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.logs	Interface	使用可能

- 2つの Notebook インスタンスが作成されている

Amazon SageMaker

ダッシュボード

検索

Ground Truth

ラベリングジョブ

データセットのラベリング

ラベリング労働力

ノートブック

ノートブックインスタンス

ライフサイクル設定

Git リポジトリ

Amazon SageMaker > ノートブックインスタンス

ノートブックインスタンス

アクション

Create notebook instance

Q ノートブックインスタンスを検索

名前	インスタンス	作成時刻	ステータス	アクション
WS2-SandboxNotebook	ml.t3.xlarge	Aug 07, 2019 13:55 UTC	InService	Open Jupyter   Open JupyterLab
WS1-LandingNotebook	ml.t3.medium	Aug 07, 2019 12:04 UTC	InService	Open Jupyter   Open JupyterLab

## Step2 カスタムコンテナ作成

Amazon SageMaker > ノートブックインスタンス

- WS1-LandingNotebook インスタンスで Open Jupyter をクリック

ノートブックインスタンス

アクション

Create notebook instance

Q ノートブックインスタンスを検索

名前	インスタンス	作成時刻	ステータス	アクション
WS2-SandboxNotebook	ml.t3.xlarge	Aug 07, 2019 13:55 UTC	InService	Open Jupyter   Open JupyterLab
WS1-LandingNotebook	ml.t3.medium	Aug 07, 2019 12:04 UTC	InService	Open Jupyter   Open JupyterLab

- /khlab-handson/scikit\_custom/ create\_docker.ipynb をオープン

1		khlab-handson / scikit_custom		Name	Last Modified	File size
	..				数秒前	
<input type="checkbox"/>	container				16時間前	
<input type="checkbox"/>	data				16時間前	
<input checked="" type="checkbox"/>	create_docker.ipynb				16時間前	52.5 kB
<input type="checkbox"/>	scikit_learn.ipynb				16時間前	27.8 kB
<input type="checkbox"/>	stack.png				16時間前	13.6 kB

- jupyter create\_docker Last Checkpoint: 14分前 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted | conda\_python3

Custom Container Hands-on

インターネットにつながる環境でコンテナを作り、インターネットにつながらないSandbox環境でモデル開発をする方法のハンズオンをします。

本ハンズオンでは scikit-learn を使い 決定木 (decision tree) のアルゴリズムを例にします。

  - [scikit-learn](#)
  - [decision tree](#)

## Part 1: SageMakerのNotebookを使って学習コンテナの作成と登録

LandingVPCインターネットGatewayを有し、NATを使ってインターネット上のライブラリーをダウンロードすることができます。Sandbox環境ではインターネットにアクセスできないため、このLandingVPC環境で学習に必要なライブラリーを備えたdockerコンテナを作り、ECRに登録します。

### SageMakerによるDocker container の実行

学習と推論は同じコンテナイメージを使います。Amazon SageMaker はコンテナに `train` と `serve` のコマンドを渡して、学習と推論を使い分けています。真的には次の 2 つのコマンドを使用して、イメージを実行します。

```
docker run <イメージ> train
```

```
docker run <イメージ> serve
```

### 学習コンテナについて

学習コンテナは `/opt/ml` 以下のつぎのディレクトリ構成にあるファイルを利用します。

```
/opt/ml
├── input
│   ├── config
│   │   ├── hyperparameters.json
│   │   └── resourceConfig.json
│   └── data
│       ├── <channel_name>
│       └── <input data>
└── model
    └── <model files>
```

- docker push が成功すればOK



```
# Build the docker image locally with the image name and then push it to ECR
# with the full name.

docker build -t ${algorithm_name} .
docker tag ${algorithm_name} ${fullname}

docker push ${fullname}
e79142719515: Preparing
aeda103e78c9: Preparing
2558e637fbff: Preparing
f749b9b0fb21: Preparing
2558e637fbff: Waiting
f749b9b0fb21: Waiting
3824abd4897b: Pushed
e79142719515: Pushed
aeda103e78c9: Pushed
2558e637fbff: Pushed
fbbccd7bbcb2: Pushed
f749b9b0fb21: Pushed
d44c7b5b3cd3: Pushed
latest: digest: sha256:589373ffa353082317342efb15f2f3ddbe57ff76a8c4c96203a3fbf035e11b23 size: 1782
```

- Notebookを最後まで実行し、S3に関連ファイルをアップロードする

### 関連ファイルをS3にアップロード

Sandbox環境にデータのnotebookを送るため、S3にアップロードします。

```
In [6]: import sagemaker as sage

prefix = 'LAB-handson'
SRC_DIRECTORY = '../scikit_custom'

sess = sage.Session()

src_location = sess.upload_data(SRC_DIRECTORY, key_prefix=prefix)

In [7]: print(src_location)

s3://sagemaker-ap-northeast-1-925889618331/LAB-handson
```

このS3のパスをつぎのステップで使います。 パスをメモ帳などにコピー&ペストしておいてください。

## 確認

- ECRにイメージが登録されていることを確かめる



ECR > リポジトリ

Amazon Container Services

Amazon ECS

Clusters

Task definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

ECR > リポジトリ

リポジトリ (1)

リポジトリの検索

リポジトリの作成

作成時刻

sagemaker-decision-trees

925889618331.dkr.ecr.ap-northeast-1.amazonaws.com/sagemaker-decision-trees

19/08/12 16:12:59

ECR > リポジトリ > sagemaker-decision-trees

sagemaker-decision-trees

イメージ (1)

イメージの検索

Delete

イメージタグ

イメージ URI

プッシュされた日時

ダイジェスト

サイズ (MB)

latest

925889618331.dkr.ecr.ap-northeast-1.amazonaws.com/sagemaker-decision-trees:latest

19/08/12 16:14:06

sha256:3b62504d4...

146.04

## Step3 Sandbox環境でモデル学習と推論デプロイ

Amazon SageMaker > ノートブックインスタンス

WS2-SandboxNotebook インスタンスで Open Jupyter をクリック

ノートブックインスタンス

Actions

Create notebook instance

ノートブックインスタンスを検索

名前

インスタンス

作成時刻

ステータス

アクション

WS2-SandboxNotebook

ml.t3.xlarge

Aug 07, 2019 13:55 UTC

InService

Open Jupyter | Open JupyterLab

WS1-LandingNotebook

ml.t3.medium

Aug 07, 2019 12:04 UTC

InService

Open Jupyter | Open JupyterLab

## 準備作業

- Jupyter からTerminalをオープン

Select items to perform actions on them.

Upload

New

0 /

The notebook list is empty.

Notebook:

R (Beta)

Sparkmagic (PySpark)

Sparkmagic (Spark)

Sparkmagic (SparkR)

conda\_amazonei\_mxnet\_p27

conda\_amazonei\_mxnet\_p36

conda\_amazonei\_tensorflow\_p27

conda\_amazonei\_tensorflow\_p36

conda\_chainer\_p27

conda\_chainer\_p36

conda\_mxnet\_p27

conda\_mxnet\_p36

conda\_python2

conda\_python3

conda\_pytorch\_p27

conda\_pytorch\_p36

conda\_tensorflow\_p27

conda\_tensorflow\_p36

Other:

Text File

Folder

Terminal

s://ws2-sandboxnotebook.notebook.ap-northeast-1.sagemaker.aws/tree#

- S3からnotebook関連ファイルをコピー

Files &gt; New &gt; Terminal

Terminal で次のコマンドを実行します。

以下の**S3のパス**は、Step 2 でアップロードしたパスに置き換えてください。

1. `cd SageMaker`
2. `aws s3 cp --recursive s3://sagemaker-ap-northeast-1-<アカウントID>/LAB-handson scikit-custom`

## endpoints.jsonを修正

```
/home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/botocore/data/endpoints.json
```

の hostname に VPC endpoint のホスト名を追記していく

- VPC > エンドポイント  
WS2でフィルターをかけて、WS2-SandboxVPCのエンドポイントのみを表示する

VPC ダッシュボード

VPC でフィルタリング:  フィルターの追加

Virtual Private Cloud

VPC

サブネット

ルートテーブル

インターネットゲートウェイ

Egress Only インターネットゲートウェイ

DHCP オプションセット

Elastic IP

エンドポイント

エンドポイントのサービス

エンドポイントの作成 アクション

Name	エンドポイント ID	VPC ID	サービス名	エンドポイント
<input checked="" type="checkbox"/>	vpce-00d752cd93019bd37	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.sts	Interface
<input type="checkbox"/>	vpce-028d7223008a8a351	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.ecr.api	Interface
<input type="checkbox"/>	vpce-05876bc4d5a453423	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.sagemaker.runtime	Interface
<input type="checkbox"/>	vpce-05ecabe40e215d758	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.sagemaker.api	Interface
<input type="checkbox"/>	vpce-07056cfaeda83998	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.logs	Interface
<input type="checkbox"/>	vpce-07e1055f4fb1e89ab	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.s3	Gateway
<input type="checkbox"/>	vpce-089b1b5779cb5ca1c	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.codecommit	Interface
<input type="checkbox"/>	vpce-09c097df18f6cc21a	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	aws.sagemaker.ap-northeast-1.notebook	Interface
<input type="checkbox"/>	vpce-09c46259b09ff3431	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.ssm	Interface
<input type="checkbox"/>	vpce-0de6f424d76a78ed5	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.git-codecommit	Interface
<input type="checkbox"/>	vpce-0decc11fc98a41e23	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.ssmmessages	Interface

エンドポイント: vpce-00d752cd93019bd37

- エンドポイントを選択して詳細を表示する

VPC ダッシュボード

VPC でフィルタリング:  フィルターの追加

Virtual Private Cloud

VPC

サブネット

ルートテーブル

インターネットゲートウェイ

Egress Only インターネットゲートウェイ

DHCP オプションセット

Elastic IP

エンドポイント

エンドポイントのサービス

NAT ゲートウェイ

ピアリング接続

エンドポイントの作成 アクション

Name	エンドポイント ID	VPC ID	サービス名	エンドポイント
<input checked="" type="checkbox"/>	vpce-00d752cd93019bd37	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.sts	Interface
<input type="checkbox"/>	vpce-028d7223008a8a351	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.ecr.api	Interface
<input type="checkbox"/>	vpce-05876bc4d5a453423	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.sagemaker.runtime	Interface
<input type="checkbox"/>	vpce-05ecabe40e215d758	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.sagemaker.api	Interface
<input type="checkbox"/>	vpce-07056cfaeda83998	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.logs	Interface
<input type="checkbox"/>	vpce-07e1055f4fb1e89ab	vpc-0a17f97bb5623f7e5   WS2-SandboxVPC	com.amazonaws.ap-northeast-1.s3	Gateway

エンドポイント: vpce-00d752cd93019bd37

詳細 サブネット セキュリティグループ ポリシー 通知 タグ

エンドポイント ID vpce-00d752cd93019bd37 VPC ID vpc-0a17f97bb5623f7e5 | WS2-SandboxVPC

ステータス 使用可能 作成時刻 2019年8月21日 20:26:25 UTC+9

サービス名 com.amazonaws.ap-northeast-1.sts エンドポイントタイプ Interface

DNS 名 vpce-00d752cd93019bd37-4h94mb9s.sts.ap-northeast-1.vpc.amazonaws.com (Z2E726K9Y6RL4W)  
vpce-00d752cd93019bd37-4h94mb9s-ap-northeast-1a.sts.ap-northeast-1.vpc.amazonaws.com (Z2E726K9Y6RL4W)  
vpce-00d752cd93019bd37-4h94mb9s-ap-northeast-1c.sts.ap-northeast-1.vpc.amazonaws.com (Z2E726K9Y6RL4W)

- エンドポイントの詳細にあるDNS名を1つ選んで、endpoints.jsonの該当サービスのhostnameとして登録していく

エンドポイント: vpce-00d752cd93019bd37

詳細 サブネット セキュリティグループ ポリシー

エンドポイント ID vpce-00d752cd93019bd37

ステータス 使用可能

サービス名 com.amazonaws.ap-northeast-1.sts

DNS 名 vpce-00d752cd93019bd37-4h94mb9s.sts.ap-northeast-1.vpc.amazonaws.com (Z2E726K9Y6RL4W)  
vpce-00d752cd93019bd37-4h94mb9s-ap-northeast-1a.sts.ap-northeast-1.vpc.amazonaws.com (Z2E726K9Y6RL4W)  
vpce-00d752cd93019bd37-4h94mb9s-ap-northeast-1c.sts.ap-northeast-1.vpc.amazonaws.com (Z2E726K9Y6RL4W)

- 登録するエンドポイント
  - sts
  - logs

- notebook
- api.sagemaker
- runtime.sagemaker
- api.ecr
- endpoints.jsonにhostname登録の例  
 /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages/botocore/data/endpoints.json  
 エンドポイントのDNS名(vpceで始まる)を **hostname** として 追加した例です。

```

1.     "sts" : {
2.         "defaults" : {
3.             "credentialScope" : {
4.                 "region" : "ap-northeast-1"
5.             },
6.             "hostname" : "vpce-07a04bf233a7bbccb-yxh1tfbr.sts.ap-northeast-1.vpce.amazonaws.com"
7.         },
8.         "endpoints" : {
9.             "ap-east-1" : {
10.                 "credentialScope" : {
11.                     "region" : "ap-east-1"
12.                 },
13.                 "hostname" : "sts.ap-east-1.amazonaws.com"
14.             },
15.             "ap-northeast-1" : {
16.                 "credentialScope" : {
17.                     "region" : "ap-northeast-1"
18.                 },
19.                 "hostname" : "vpce-07a04bf233a7bbccb-yxh1tfbr.sts.ap-northeast-1.vpce.amazonaws.com"
20.             },

```

```

1.     "api.sagemaker" : {
2.         "endpoints" : {
3.             "ap-northeast-1" : {
4.                 "hostname" : "vpce-0a8e80e89e089bef4-0oxjfya5.api.sagemaker.ap-northeast-1.vpce.amazonaws.com"
5.             },

```

```

1.     "logs" : {
2.         "endpoints" : {
3.             "ap-east-1" : { },
4.             "ap-northeast-1" : {
5.                 "hostname" : "vpce-07b1d8b6b579c9244-9xycdsvm.logs.ap-northeast-1.vpce.amazonaws.com"
6.             },

```

```

1.         "runtime.sagemaker" : {
2.             "endpoints" : {
3.                 "ap-northeast-1" : {
4.                     "hostname" : "vpce-0b91622fd0b7b989a-xi93pg1w.runtime.sagem
aker.ap-northeast-1.vpce.amazonaws.com"
5.                 }
6.             }
7.         },

```


```

1.         "api.ecr" : {
2.             "endpoints" : {
3.                 "ap-east-1" : {
4.                     "credentialScope" : {
5.                         "region" : "ap-east-1"
6.                     },
7.                     "hostname" : "api.ecr.ap-east-1.amazonaws.com"
8.                 },
9.                 "ap-northeast-1" : {
10.                     "credentialScope" : {
11.                         "region" : "ap-northeast-1"
12.                     },
13.                     "hostname" : "vpce-0a1d76bfb01161444-m5pyrmpt.api.ecr.ap-no
rtheast-1.vpce.amazonaws.com "
14.                 },

```

## モデル学習の実行

- / scikit-custom / scikit-learn.ipynb を開く

 jupyter
 

Open JupyterLab
 Quit

Files
 Running
 Clusters
 Conda
 SageMaker Examples

Select items to perform actions on them.
 Upload
 New
 Refresh

0 / scikit-custom		Name	Last Modified	File size
	..		数秒前	
	container		31分前	
	data		31分前	
	create_docker.ipynb		39分前	53.6 kB
	scikit_learn.ipynb		39分前	25.7 kB
	stack.png		39分前	13.6 kB

- notebookに従って、一行ずつに実行  
Shift + Enter でカーソル行を実行します。

scikit\_learn

ws1-landingnotebook.notebook.ap-northeast-1.sagemaker.aws/notebooks/github/sagam...

jupyter scikit\_learn Last Checkpoint: 1時間前 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted conda\_python3

Run

Markdown

## カスタムコンテナハンズオン part2

### Part 2: カスタムコンテナを使って Amazon SageMakerで学習と推論をする

#### 環境セットアップ

S3バケットとロールを準備します。

```
In [5]: # S3 prefix
prefix = 'LAB-scikit-iris'

# Define IAM role
import boto3
import re

import os
import numpy as np
import pandas as pd
from sagemaker import get_execution_role

role = get_execution_role()
```

#### SageMakerのセッションを取得

SageMakerを操作するためのセッションを取得します。

```
In [6]: import sagemaker as sage
from time import gmtime, strftime

sess = sage.Session()
```

#### 学習データをS3にアップロード