

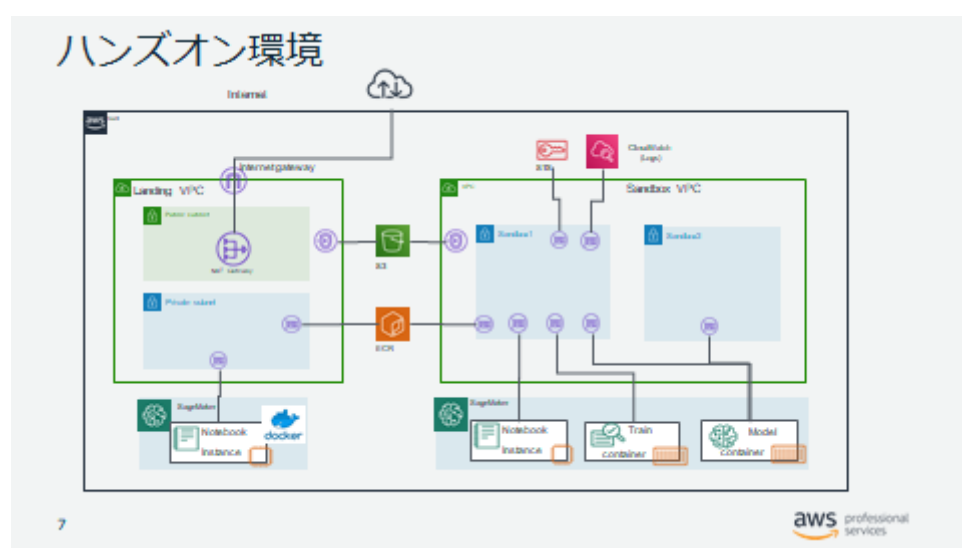
# SageMaker Sandbox handson

Handson

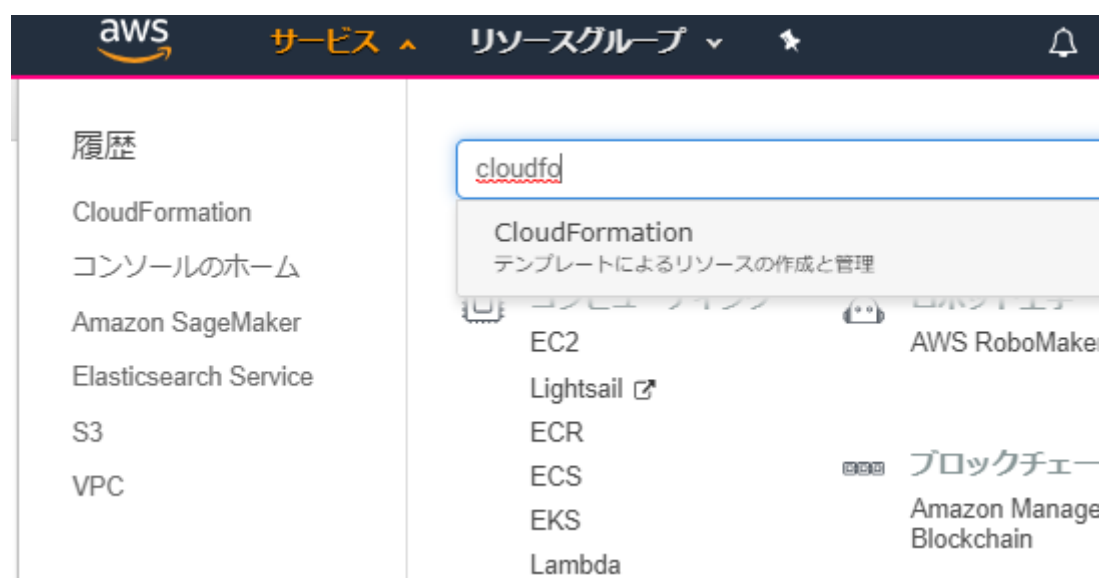
## Step1 CloudFormationによる環境作成

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. 「テンプレートを Amazon S3 にアップロード」のファイル選択で `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

## 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

WS1

## パラメータ

SageMakerInstanceType

ml.t3.medium

The type of SageMaker notebook to be provisioned.

SageMakerS3Bucket

&lt;Optional&gt; Name of a pre-existing bucket that SageMaker will be granted full access

キャンセル

戻る

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 ⓘ

0-180

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		+

## ▶ 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」

**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

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

C

⚙

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

## 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
- Amazon S3 テンプレート URL の指定

キャンセル 次へ

6 「新しいスタックの作成」を選択し、「テンプレートを Amazon S3 にアップロード」のファイル選択で `sandbox-sandbox.yaml` を指定する

CloudFormation ▾ スタック > スタックの作成

## 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

### パラメータ

SageMakerInstanceType  The type of SageMaker notebook to be provisioned.

SageMakerS3Bucket  <Optional> Name of a pre-existing bucket that SageMaker will be granted full access

キャンセル 戻る 次へ

7. スタック名を「WS2」にして同様に作成する

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									
▶ 22:59:06 UTC+0900	CREATE_COMPLETE	AWS::CloudFormation::Stack	WS2	状況の理由					
▶ 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	PrivateSubnetNetworkAclAssociation						

## 8. スタックが CREATE\_COMPLETE になれば完成

### 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: <input type="text" value="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						
	<input type="checkbox"/>	Name	VPC ID	状態	IPv4 CIDR	IPv6 CIDR	DHCP オプションセット
	<input type="checkbox"/>		vpc-98133dff	available	172.31.0.0/16	-	dopt-5a85d13d
	<input type="checkbox"/>	WS1-LandingVPC	vpc-07b68e0f5447b7ee8	available	10.1.0.0/16	-	dopt-5a85d13d
		WS2-SandboxVPC	vpc-0b6ab2fd074a9b13d	available	10.0.0.0/16	-	dopt-5a85d13d
		メインルートテーブル					
							rtb-8240c7e4
							rtb-0be4c3cf160c1bb16
							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: <input type="text" value="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						
	<input type="checkbox"/>	Name	サブネット ID	状態	VPC	IPv4 CIDR	利用可能な IPv4
	<input type="checkbox"/>	WS1-LandingPublic	subnet-08fdecd1037144708	available	vpc-07b68e0f5447b7ee8   WS1-LandingVPC	10.1.1.0/24	250
	<input type="checkbox"/>	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

[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
<input type="checkbox"/>	vpce-008d1b775f0f66bf2	vpc-07b68e0f5447b7ee8   WS1-LandingV...	com.amazonaws.ap-northeast-1.s3	Gateway	使用可能
<input type="checkbox"/>	vpce-053f598199715c83b	vpc-07b68e0f5447b7ee8   WS1-LandingV...	com.amazonaws.ap-northeast-1.ecr.api	Interface	使用可能
<input type="checkbox"/>	vpce-08b2095882d808718	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.s3	Gateway	使用可能
<input type="checkbox"/>	vpce-0b74491368185e056	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sagemaker.runtime	Interface	使用可能
<input type="checkbox"/>	vpce-0c914875df0f8bec5	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sagemaker.api	Interface	使用可能
<input type="checkbox"/>	vpce-0cbf0917a1c84b793	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	aws.sagemaker.ap-northeast-1.notebook	Interface	使用可能
<input type="checkbox"/>	vpce-0e237cf2091a87953	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.ecr.api	Interface	使用可能
<input type="checkbox"/>	vpce-0e4f6f22b0a2abe2e	vpc-0b6ab2fd074a9b13d   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sts	Interface	使用可能
<input type="checkbox"/>	vpce-0ef953a108d314ef4	vpc-07b68e0f5447b7ee8   WS1-LandingV...	aws.sagemaker.ap-northeast-1.notebook	Interface	使用可能
<input type="checkbox"/>	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 > ノートブックインスタンス

ノートブックインスタンス

検索

1

名前	インスタンス	作成時刻	ステータス	アクション
<input type="radio"/> WS2-SandboxNotebook	ml.t3.xlarge	Aug 07, 2019 13:55 UTC	InService	<a href="#">Open Jupyter</a>   <a href="#">Open JupyterLab</a>
<input type="radio"/> WS1-LandingNotebook	ml.t3.medium	Aug 07, 2019 12:04 UTC	InService	<a href="#">Open Jupyter</a>   <a href="#">Open JupyterLab</a>

## 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

検索

1

名前	インスタンス	作成時刻	ステータス	アクション
<input type="radio"/> WS2-SandboxNotebook	ml.t3.xlarge	Aug 07, 2019 13:55 UTC	InService	<a href="#">Open Jupyter</a>   <a href="#">Open JupyterLab</a>
<input type="radio"/> WS1-LandingNotebook	ml.t3.medium	Aug 07, 2019 12:04 UTC	InService	<a href="#">Open Jupyter</a>   <a href="#">Open JupyterLab</a>

- `create_docker.ipynb` をオープン

Files

Running

Clusters

SageMaker Examples

Conda

Duplicate

Rename

Move

Download

View

Edit

Upload

New

1

/ khlab-handson / scikit\_custom

Name

Last Modified

File size

..

数秒前

container

16時間前

data

16時間前

☒ create\_docker.ipynb

16時間前

52.5 kB

☐ scikit\_learn.ipynb

16時間前

27.8 kB

☐ stack.png

16時間前

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。

jupyter create\_docker Last Checkpoint: 14分前 (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

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。 具体的には次の 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
```

## 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

リポジトリ名

URI

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

ノートブックインスタンスを検索

Actions

Create notebook instance

作成時刻

ステータス

アクション

名前

インスタンス

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.

☐ 0 ☐ /

<input type="checkbox"/>	github
<input type="checkbox"/>	khlab-handson
<input type="checkbox"/>	sandbox

Upload

New

Refresh

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

- S3からnotebook関連ファイルをコピー  
Files > New > Terminal  
Terminal で次のコマンドを実行します

```
cd SageMaker
```

```
aws s3 cp --recursive s3://sagemaker-ap-northeast-1-925889618331/LAB-handson scikit-custom
```

## endpoints.jsonを修正

/home/ec2-user/anaconda3/envs/chainer\_p36/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

VPC ダッシュボード  
 VPC でフィルタリング:

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

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

<input type="checkbox"/>	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
<input type="checkbox"/>		vpce-035b81951dc7ef497	vpce-0ce421477b77ba823   WS2-Sandbox...	aws.sagemaker.ap-northeast-1.notebook	Interface
<input type="checkbox"/>		vpce-06799cf185df5570	vpce-048edf8487f1fe11e   WS1-LandingVPC	com.amazonaws.ap-northeast-1.s3	Gateway
<input type="checkbox"/>		vpce-07a04bf233a7bbccb	vpce-0ce421477b77ba823   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sts	Interface
<input type="checkbox"/>		vpce-07b1d8b6b579c9244	vpce-0ce421477b77ba823   WS2-Sandbox...	com.amazonaws.ap-northeast-1.logs	Interface
<input type="checkbox"/>		vpce-07d42c7c3adb6a940	vpce-0ce421477b77ba823   WS2-Sandbox...	com.amazonaws.ap-northeast-1.s3	Gateway
<input type="checkbox"/>		vpce-0935ba19263a7cc43	vpce-048edf8487f1fe11e   WS1-LandingVPC	com.amazonaws.ap-northeast-1.sagemaker.api	Interface
<input type="checkbox"/>		vpce-0a1d76bfb01161444	vpce-0ce421477b77ba823   WS2-Sandbox...	com.amazonaws.ap-northeast-1.ecr.api	Interface
<input type="checkbox"/>		vpce-0a8e80e89e089bef4	vpce-0ce421477b77ba823   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sagemaker.api	Interface
<input type="checkbox"/>		vpce-0b91622fd0b7b989a	vpce-0ce421477b77ba823   WS2-Sandbox...	com.amazonaws.ap-northeast-1.sagemaker.runtime	Interface
<input type="checkbox"/>		vpce-0d52bee6ec6c87ab2	vpce-048edf8487f1fe11e   WS1-LandingVPC	com.amazonaws.ap-northeast-1.logs	Interface

- 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-07a48d414059bafb4

詳細	サブネット	セキュリティグループ	通知
<p>エンドポイント ID    vpce-07a48d414059bafb4</p> <p>ステータス    使用可能</p> <p>サービス名    com.amazonaws.ap-northeast-1.ecr.api</p> <p>DNS 名    vpce-07a48d414059bafb4-0tiznt17.api.ecr.ap-northeast-1.vpce.amazonaws.com (Z2E726K9Y6RL4W)</p> <p>vpce-07a48d414059bafb4-0tiznt17-ap-northeast-1a.api.ecr.ap-northeast-1.vpce.amazonaws.com (Z2E726K9Y6RL4W)</p>			

- 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/chainer\_p36/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。

```
"sts" : {
  "defaults" : {
    "credentialScope" : {
      "region" : "ap-northeast-1"
    },
    "hostname" : "vpce-07a04bf233a7bbccb-yxh1tfbr.sts.ap-northeast-1.vpce.amazonaws.com"
  },
  "endpoints" : {
    "ap-east-1" : {
      "credentialScope" : {
        "region" : "ap-east-1"
      },
      "hostname" : "sts.ap-east-1.amazonaws.com"
    },
    "ap-northeast-1" : {
      "credentialScope" : {
        "region" : "ap-northeast-1"
```

```
    },
    "hostname" : "vpce-07a04bf233a7bbccb-yxh1tfbr.sts.ap-northeast-1.vpce.amazonaws.com"
  },
```

```
  "api.sagemaker" : {
    "endpoints" : {
      "ap-northeast-1" : {
        "hostname" : "vpce-0a8e80e89e089bef4-0oxjfy5.api.sagemaker.ap-northeast-1.vpce.amazonaws.com"
      },
```

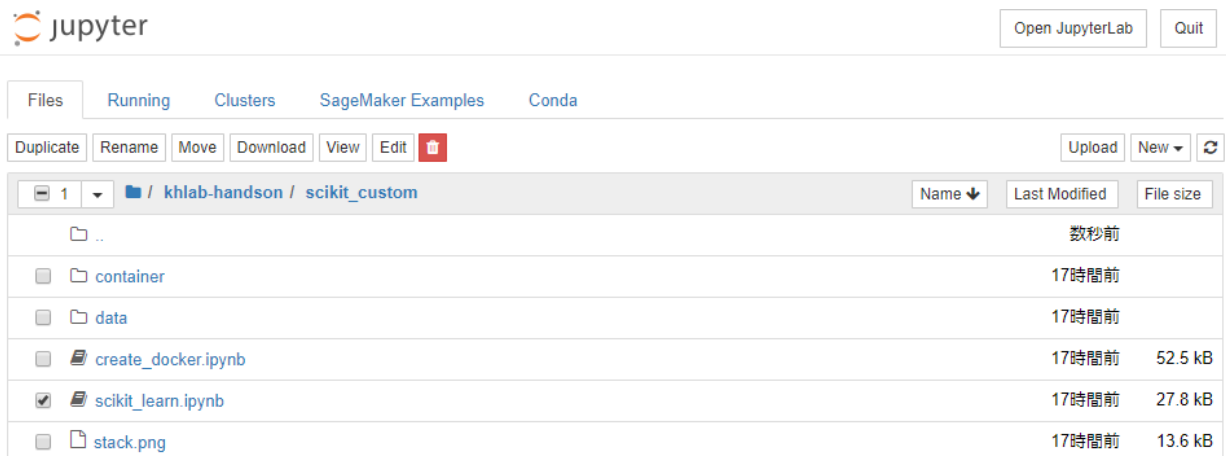
```
    "logs" : {
      "endpoints" : {
        "ap-east-1" : { },
        "ap-northeast-1" : {
          "hostname" : "vpce-07b1d8b6b579c9244-9xycdsvm.logs.ap-northeast-1.vpce.amazonaws.com"
        },
```

```
    "runtime.sagemaker" : {
      "endpoints" : {
        "ap-northeast-1" : {
          "hostname" : "vpce-0b91622fd0b7b989a-xi93pg1w.runtime.sagemaker.ap-northeast-1.vpce.amazonaws.com"
        }
      }
    },
```

```
  "api.ecr" : {
    "endpoints" : {
      "ap-east-1" : {
        "credentialScope" : {
          "region" : "ap-east-1"
        },
        "hostname" : "api.ecr.ap-east-1.amazonaws.com"
      },
      "ap-northeast-1" : {
        "credentialScope" : {
          "region" : "ap-northeast-1"
        },
        "hostname" : "vpce-0a1d76bfb01161444-m5pyrmpt.api.ecr.ap-northeast-1.vpce.amazonaws.com "
      },
```

# モデル学習の実行

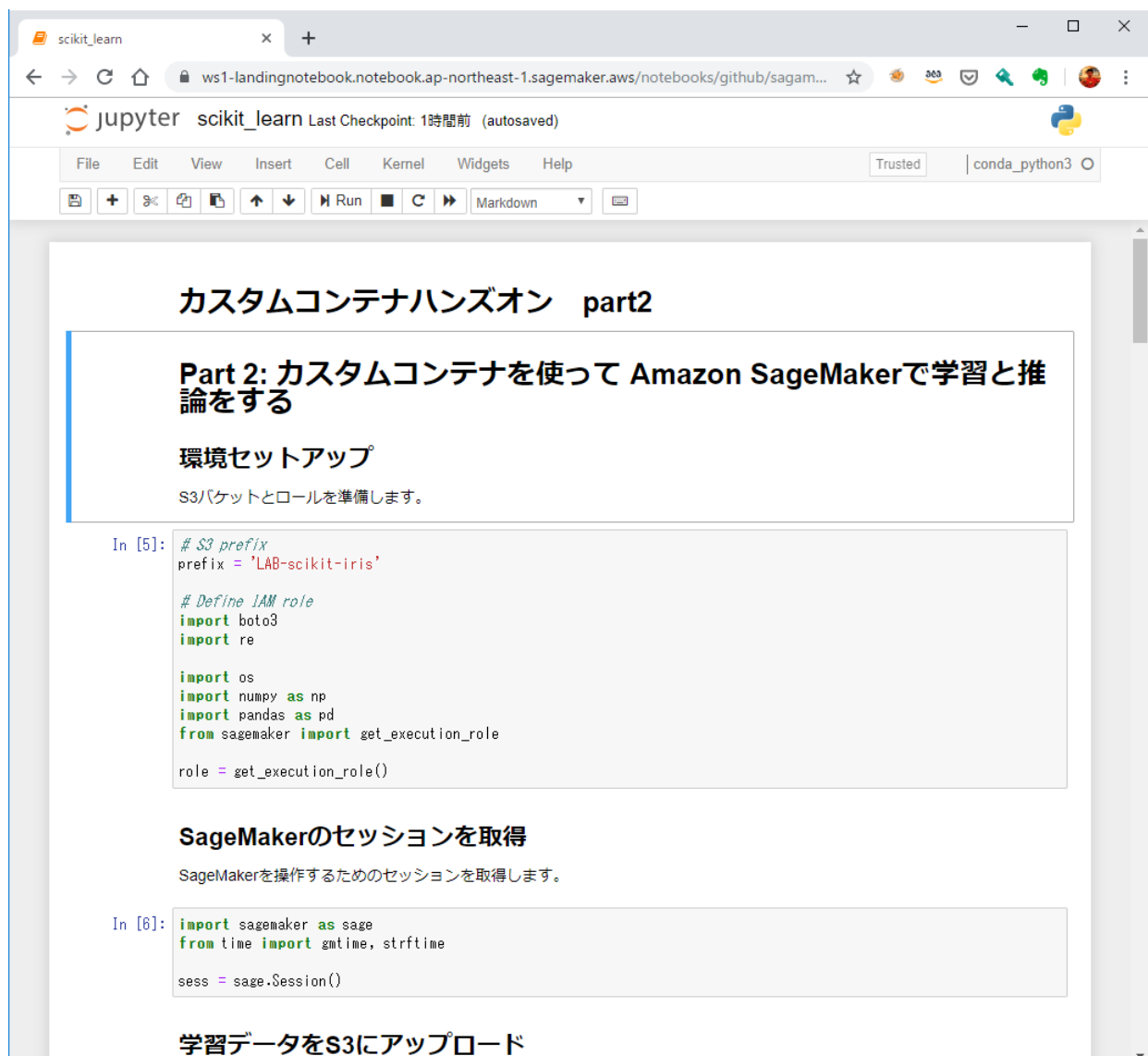
- scikit-learn.ipynb を開く



The image shows the JupyterLab file browser interface. At the top, there are tabs for 'Files', 'Running', 'Clusters', 'SageMaker Examples', and 'Conda'. Below the tabs, there are buttons for 'Duplicate', 'Rename', 'Move', 'Download', 'View', 'Edit', and a trash icon. On the right, there are buttons for 'Upload', 'New', and a refresh icon. The main area displays a file list for the directory '/ khlab-handson / scikit\_custom'. The list includes a '..' directory, a 'container' directory, a 'data' directory, a 'create\_docker.ipynb' file, a 'scikit\_learn.ipynb' file (which is selected), and a 'stack.png' file. Each entry shows its name, last modified time, and file size.

Name	Last Modified	File size
..	数秒前	
container	17時間前	
data	17時間前	
create_docker.ipynb	17時間前	52.5 kB
scikit_learn.ipynb	17時間前	27.8 kB
stack.png	17時間前	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。



The image is a screenshot of a Jupyter notebook titled 'scikit\_learn'. The browser address bar shows the URL 'ws1-landingnotebook.notebook.ap-northeast-1.sagemaker.aws/notebooks/github/sagam...'. The JupyterLab interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and markdown. The notebook content is in Japanese and is titled 'カスタムコンテナハンズオン part2'. It contains two sections: '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' and '環境セットアップ'. The '環境セットアップ' section includes a code cell with Python code for setting up the environment, including importing boto3, re, os, numpy, pandas, and sagemaker, and defining a role. The next section is 'SageMakerのセッションを取得', which includes a code cell with Python code for importing sagemaker, time, and defining a session. The final section is '学習データをS3にアップロード'.

## 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