# 



AWS Immersion Day

Personalize Hands-On Lab

# Overview

[Amazon Personalize](https://aws.amazon.com/personalize/) is a machine learning service that makes it easy for developers to create individualized recommendations for customers using their applications.

Machine learning is being increasingly used to improve customer engagement by powering personalized product and content recommendations, tailored search results, and targeted marketing promotions. However, developing the machine learning capabilities necessary to produce these sophisticated recommendation systems has been beyond the reach of most organizations today due to the complexity of developing machine learning functionality. Amazon Personalize allows developers with no prior machine learning experience to easily build sophisticated personalization capabilities into their applications, using machine learning technology perfected from years of use on Amazon.com.

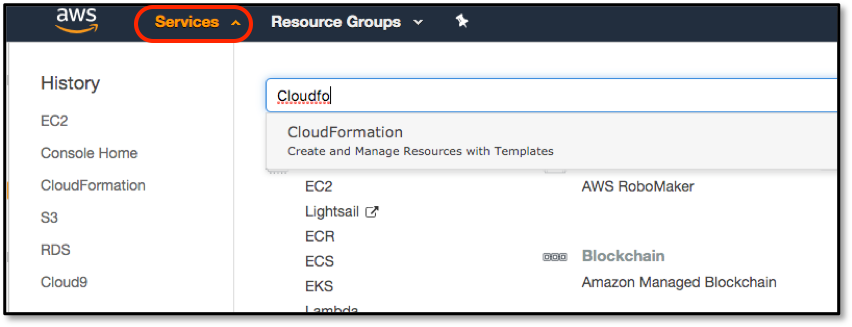
This lab will walk you through the following:

* Setting up a SageMaker Jupyter Notebook environment to interact with Amazon Personalize
* Downloading and preparing training data
* Importing prepared data into Amazon Personalize
* Building an ML model based upon the Hierarchical Recurrent Neural Network (HRNN) algorithm
* Testing your model by deploying an Amazon Personalize campaign

A live demo of what we’ll be building today is available here, but using a movie suggestion dataset: <https://www.personalisevideorecs.info/recommend/>

# Deploy the CloudFormation Template

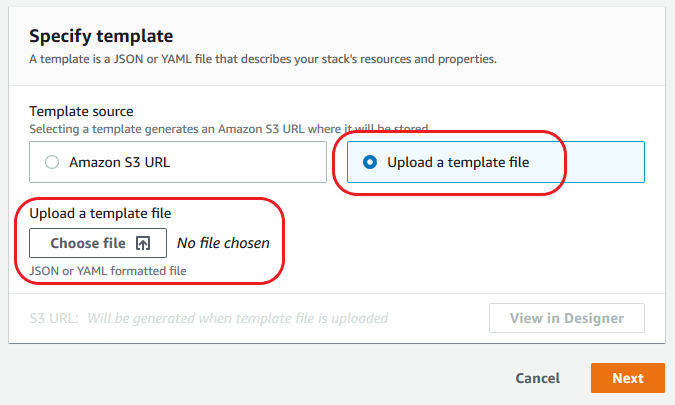
1. Click on the **Services** dropdown and select **CloudFormation** from the list of all services by entering CloudFormation into the Find services box. This will bring you to the Amazon CloudFormation console home page.

[](https://ai-services.go-aws.com/images/consoleCfnSelect.png)

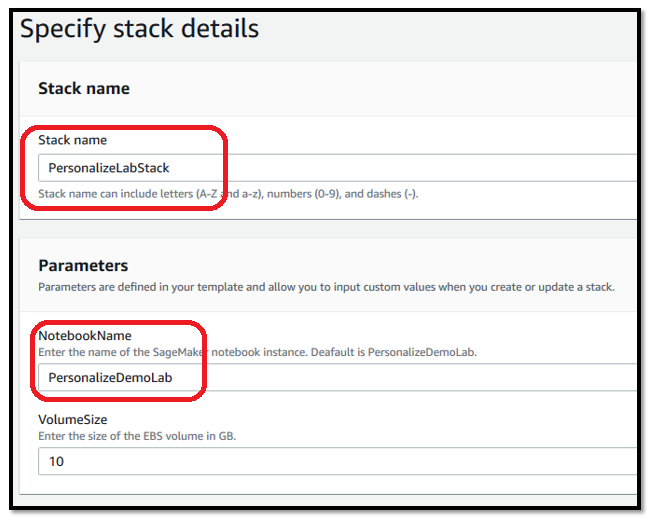
1. We are going to deploy a pre-built SageMaker Jupyter notebook that is populated with the required files from a GitHub repository. This is done automatically by using a CloudFormation template. Furthermore, this template will generate an IAM role that we will use later to allow SageMaker to have enough permissions to use S3 and Personalize services. Navigate to the following link which will open the template we need. Using your browser, save the page on your local machine as a YML file.

[Github.com/vavourak/bp-personalize-lab-2020/master/personalize\_cloudformation\_template.yml](https://raw.githubusercontent.com/vavourak/bp-personalize-lab-2020/master/personalize_cloudformation_template.yml)

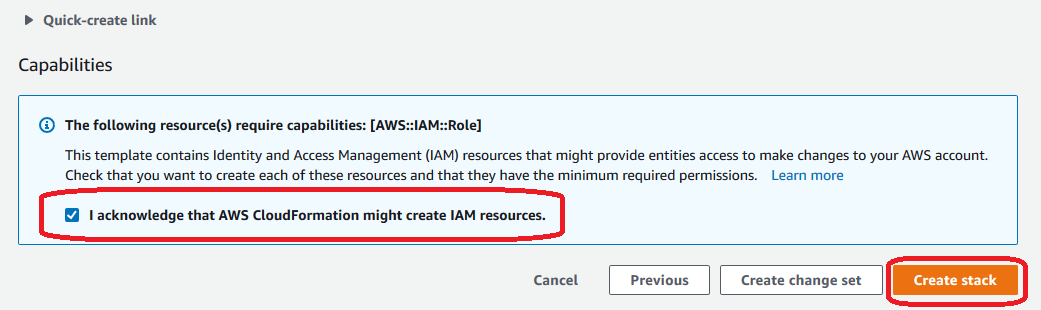
1. On the CloudFormation screen, click on the **Create Stack** button to start the deployment wizard, and in the **Choose a template** section select **Upload a template file**, click on the **Choose file** button, and select the template file that you just downloaded. Then click on **Next**.



1. The next screen asks for more configuration parameters, but only two of these are required: **Stack name** and **NotebookName**. For Stack name enter something simple, such as *PersonalizeLabStack*, and for NotebookName something like *PersonalizeDemoLab*, then click **Next** (not shown).



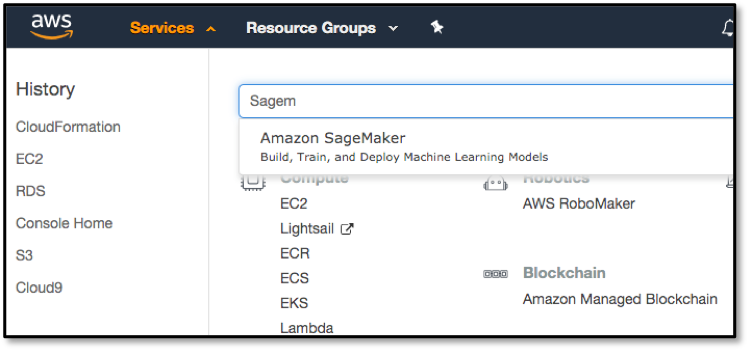
1. There will then be two additional screens. The first is called Options, but we have none to enter, so just click on **Next**. The second is the final Review screen, there is a confirmation box we need to check before we continue. This confirms that our CF template can create the required IAM role as mentioned earlier. Once checked, click *Create stack*. This will then go and create the environment, which will take around 5 minutes. Once the console returns to the main CloudFormation screen, you can continue with the next lab step.



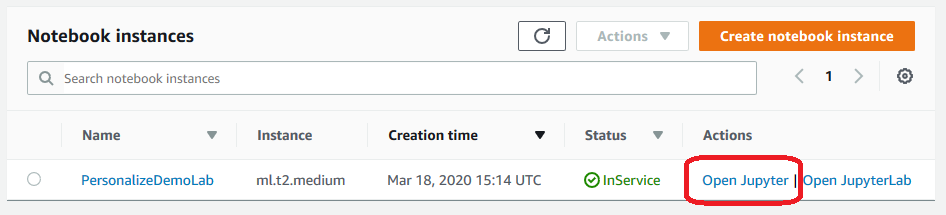
# Launch Jupyter Notebook

### Launching a Jupyter Notebook using Amazon SageMaker

1. Click on **Amazon SageMaker** from the list of all services by entering *Sagemaker* into the **Find services** box. This will bring you to the Amazon SageMaker console homepage.

[](https://ai-services.go-aws.com/images/consoleSMSelect.png)

1. The CloudFormation template created a Jupyter notebook instance for us, so go to **Notebook instances** in the Amazon SageMaker console, and click the **Open Jupyter** button at the right of the screen.



# Notebook Setup

**Exploring the files**

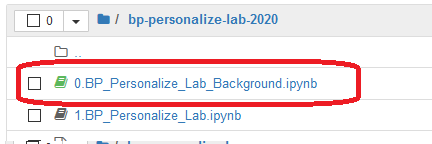
With our CloudFormation template, the notebook was automatically populated with files from the lab’s Git repository, including the datasets. The lab contains 2 notebook *.ipynb* files that we will use:

* 1.BP\_Personalize\_Lab.ipynb – The main notebook we will work through.
* 0.BP\_Personalize\_Lab\_Background.ipynb – A notebook with code that we will use to do processing in the background to speed up the lab.

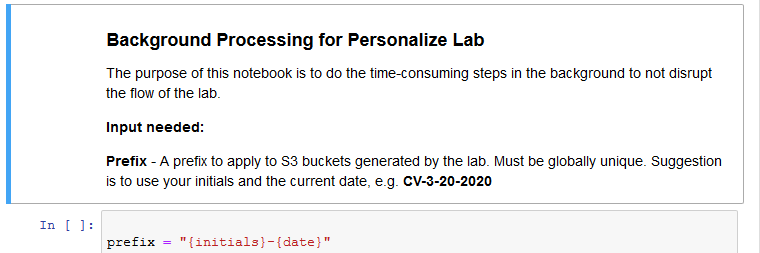
**Starting the Background Processing and Working Through a Jupyter Notebook**

There are some steps in the lab take some time to complete (10-30 minutes). In order to reduce waiting times, we will start some of the processing in the background ahead of time. The *0.BP\_Personalize\_Lab\_Background* notebook contains essentially the exact same code we will go through later. Let’s start the background processing:

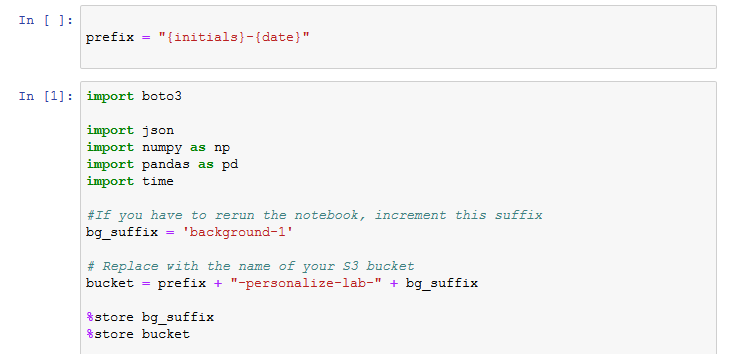
1. Click on the *0.BP\_Personalize\_Lab\_Background.ipynb* file to start up the Jupyter notebook in a new browser tab.



1. A notebook consists of a number of cells; in SageMaker these will typically either be *Code* or *Markdown* cells. Markdown is used to allow for documentation to be defined inline with the code, giving the author a rich set of markdown formatting options. The first cell in this notebook, which is called **Background Processing for Personalize Lab**, is Markdown, and if you select any cell then the whole cell is highlighted.



1. The first Markdown cell describes what the following Code cell is going to do – for the sake of this lab you do not have to understand the code that is being run in the Code cell, rather you should just appreciate what the notebook is doing and how you interact with a Jupyter notebook.



1. To the left of a Code module is a set of empty braces **[ ]**. By highlighting the cell and then selecting the *Run* command in the menu bar, the Jupyter notebook will execute this code, outputting and code outputs to the notebook screen and keeping any results data internally for re-use in future steps. Do this now to execute the first code cell.

\*Note: if a Markdown cell is highlighted, then clicking **Run** will move the highlight to the next cell\*

1. Whilst the code is executing the braces will change to be **[\*]**, indicating that it is executing, and once complete will change to **[1]**. Future cells will have increasing numbers inside the braces, and this helps you see the order in which cells have been executed within the notebook. Directly below the code, but still within the Code cell, is the output from the code execution - this will include any error messages that your code has thrown.

[](https://ai-services.go-aws.com/images/loadBoto3Post.png)

1. Now let’s execute the two code cells in this notebook to start the background processing.
2. Once started, let’s go back to the browser tab that has the Jupyter file list. We will leave the background notebook tab open and processing.
3. Click on the *1.BP\_Personalize\_Lab.ipynb* notebook to open a new tab. This is the main workflow of this lab. Now please continue to work through the notebook - read the comments prior to each Code cell in order to get an understanding as to what is going on, as these explain why we are doing each step and how it ties in to using the Amazon Personalize service.