Elastic Map Reduce (EMR) and SageMaker Lab

**Overview**

In this lab you’ll train and deploy a machine learning model that estimates the age of abalone based on their physical characteristics.

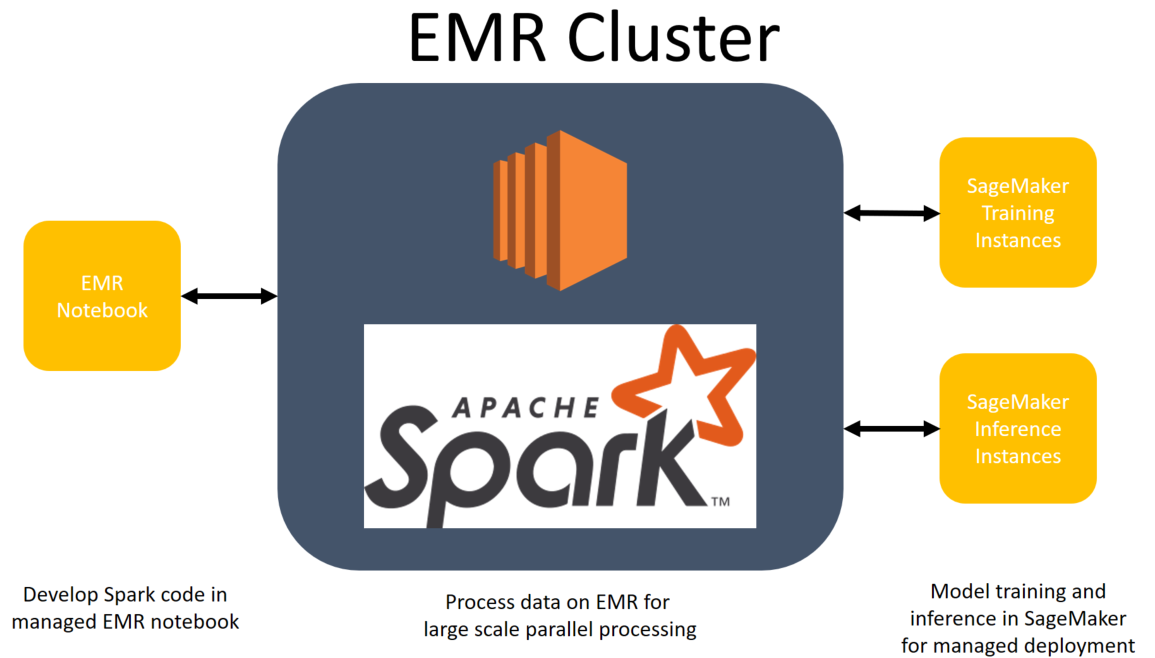
**Motivation**

[Apache Spark](https://spark.apache.org/) on EMR is a popular tool for processing data for machine learning. Using Spark you can enrich and reformat large datasets. EMR Notebooks are familiar Jupyter notebooks that can connect/disconnect to EMR clusters and run Spark jobs on the cluster. The notebook code is persisted durably to S3. By using EMR Notebooks you separate the execution (EMR) and code development (Notebook) environments. Using EMR Notebooks allows you to share the EMR cluster between users and move the development environment to other clusters as needed.

After processing the data in Spark, you’ll need to create, train, and host a machine learning model that estimates the abalone age. Amazon SageMaker provides managed infrastructure for model training and deployment. By using SageMaker for model training and deployment we decouple the modeling from the data processing. This allows you to right size all of the infrastructure for each component.

In this lab you create an Elastic Map Reduce (EMR) notebook that runs Spark jobs on the EMR cluster created in EMR Lab #1. You’ll use PySpark to create a regression model that estimates abalone age. You’ll deploy the regression model in Amazon SageMaker and measure the effectiveness of your model.

***Note:*** *Make sure to have Livy installed on the EMR Cluster from EMR Lab #1.*

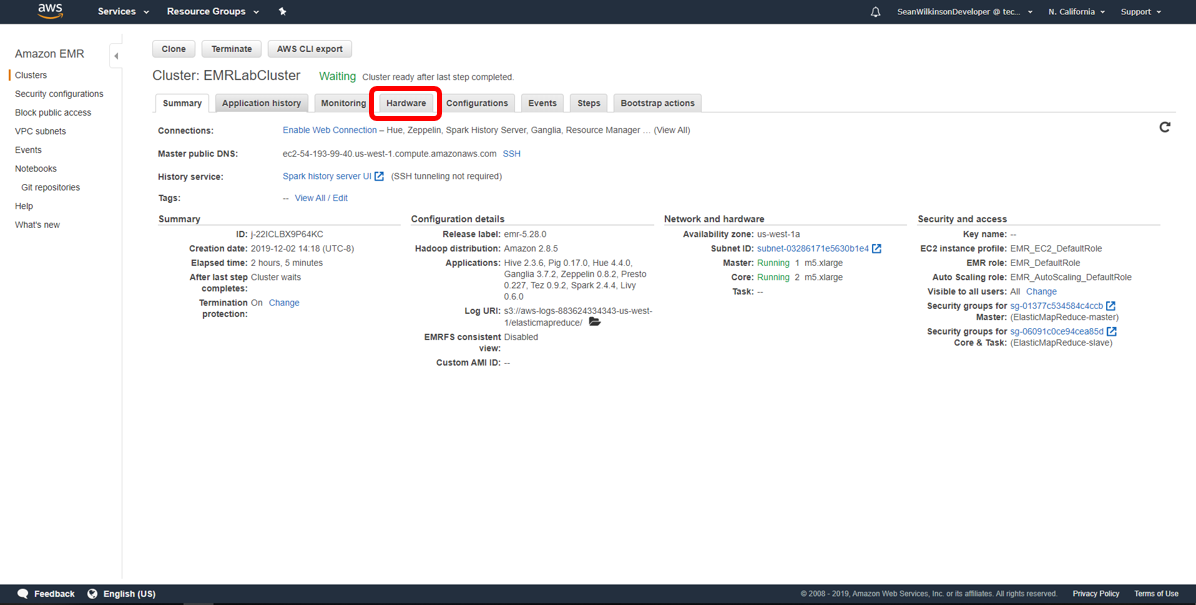


**Image 1 Architecture Overview**

1. **Modify IAM role for EMR**

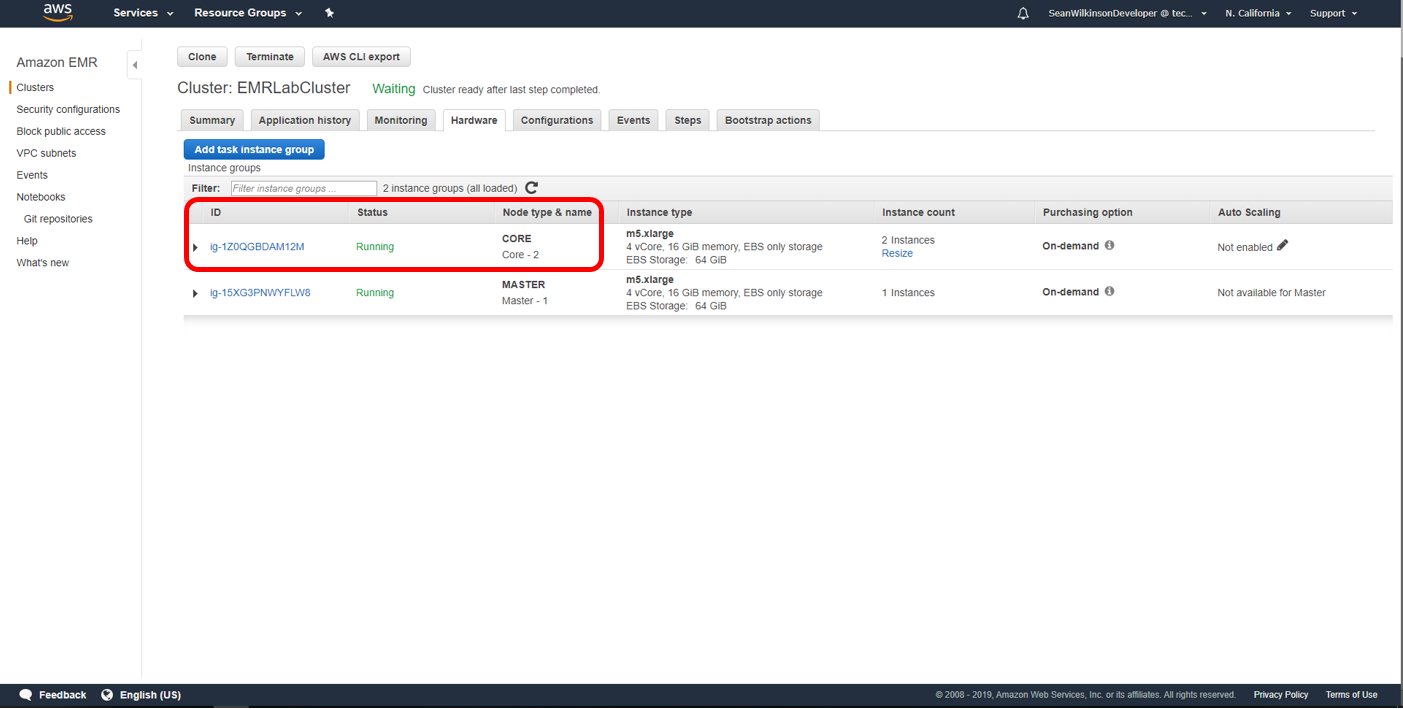
For this lab you’ll use the EMR cluster previously created in EMR Lab #1. In order for the EMR cluster to access SageMaker resources, you need to expand its IAM permissions.

* 1. Navigate to the EMR console within AWS.
  2. Select the Cluster you created in Lab # 1.
  3. Click on the ‘Hardware’ tab as shown below.



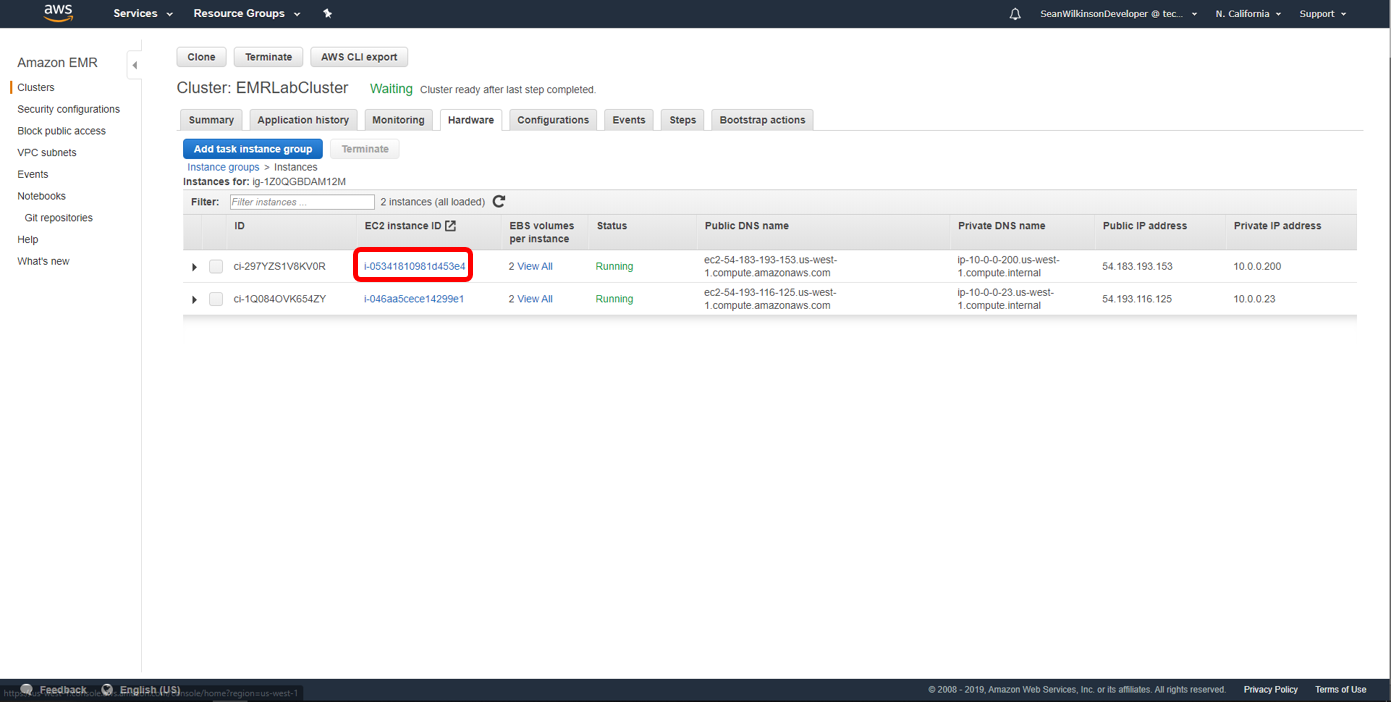
**Image 2 Step 1c**

* 1. Click on the ‘ID’ for the core nodes in your cluster. You can tell if the instance ID corresponds to core nodes if the Node type column says ‘CORE’.



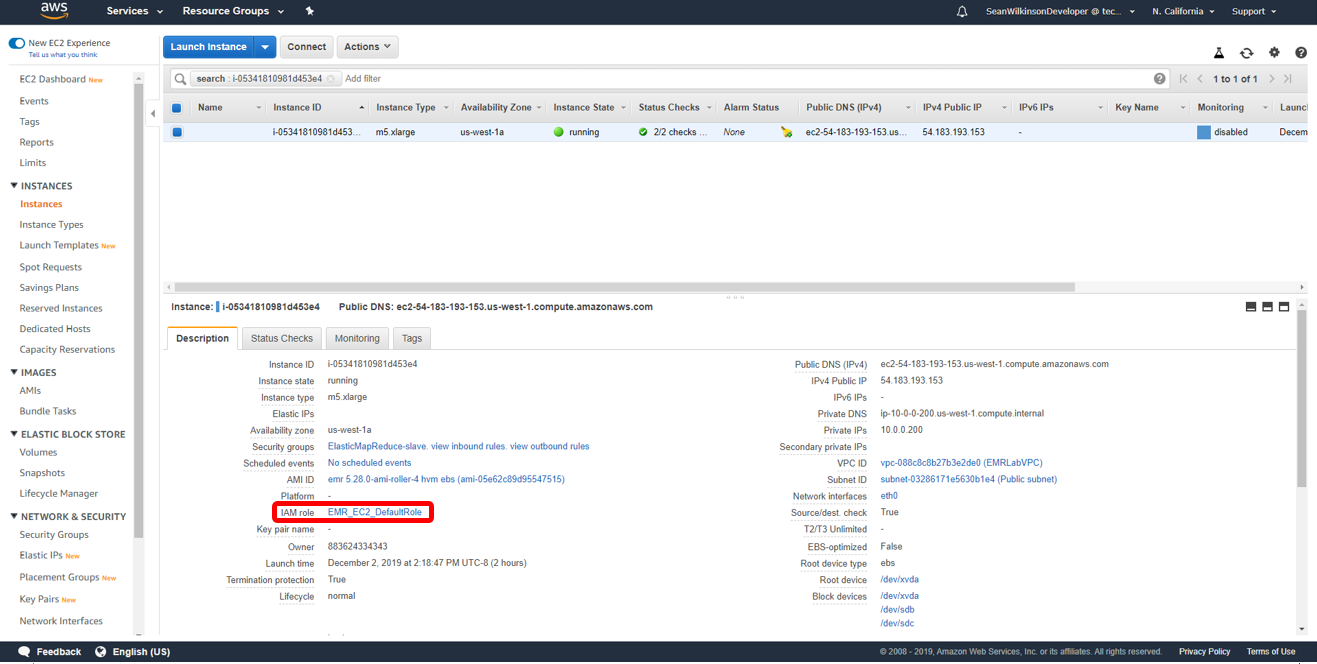
**Image 3 Step 1d**

* 1. Select the ‘EC2 instance ID’ for one of the core nodes. They all have the same IAM role associated with them, so you may select any of the core nodes.



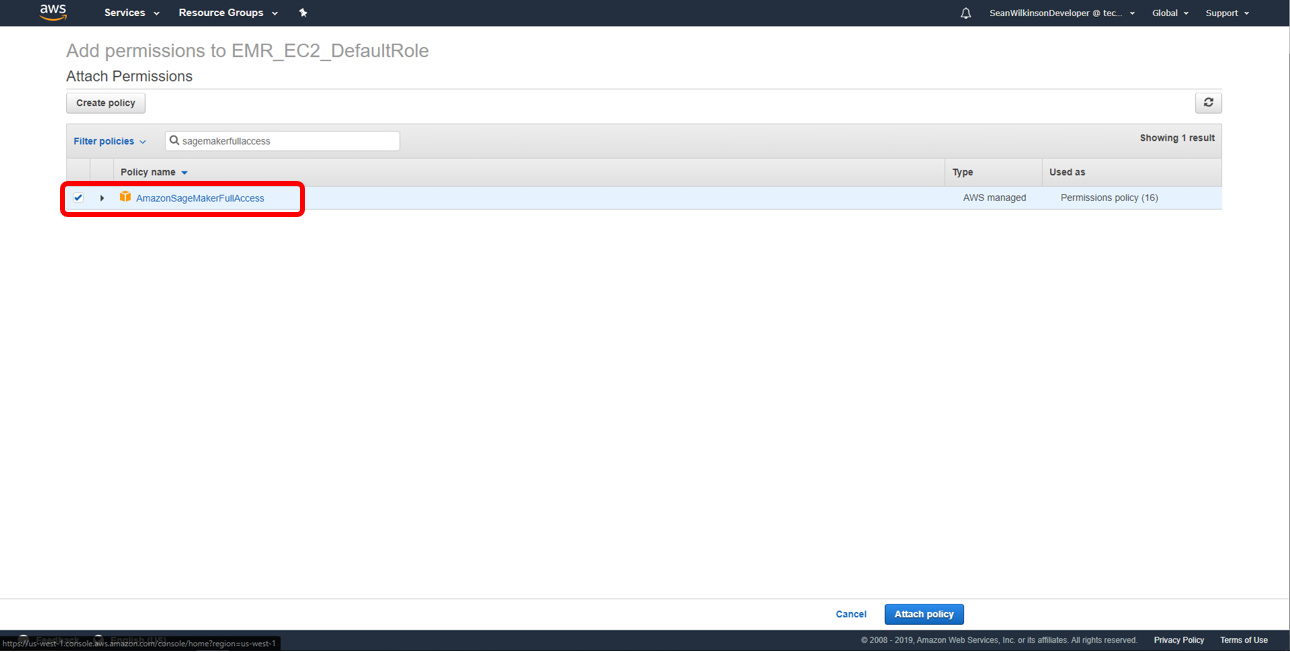
**Image 4 Step 1e**

* 1. Scroll down and select the role associated with the core nodes. It will be next to ‘IAM role’, as shown below.



**Image 5 Step 1f**

* 1. Click ‘Attach policies’.
  2. Search for ‘SageMaker’ in the search bar.
  3. Select the box next to ‘AmazonSageMakerFullAccess’, then click ‘Attach policy’.



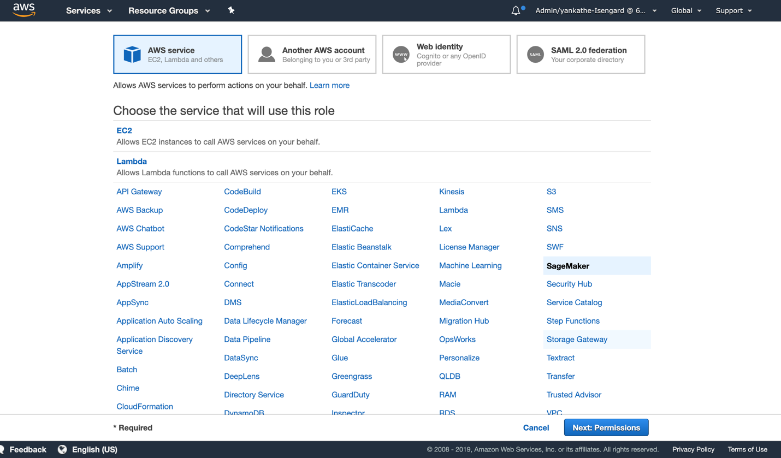
**Image 6 Step 1i**

Your EMR cluster now has permissions to access SageMaker.

1. **Create a SageMaker IAM role**

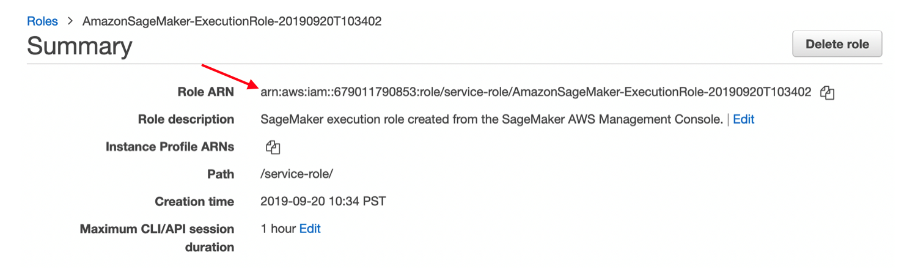
During model training, SageMaker requires an IAM role to access S3 training data and create compute resources. First, let’s create an IAM role for SageMaker.

* 1. In the AWS console, select ‘Services’, then search for IAM.
  2. On the left panel, click ‘Roles’.
  3. Click ‘Create Role’ and select type of trusted entity to be AWS service.
  4. Choose the service that will use this role to be SageMaker.
  5. Click ‘Next:Permissions’.

****

**Image 7 Step 2e**

* 1. Click ‘Next:Tags’.
  2. Click ‘Next:Review’.
  3. For Role Name, enter ‘SageMaker-EMR-ExecutionRole’ and Create Role.
  4. Search for the newly created role within the IAM console. Click on the Role name.
  5. Copy the entire Role ARN and paste it into a text file. You will enter this ARN in to the Jupyter notebook at a later step.

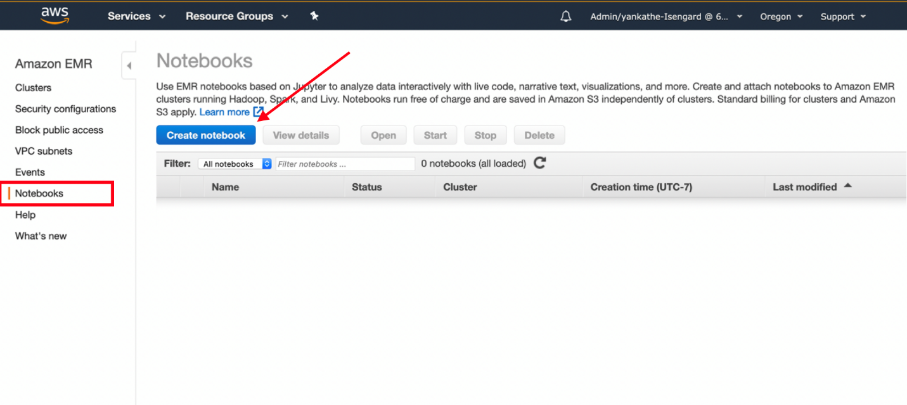


**Image 8 Step 2j**

1. **Create an EMR Notebook**

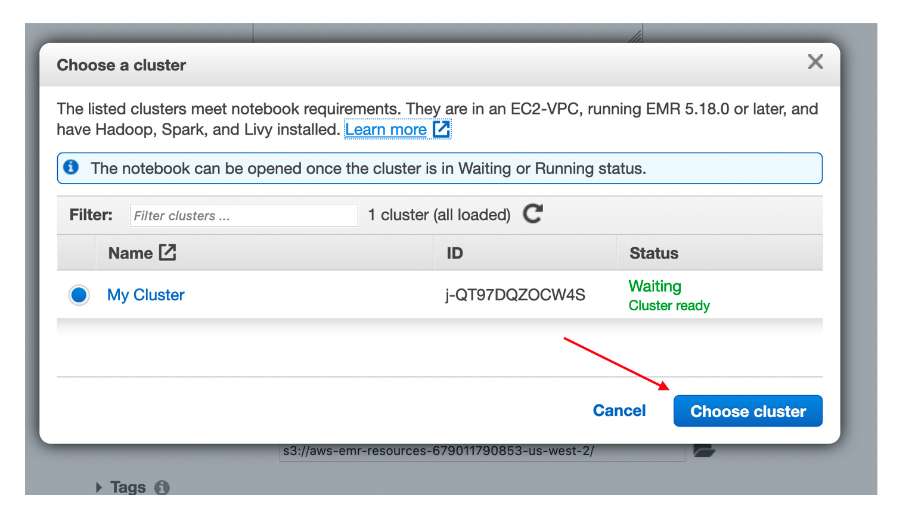
With the IAM permission set up, you can now create your EMR Notebook. EMR Notebooks are serverless Jupyter notebooks that connect to an EMR cluster using [Apache Livy](https://livy.apache.org/). They come preconfigured with Spark, allowing you to interactively run Spark jobs in a familiar Jupyter environment. The code and visualizations that you create in the notebook are saved durably to S3.

1. In the EMR console, click ‘Notebooks’.
2. Click ‘Create notebook’.



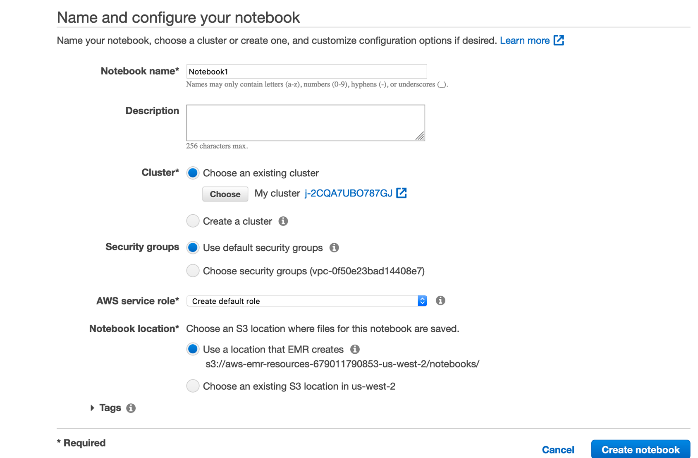
**Image 9 Step 3b**

1. Name the notebook ‘Notebook1’ and add an optional description.
2. Choose an existing cluster, and click ‘Choose’.
3. Click the radio button next to the cluster you created in Lab #1 and click ‘Choose cluster’.



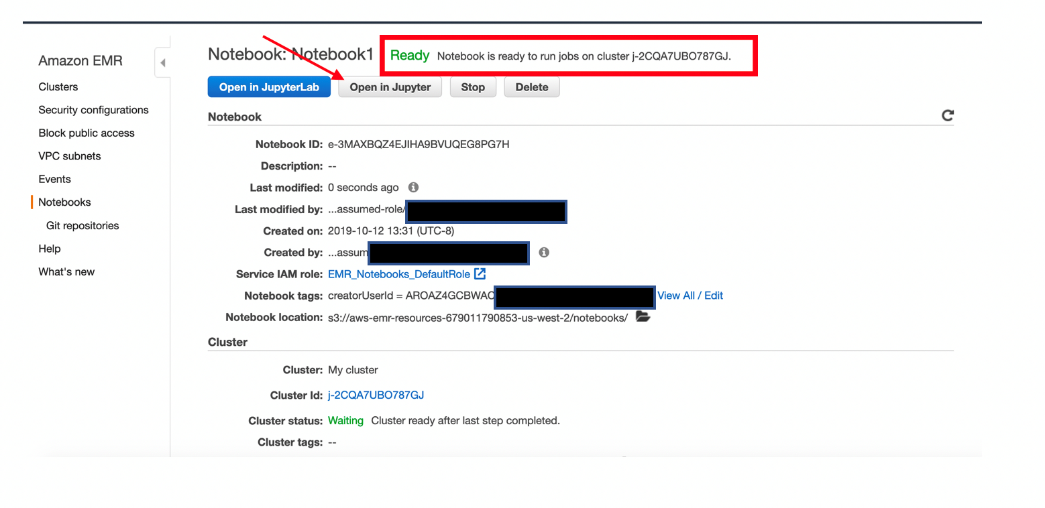
**Image 10 Step 3e**

1. Leave Security Groups as default.
2. Leave AWS service role as ‘Create default role’.
3. Under ‘Notebook Location’, select ‘Use a location that EMR creates’.



**Image 11 Step 3h**

1. Click ‘Create notebook’.
2. Refresh the screen until ‘Starting’ changes to ‘Pending’ and then ‘Ready’.
3. Click ‘Open in Jupyter’ to open your EMR Notebook.

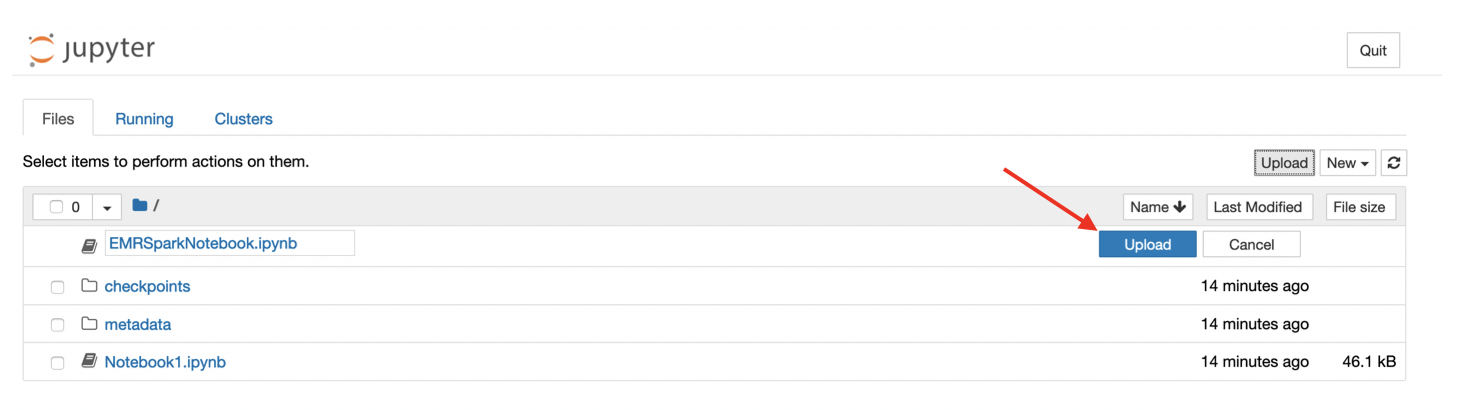


**Image 12 Step 3k**

1. **Run the example PySpark SageMaker notebook**
   1. Download the [example PySpark SageMaker](https://emr-lab-income-dataset.s3-us-west-2.amazonaws.com/EMRSparkNotebook.ipynb).
   2. In your Jupyter notebook, click the ‘Upload’ button and choose the .ipynb file downloaded in step 4a. Click ‘Open’.

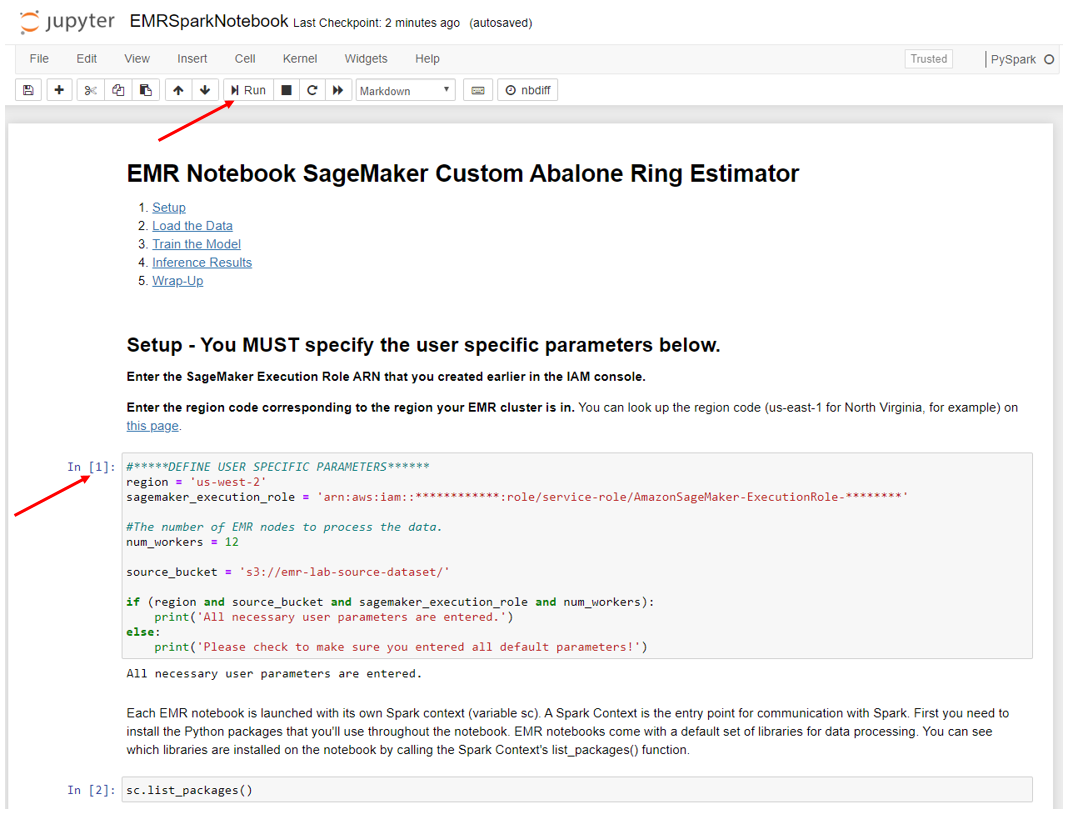
**Image 13 Step 4b**

* 1. Make sure to click ‘Upload’ once more.



**Image 14 Step 4c**

* 1. Click on the EMRSparkNotebook.ipynb to open it.
  2. Follow the instructions embedded in the notebook. Click on the “Run” button at the top of the screen to execute the notebook cell by cell. Once you finish the notebook, you completed the lab.
  3. The text to the left of the cell indicates whether the cell has been ran. The brackets in the text will be empty prior to running a cell (‘In []’). It will have an asterisk inside during execution (‘In [\*]’). And they will show the cell execution number after execution completes (‘In [1]’).



**Image 15 Step 4f**

* 1. Remember to enter the SageMaker ARN and the region code in the first cell. The notebook will throw errors otherwise!

1. **(Optional)** The Jupyter notebook contains built-in options you can choose to run. If time permits, step through them and try some of the following:
   1. Choose a different number of worker nodes to improve parallelization within your EMR cluster (Set num\_workers = 5)
   2. Do some feature engineering to improve your model’s predictions. Modify columns to see if it will improve the model.
   3. Choose a different model
      1. Pick between XGBoost or LinearLearner and see which model performs better.

**Try to lower the RSME (root-mean-square error) and improve your model.**