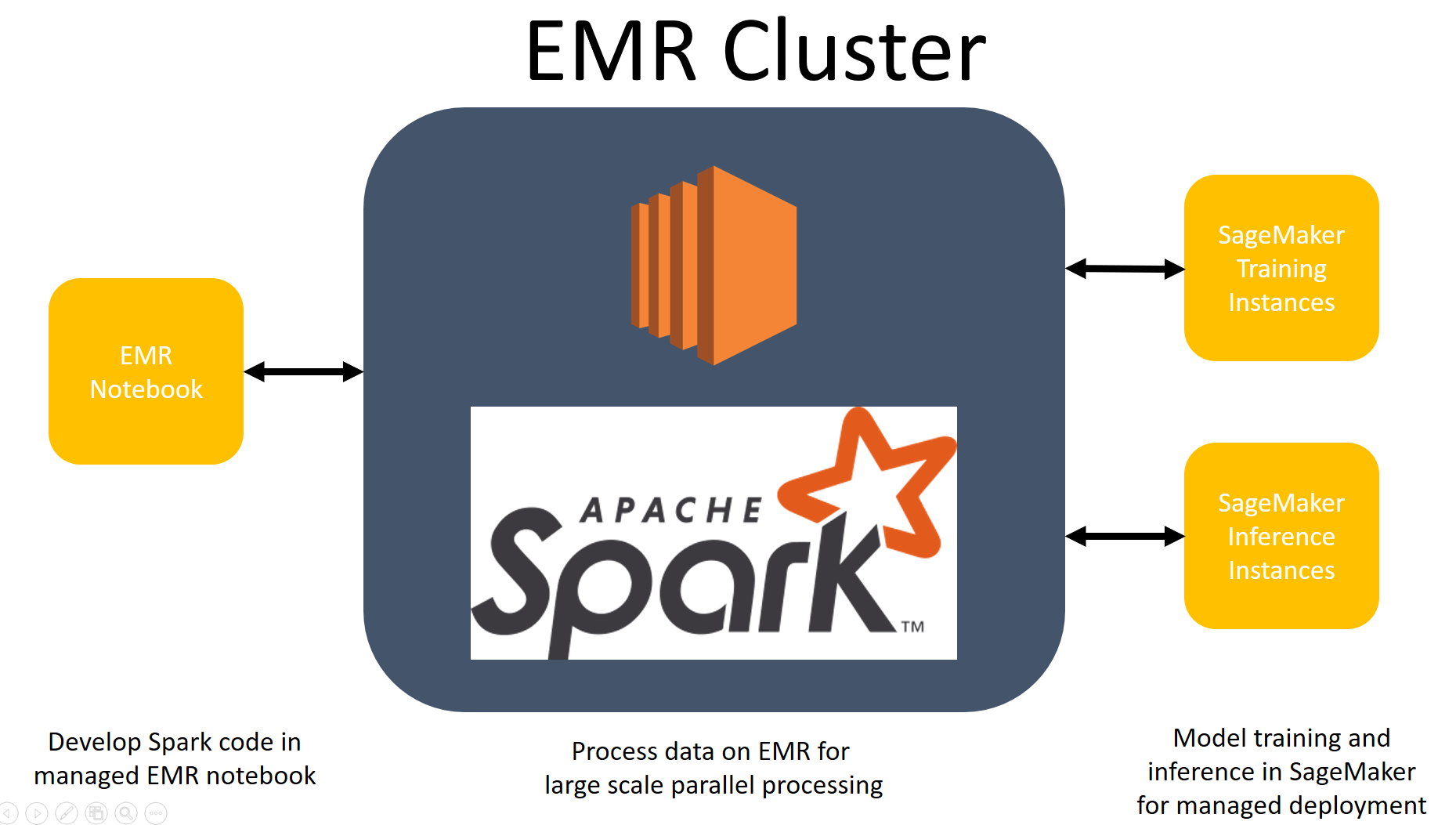
Elastic Map Reduce (EMR) and SageMaker Lab

This lab demonstrates how to create an Elastic Map Reduce (EMR) notebook that runs Spark jobs on the EMR cluster created in EMR Lab #1. The EMR notebook runs Spark to process data and calls SageMaker to create a machine learning model.

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

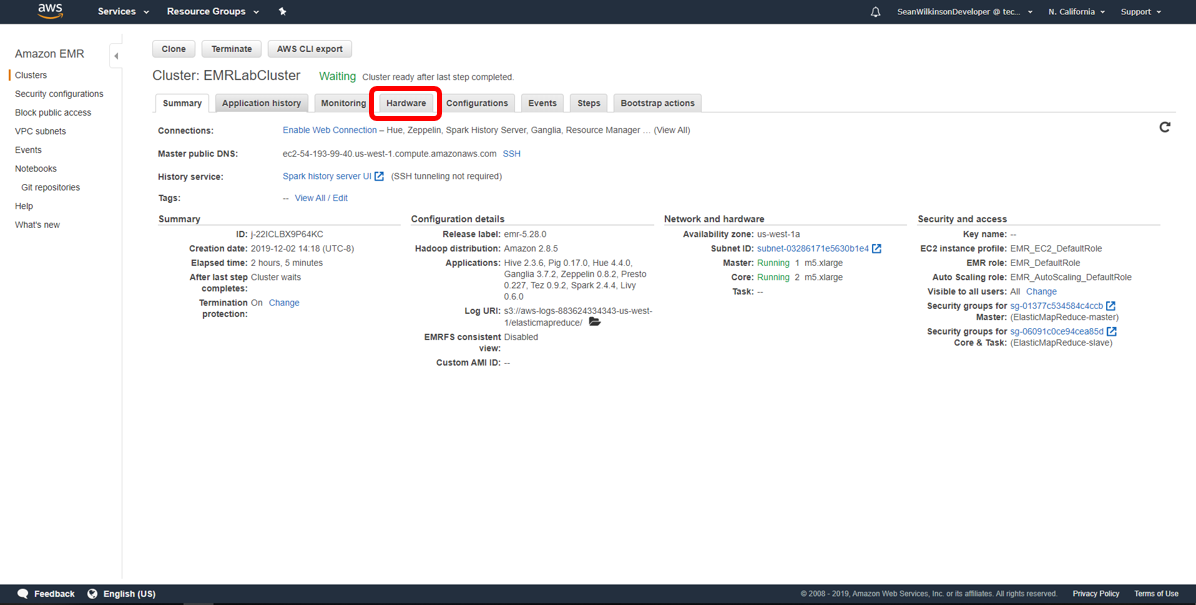


Amazon SageMaker provides a managed and repeatable machine learning model deployment environment. Apache Spark on EMR provides a distributed computing engine to process large datasets. By using the services together you can decouple the data engineering and cleaning stage from model deployment.

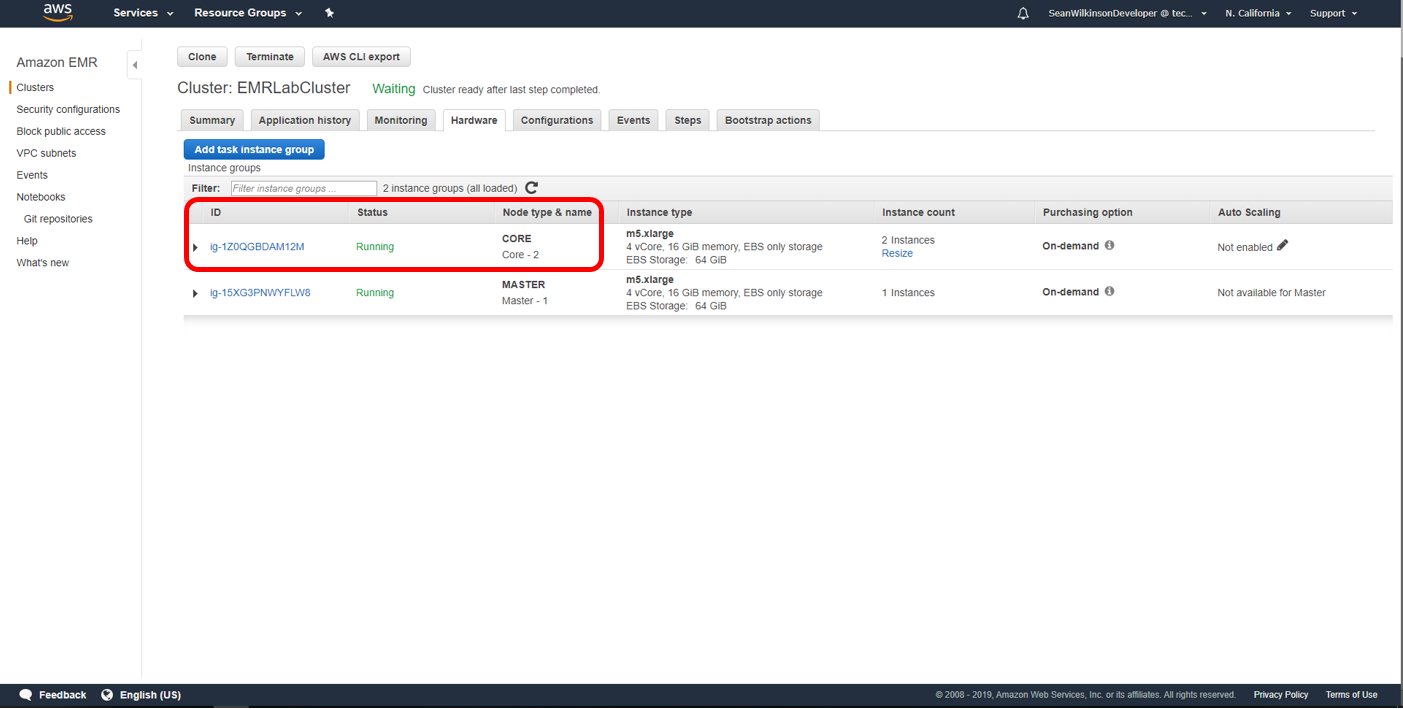
1. **Modify IAM role for EMR**

For this lab we’ll use the EMR cluster previously created in Lab #1. In order for the EMR cluster to create a SageMaker model, we need to expand its IAM permissions to include SageMaker.

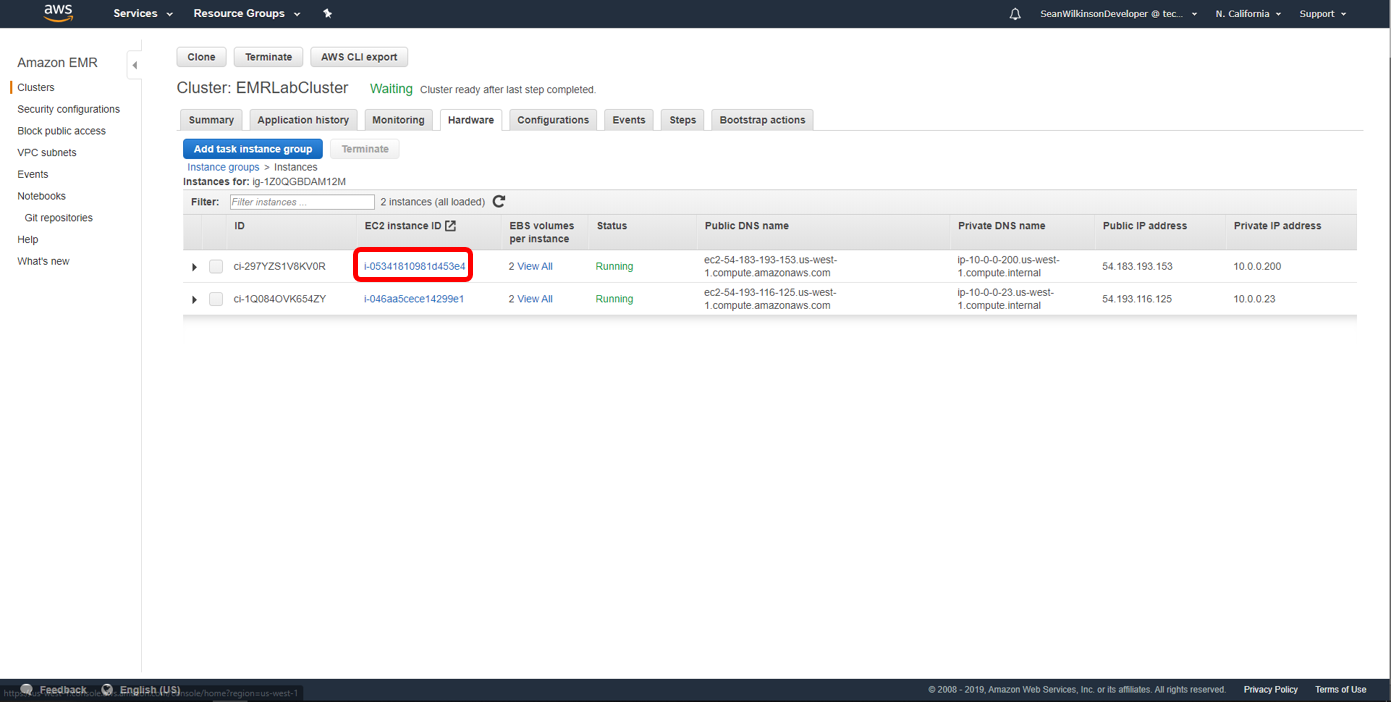
* 1. Navigate to the EMR console within AWS.
  2. Select the Cluster you created in Lab # 1.
  3. Click on the “Hardware” tab as shown below.



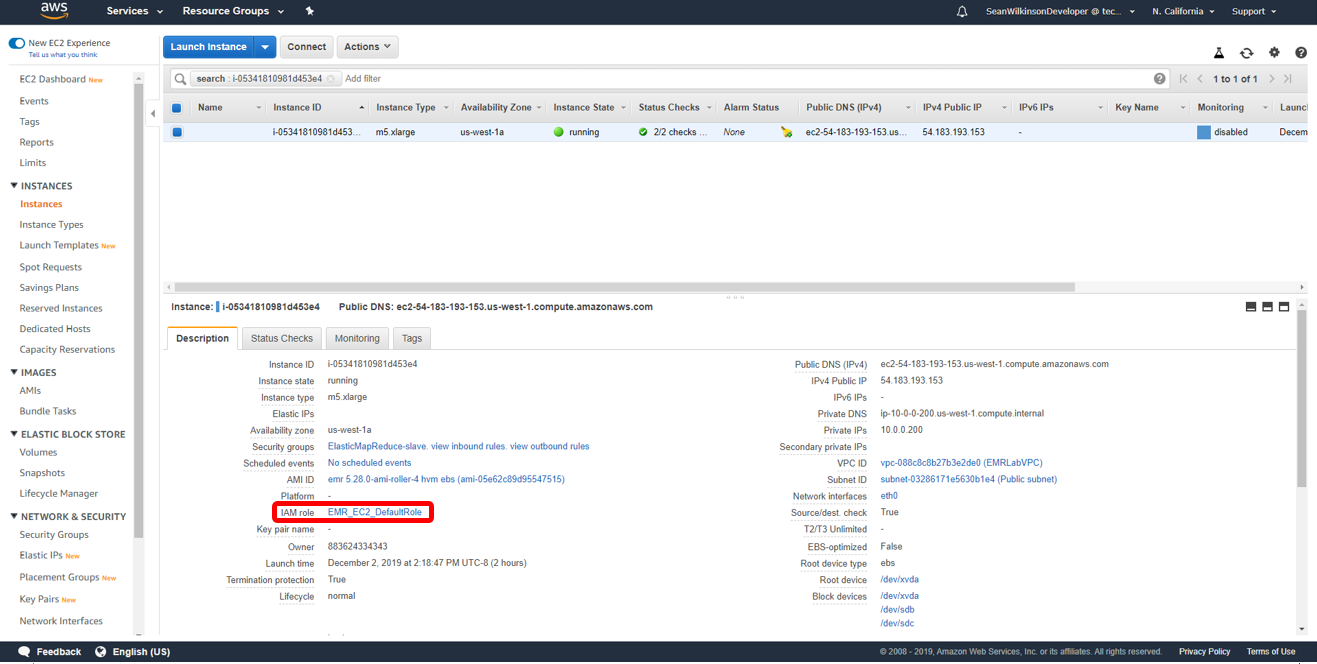
* 1. Click on the ‘EC2 instance ID’ for one of the Core nodes in your cluster. You can tell if the instance ID corresponds to a Core node if the Node type column says ‘CORE’.



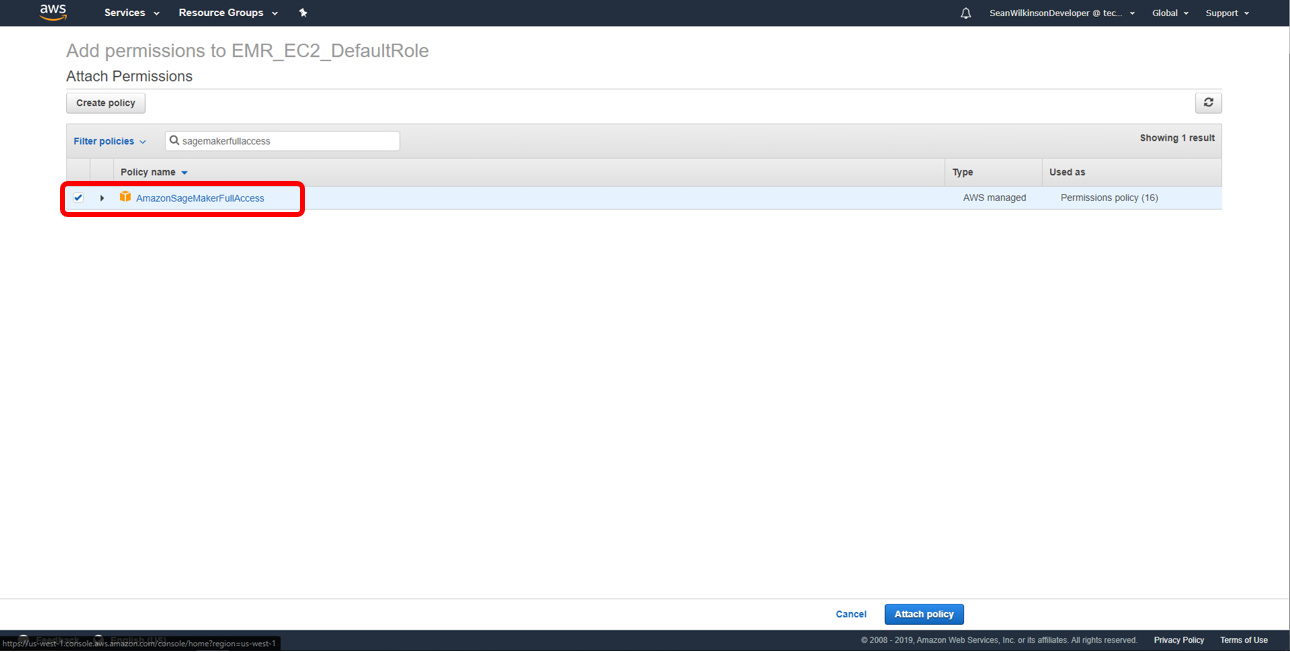
* 1. Select an instance ID for one of the Core nodes. It doesn’t matter which one you select as they all use the same IAM role.



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



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

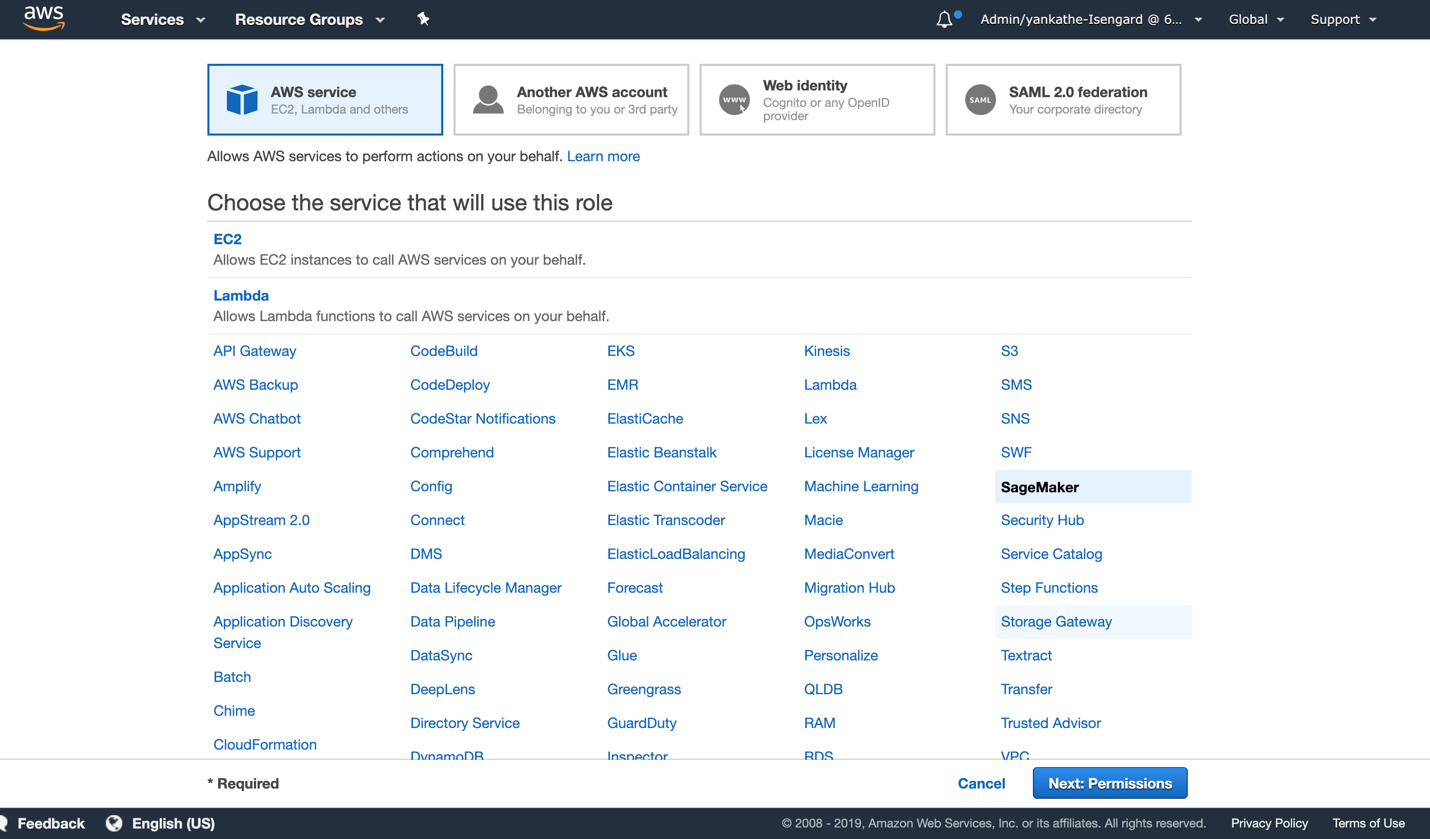


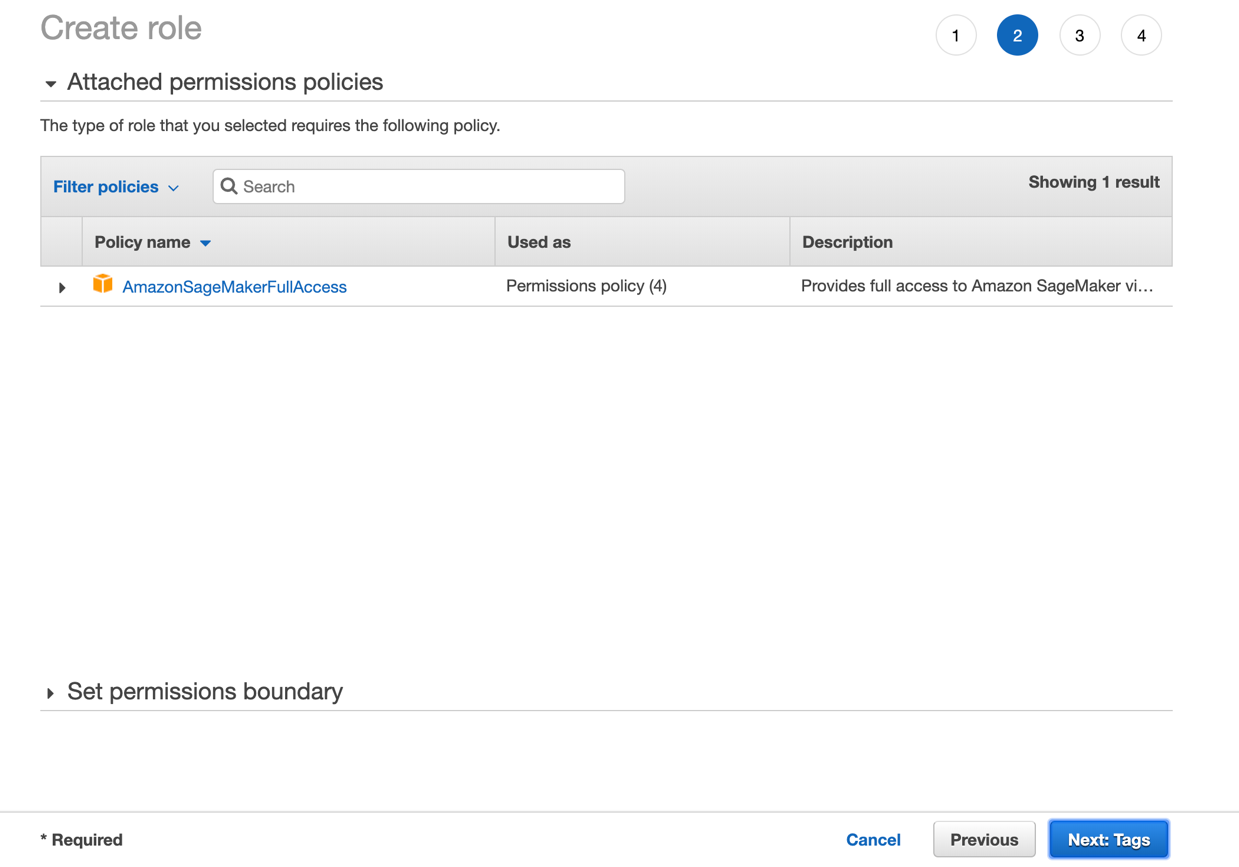
Your EMR cluster now permissions to call SageMaker.

1. **Create a SageMaker role**

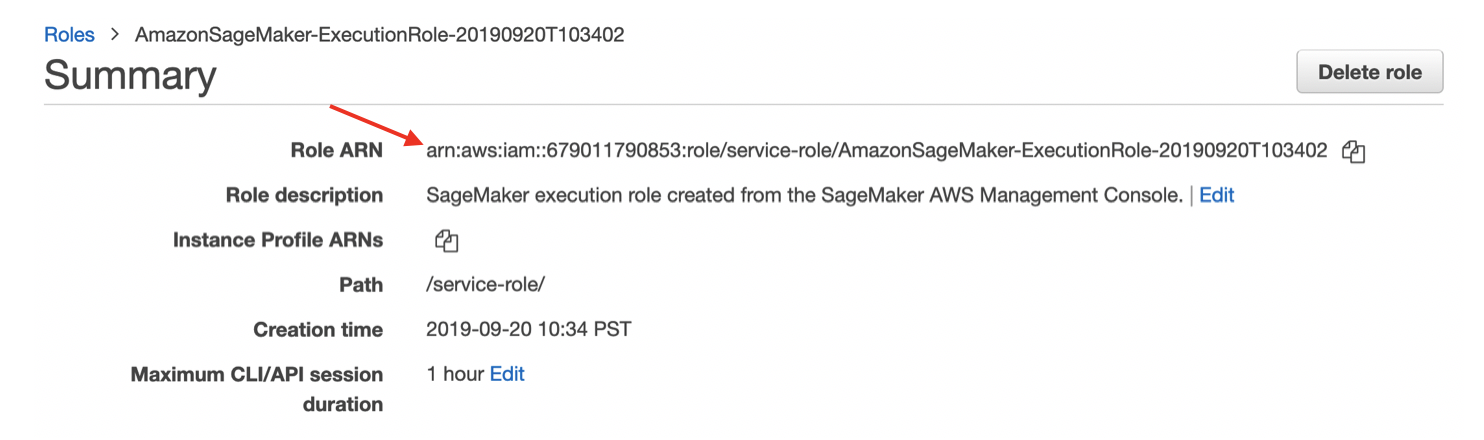
SageMaker requires an IAM role to access S3 and create SageMaker resources. First, let’s create an IAM role for SageMaker.

* 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.

****

****

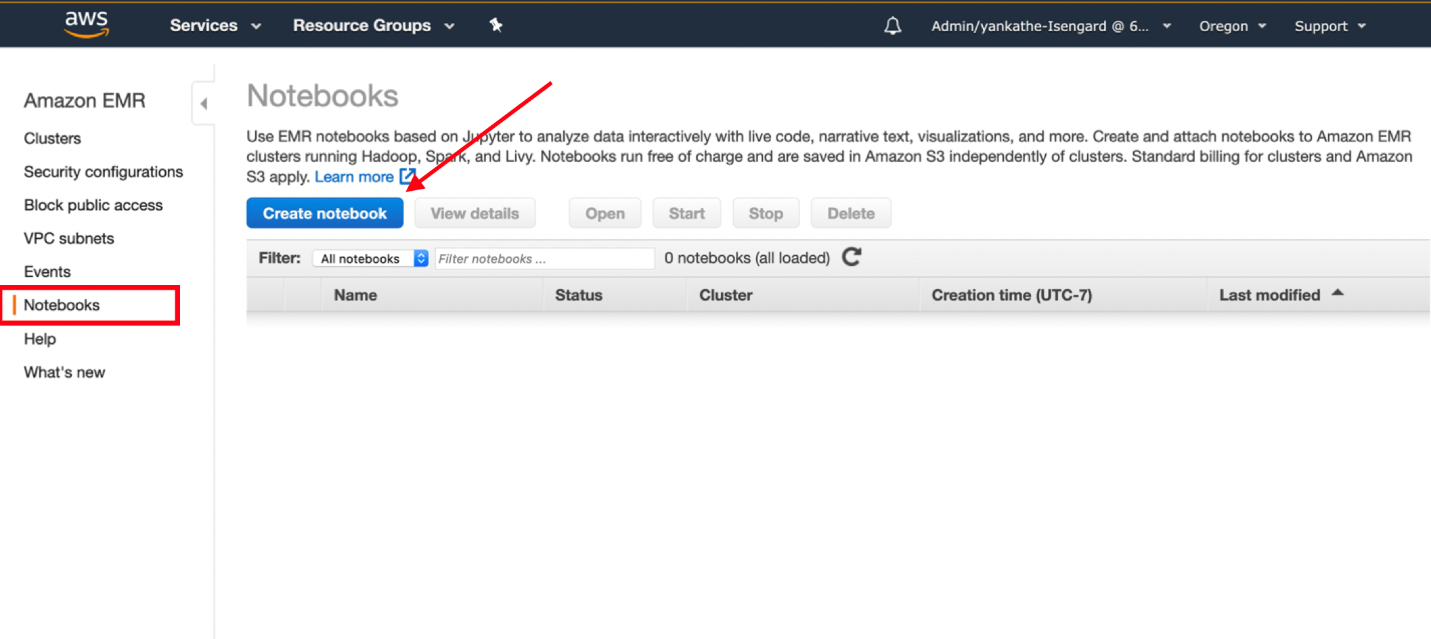
* 1. Click ‘Next:Tags’.
  2. Click ‘Next:Review’.
  3. For Role Name, enter ‘SageMaker-EMR-ExecutionRole’ and Create Role.
  4. Search for the newly created role within the IAM console. Click on the Role name.
  5. Copy the entire Role ARN and paste it into a text file. You will enter this ARN in to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



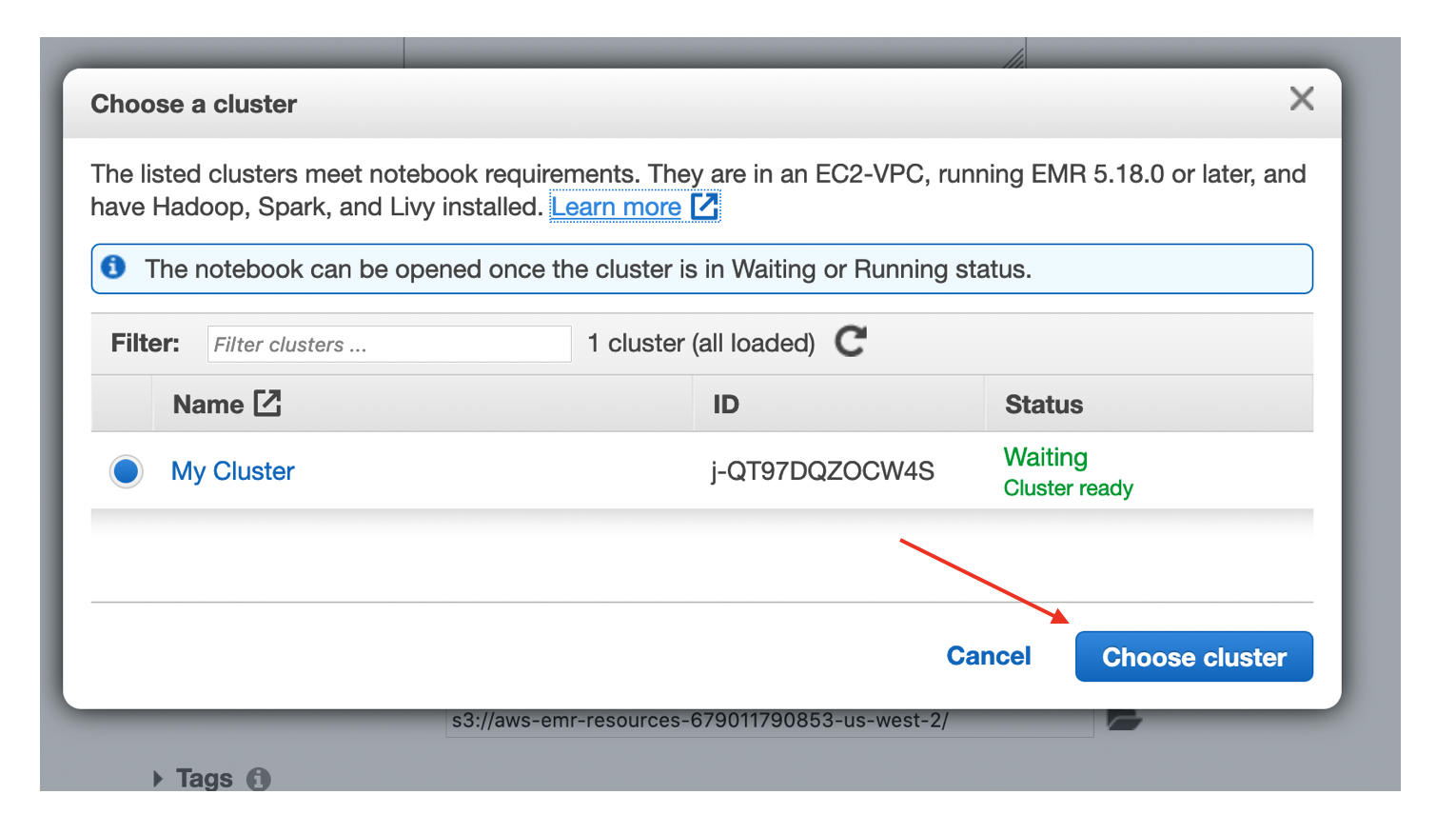
1. **Create an EMR Notebook**

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.

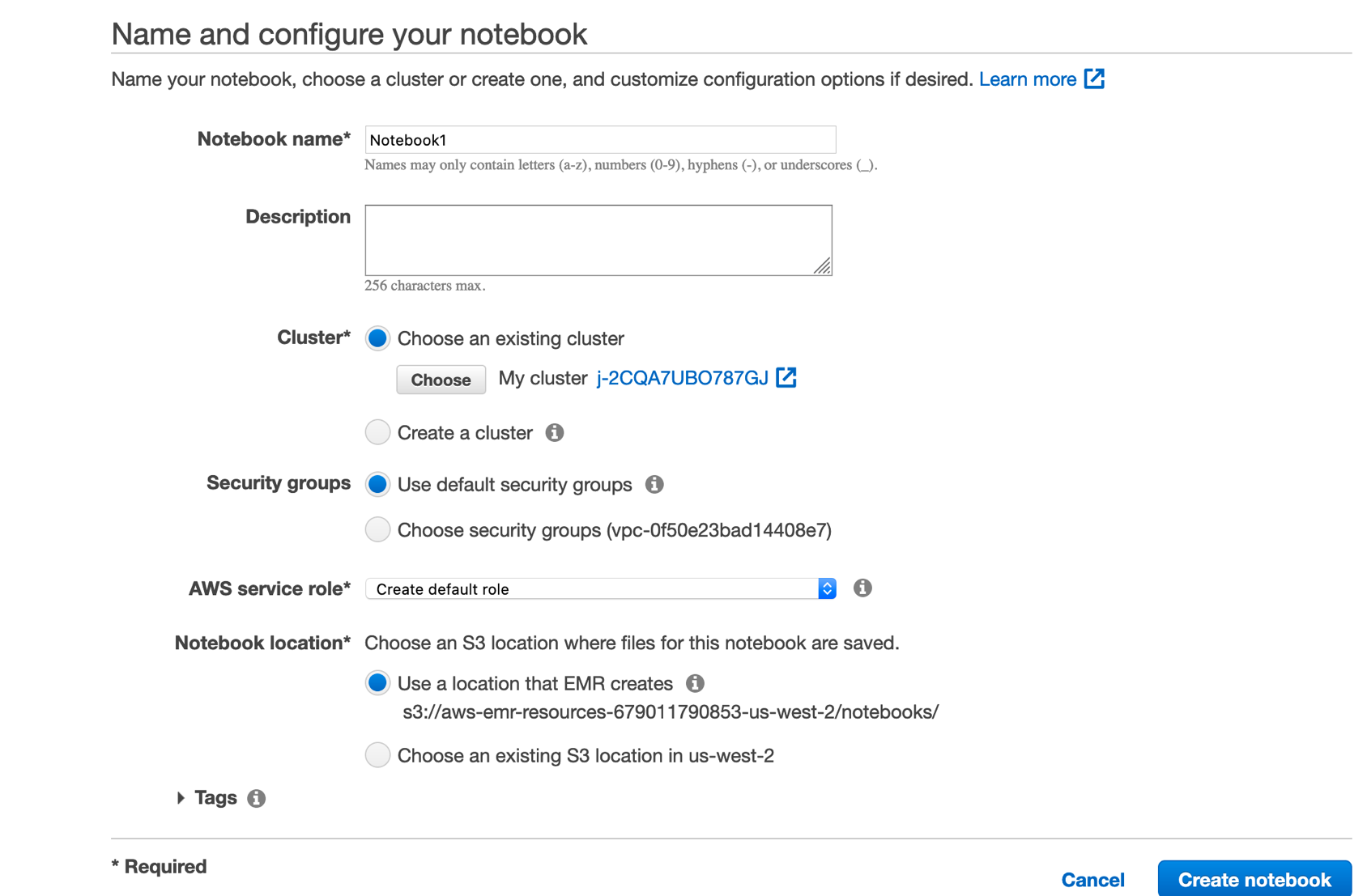
1. In the EMR console, click ‘Notebooks’.
2. Click ‘Create notebook’.



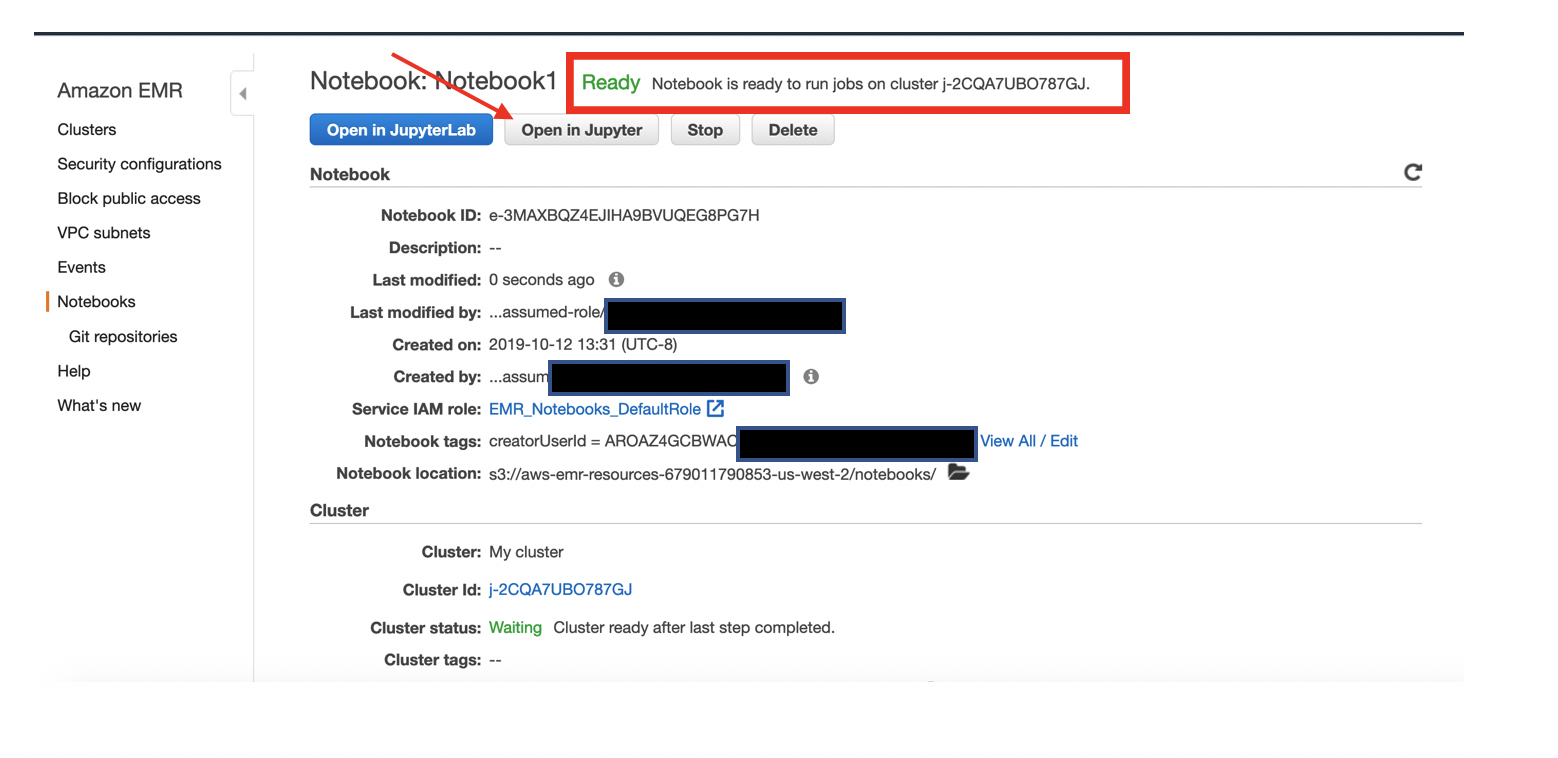
1. Name the notebook ‘Notebook1’ and add an optional description.
2. Choose an existing cluster, and click ‘Choose’.
3. Click the radio button next to the cluster you created in Lab #1 and click ‘Choose cluster’.



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’.



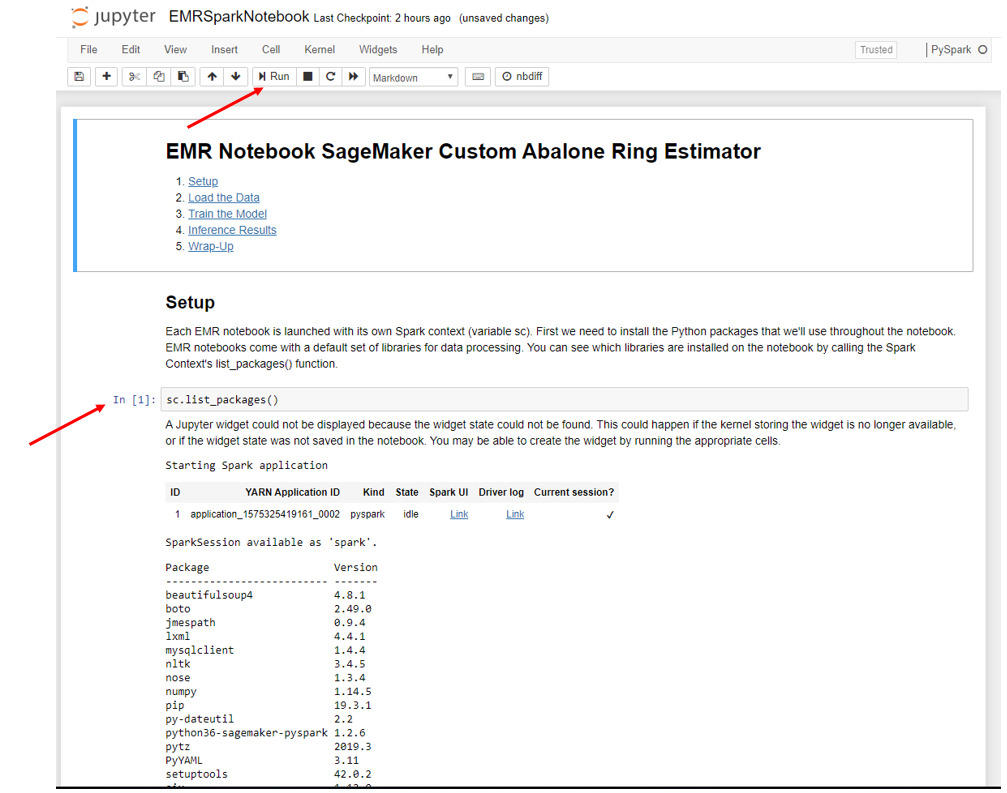
1. Click ‘Create notebook’.
2. Refresh the screen until ‘Starting’ changes to ‘Pending’ and then ‘Ready’. This usually takes about 20 seconds.
3. Click ‘Open in Jupyter’.



1. **Run the example PySpark SageMaker notebook**
   1. Download the example PySpark SageMaker notebook here:
   2. In Jupyter, click the ‘Upload’ button and choose the .ipynb file downloaded in step 3a. Click ‘Open’.



* 1. Make sure to click ‘Upload’ once more.
  2. Click on the EMRSparkNotebook.ipynb to open it.
  3. 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.
     1. 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]’).



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.**