# Using Spark ML Lib on AWS

This document outlines an investigation into using the Spark ML Lib to train and deploy machine learning models on AWS infrastructure. The goal is to show existing Spark applications and models can be migrated to AWS.

## Running Spark on AWS

Spark is a distributed compute model and requires specific underlying infrastructure to run. You can create and manage that infrastructure yourself (installing Hadoop on a cluster of EC2 instances. However, the recommended way to execute Spark jobs on AWS is with an EMR cluster. This is a managed Hadoop cluster that you can provision with the required libraries (including Spark).

The following blog demonstrates how to provision an EMR cluster and set up the permission for a Sagemaker Notebook to connect to the cluster. It will make use of a library called Livy to enable the connection between the notebook and the cluster. This allows the user to define jobs in their Notebook code and send them to the EMR cluster for execution.

<https://aws.amazon.com/blogs/machine-learning/build-amazon-sagemaker-notebooks-backed-by-spark-in-amazon-emr/>

Critical points here:

* You will need the Private IP address of the EMR cluster master node.
  + This is added to the Sparkmagic config so the Notebook can find EMR.
* You will need to allow the Notebook to communicate with EMR over port **8998**
  + Do this using security groups.
* You use Sparkmagic Kernel in Sagemaker and configure it to use the EMR master.

This will give you a kernel that can execute Spark jobs, written in either Python, R or Scala.

We have used this method to build a ML Model using Spark ML Lib in Scala. The code was taken from this blogpost (<https://towardsdatascience.com/training-your-first-classifier-with-spark-and-scala-893d7c6f7d88>) and run inside a Sagemaker Notebook with the Sparkmagic Kernel.

Here is the example Notebook:

<https://github.com/john-hawkins/amazon-sagemaker-spark/blob/main/Spark-Scala-ML-Lib.ipynb>

**Deploying Spark ML Lib Models**

The models built using the approach above can used in batch jobs on EMR. For effective online model processing we recommend deploying them to Sagemaker endpoints. To do this they need to be serialised and deployed using the MLeap library.

<https://s3-us-west-2.amazonaws.com/sparkml-mleap/0.9.6/jar/mleap_spark_assembly.jar>

This means you will need to include the MLeap library on all nodes of your EMR cluster.

The recommended way to do that is to copy the library to a private S3 bucket (called BUCKET below). Then use the bootstrap script in your EMR cluster to copy it to the required path on every node: /usr/lib/spark/jars.

You can use the following bootstrap script (Replace BUCKET with your S3 Bucket).

#!/bin/bash

# Copy the jar from an S3 bucket to all nodes of an EMR cluster.

sudo aws s3 cp s3://BUCKET/mleap\_spark\_assembly.jar /usr/lib/spark/jars/

This script also needs to be placed into an S3 bucket and used in the process of creating your EMR cluster. For additional information about installing packages onto EMR see this blog post:

<https://aws.amazon.com/premiumsupport/knowledge-center/emr-permanently-install-library/>

Additionally, there are examples of installing packages into EMR from a Notebook. However, this only works if those packages are in a Repo that the cluster can access. <https://medium.com/@msantino/how-to-install-additional-jar-packages-on-jupyter-notebook-running-on-aws-emr-4563d6eea68>

With this library in place on your cluster you can serialise Spark ML Lib models.

Here is an notebook example:

## **Using Spark from Sagemaker Studio**

In the previous example we used a classic Sagemaker Notebook to connect to EMR to run our spark jobs. In the next blog post we are shown how to connect Sagemaker Studio to an EMR cluster.

<https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-studio-notebooks-backed-by-spark-in-amazon-emr/>

There are some additional architectural extensions in this post. Firstly the EMR cluster is deployed to use Kerberos authentication, so that all interaction with the cluster from clients is secure. Secondly, the deployment has been

**CFN TEMPLATE**

https://aws-ml-blog.s3.amazonaws.com/artifacts/ml-1954/template.yaml

**Deploy with:**

https://console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new?stackName=SagemakerEMRSpark&templateURL=https://aws-ml-blog.s3.amazonaws.com/artifacts/ml-1954/template.yaml