AWS Sagemaker

Data Engineering

Exported on 07/20/2020

Table of Contents

We have used the SageMaker on the AIDA project. It was Pharma manufacturing use case.

1 Introduction

- It looks like an EC2 box that comes preconfigured with default setups of Jupyter and many different conda environments with ML packages.
- Any libraries you install on the existing conda environment would be wiped clean after the machine restart.
- Spark - you have two options for spark
    - connect to EMR
    - setup local spark
- *Another use case - a sport project in QB is using it to run a heavy machine learning model where new SM instances are spin up on demand. We never explored this as our DE and DS work was not compute intensive. You can talk to Henry / Rene re this.*

1.1 How we configured it?

1. The EBS volume requested for the machine was about 100 GB. Anything stored on this volume would be persistent and stay even when machine restarts.
    a. EBS volume directory appears as /home/ec2-user/**SageMaker**
2. The spark and conda environment was created as on any EC2 machine - they were created on the EBS volume directory.
    a. Notes for setting up conda and local spark are same as EC2 - EC2 setup[1]
    b. Create conda environment under **SageMaker** directory
    c. Create spark and hadoop directories as well under SageMaker
    d. Created 'kernel.json' for the created conda environment in step (b). This would be used for Jupyter hub. Added spark environment variables in it to make sure spark can be created in jupyter.
3. A setup shell script was created to do the following. Every time the machine reboots, it would be executed once.
    a. Change the 'ulimit' of the machine to 60K as the limit is 1K which is too low for spark usage. After this, script does soft reboot
    b. Created soft-link under conda environments for the conda environment created under /home/ec2-user/**SageMaker**
    c. Added echo command that setup environment variables for spark, java to .bashrc
    d. Creates a new kernel by copying the 'kernel.json' which has (2-b) kernel and spark variables to Jupyter hub.
    e. Added "yum install" for htop and tree - which are useful to check memory and CPU stats.

[1] https://confluence.quantumblack.com/display/~Kirtikumar.Shinde/EC2+setup

2 Issues:

1. No direct SSH: We can not do direct SSH for pycharm remote execution. You have to create the dev point through AWS Glue endpoint for enabling SSH - this is one time setup needed for SSH.
2. Machine does not shutdown automatically - so to manage cost we had to add cron utilities to shutdown machine when not in use.
3. Any works done under normal directories does not persist, so we had to be careful that all the work done is under /home/ec-user/SageMaker directory.
4. EC2 has many instance types available and all of them work. By default not all instance types are available as SageMaker - so if some of the memory optimised or CPU optimised instances would not be available for upgrade directly. You would have to work with AWS support to make them available.
5. All the users use same username - "ec2-user" - so if you have a team of people all of them share user. There might be a way to add users - but was not sure how to integrate new users with Jupyter.
   a. One option is to use different SM instances for each user, but not ideal option.
6. It is ideal solution when used for ML, but not an ideal Data Engineering solution if you are going to work with TBs of data unless you connect an EMR to the SM instance.