Lecture 8: Spark debugging, deployment and conclusion

Contents

8.1. Announcements

- Please do evaluation (10 - 15 min).

8.2. Refresher

8.2.1. Theory

- Can you list the various deployment modes available in spark?
- What is the difference between client and cluster deployment mode?
- Which deployment mode do you use to develop your program?
- spark application mainly consist of _____ and _____ .
- What are the main responsibilities of a spark Driver?
- What are various ways to interact with your EMR cluster?
- I created a 50 node Spark cluster, and my scikit-learn code doesn't speed up.
  - What do you think is going wrong?
  - What are your options to speed it up?

8.2.2. Practical - From last class DEMO

- Know how you can access web UI's in EMR?
- How to pass your spark program as part of the EMR setup?

- Do you know how to auto terminate the cluster after your job is done?
- Do you know to install the necessary packages before the cluster startup?
- Can you take your python workflow and run it as part of your EMR workflow setup?

- Can you list the various deployment modes available in spark?
  - Client
  - Cluster - this is what we are using in EMR.
  - Standalone
- What is the difference between client and cluster deployment mode?
  - Client: Driver runs on your local machine
  - Cluster: Driver runs on the cluster
- Which deployment mode do you use to develop your program?
  - Client
- spark application mainly consist of _____ and _____ .
  - Driver
  - Executors
- What are the main responsibilities of a spark Driver?
  - Driver is responsible for requesting resources (executor memory, number of executors, number of cores etc..), scheduling the tasks, and monitoring the tasks.
- What are various ways to interact with your EMR cluster?
  - Well if the question is to interact with EMR cluster to setup and make changes to an EMR cluster, then the easiest way is to use the AWS web console (the user interface). But you can also use AWS CLI.
  - if the questions is to interact with the cluster to run your spark program then you can use jupyter notebooks (or any notebooks like zeppelin, hue, etc) or you can use spark-submit command or you can use EMR steps by using EMR way to submit your spark program (like we did in last class). But for developments we use notebooks (eg: jupyter, zeppelin, hue).
- I created a 50 node Spark cluster, and my scikit-learn code doesn't speed up.
  - What do you think is going wrong?
    - The problem is that the scikit-learn code can't make use of the distributed enviornment, so it will run on a single node.
  - What are your options to speed it up?
    - You can rewrite your scikit-learn code using MLlib or explore the options on using sk-dist (the scikit-learn on spark what we mentioned in last class).

8.3. Before DEMO

You might remember I asked you to set up foxyproxy in your browser. Dynamic port forwarding and FoxyProxy are used to access EMR (Elastic MapReduce) web UIs securely over an SSH tunnel. When you create an EMR cluster, the web UIs of various components, such as Hadoop, Spark, and YARN, are available on specific cluster nodes' ports. By default, these ports are not accessible from outside the cluster for security reasons (also, you will see in the shared responsibility model that it is a managed service and AWS is responsible for security).

To access these web UIs securely from your local machine, you can set up an SSH tunnel that forwards traffic from a local port to the corresponding port on the EMR cluster's nodes. Dynamic port forwarding allows you to forward traffic to any port on the cluster's nodes dynamically.FoxyProxy is a browser extension that simplifies the process of configuring the browser to

use the SSH tunnel. It automatically switches the browser's proxy settings to use the SSH tunnel when accessing the EMR web UIs and reverts to the original settings when accessing other websites.

Using dynamic port forwarding and FoxyProxy ensures that your EMR web UI traffic is encrypted and secure, and you can access the web UIs from your local machine without exposing them to the public internet.

You can get the command for this from the AWS console. Ensure you add the rule to the master node security group to allow ssh from your IP address.

```
ssh -i ~/mds-525-001.pem -ND 8157 hadoop@ec2-34-219-14-230.us-west-2.compute.amazonaws.com
```

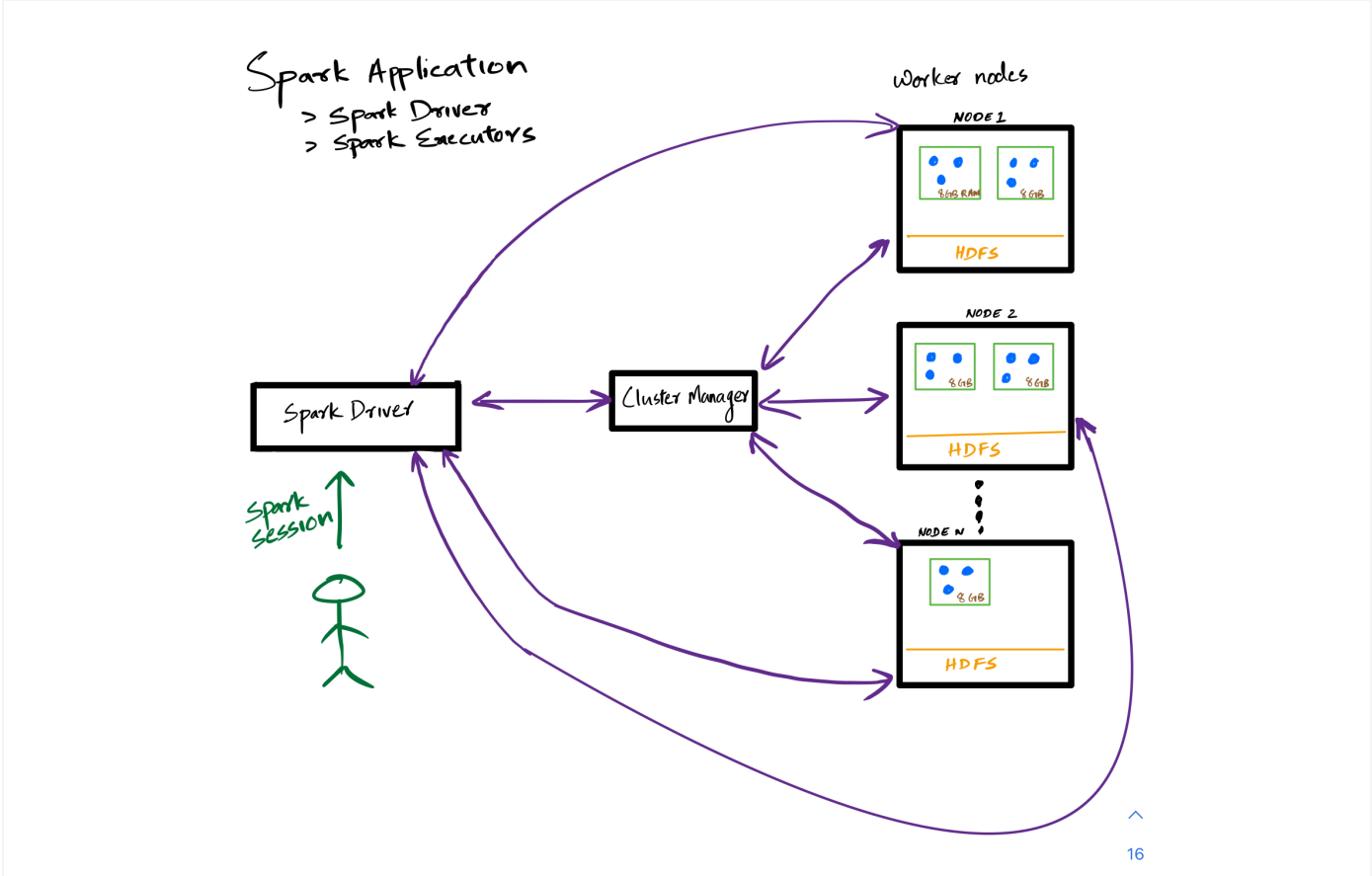
Knowing this comes handy if you want access the web UI's of the various services and needed for debugging purposes.

See also

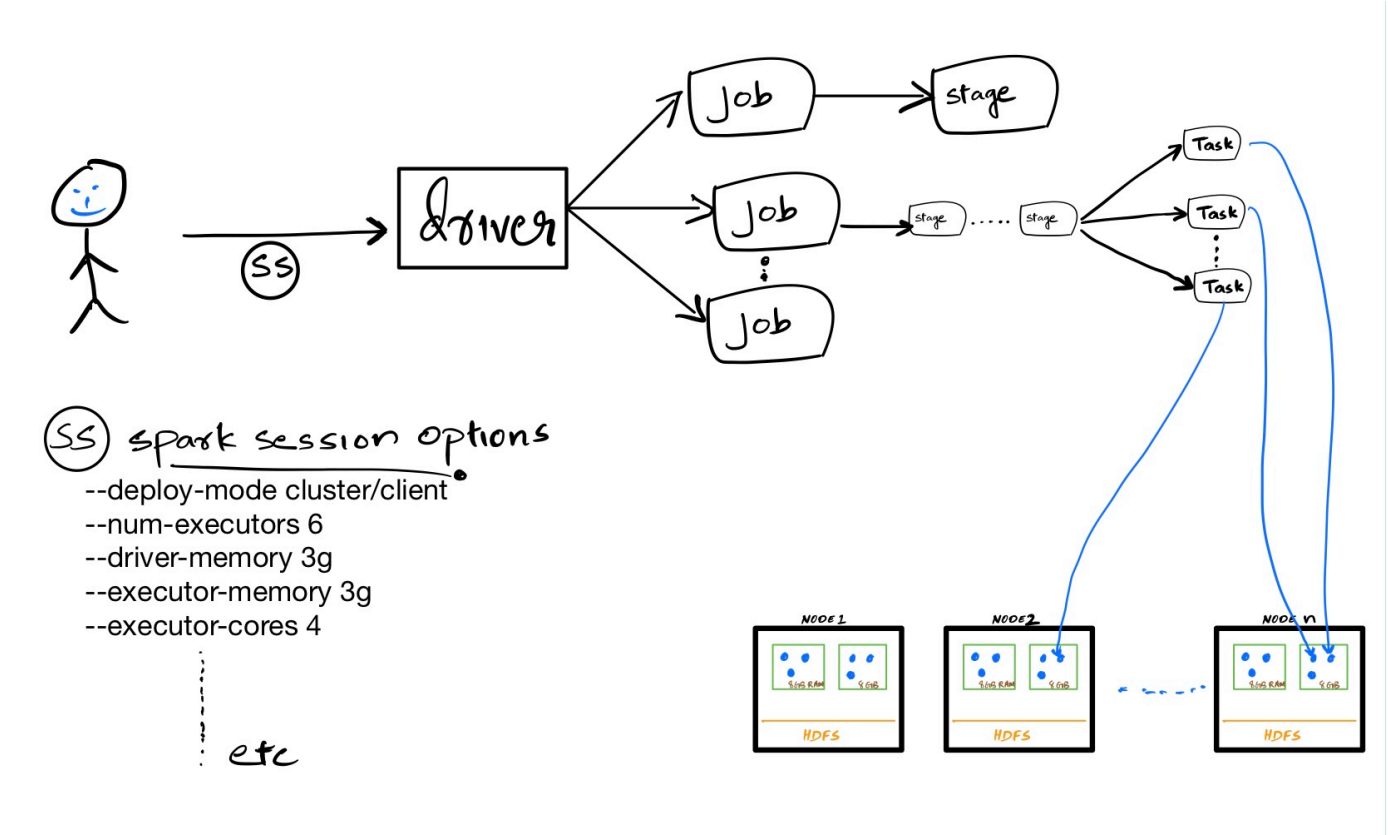
You might need to check the AWS documentation for more details.

8.4. DEMO

Let's visit behind the scenes Spark.



Adding the picture from Lecture 6. Check lecture 6 for details



User side

- Decide how you plan to interact with Spark
- If needed; Create a spark session and then pass the configurations (resources you need) and your spark program

Internals

- Creates a spark driver
- The spark driver starts its job. Here is the list...
  - Requests for resources (Resource manager)
  - The driver converts your Spark application (or your Spark program) into one or more Spark jobs.
  - Transforms each job into a DAG
  - Each Job will have multiple stages
  - Each stage will have multiple tasks
  - Each task maps to a single core and gets computed there.
  - Makes sure that all the executors are running properly.

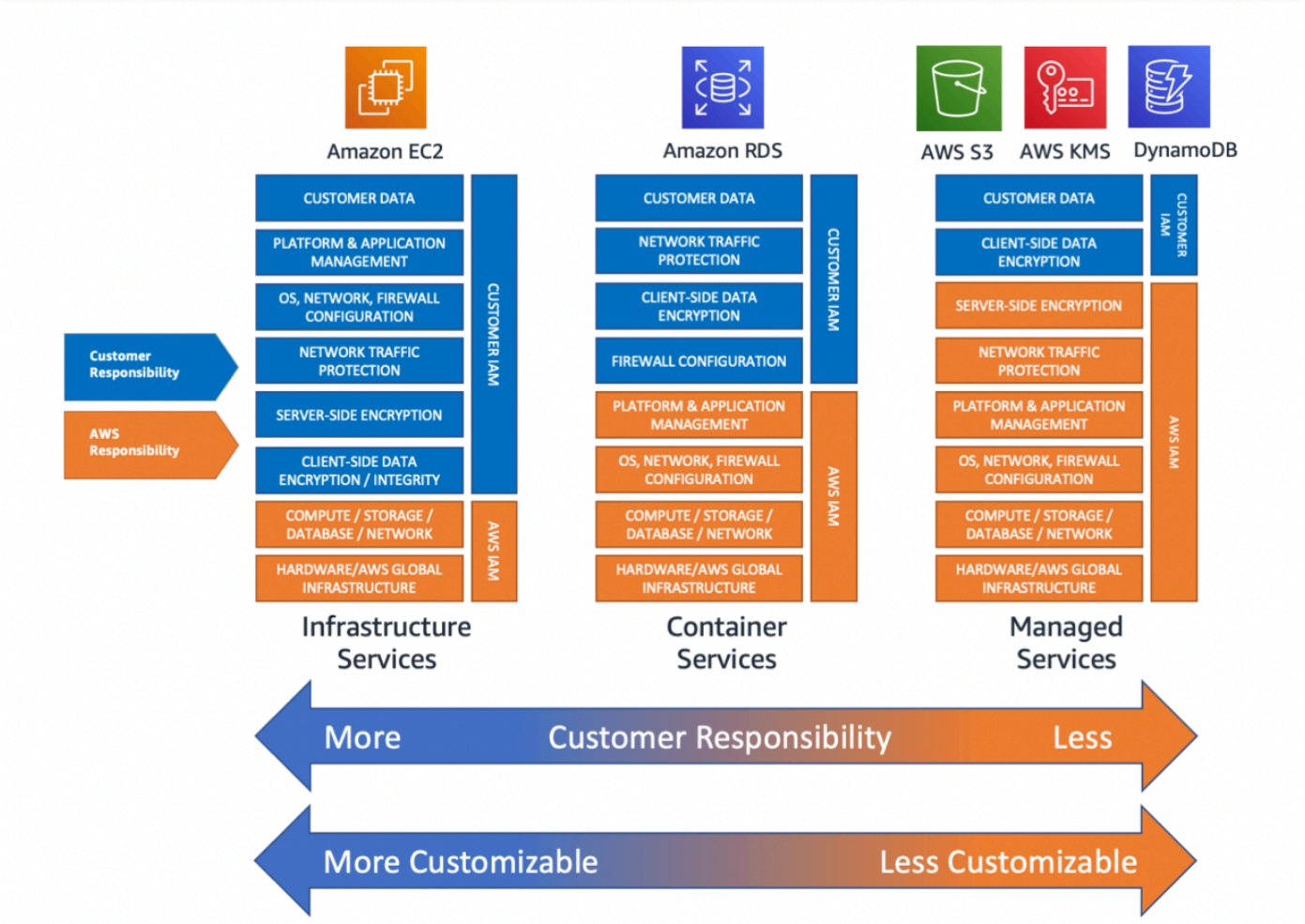
Summarizing

Here is summary of spark workflow that we explained just now.

- Submit a `spark program` along with the various `options`
  - (driver memory, executor memory, etc).
- `Driver` get the resources that you asked for
- The `driver` comes up with `multiple jobs`,
- Each `job` is made of `DAG` containing an execution plan.
- `DAG` is made of various `stages`
- `stage` consists of multiple `tasks`

8.5. Shared responsibility

The shared model provides constructive mechanisms to illustrate the separation of tasks between AWS and the customer. For example, AWS is responsible for the security and compliance of the Cloud, whereas the customer is responsible for security and compliance in the Cloud. In the below picture, we will discuss types of cloud services and compare them to each other.



See also

Here is a good article that talks about this.

8.6. Cloud services summary

Here is a table summarizing all the services we discussed in AWS and similar offerings from other cloud providers like GCP and Azure.

AWS (Amazon)	GCP (google)	Azure (Microsoft)
Virtual Private Cloud (VPC)	Google Virtual Private Cloud (VPC)	Virtual Networks (VNet)
Identity and Access Management (IAM)	Google Cloud Identity and Access Management (IAM)	Azure Active Directory
Elastic Compute Cloud (EC2)	Compute Engine	Virtual Machines
Simple Storage Service (S3)	Google Cloud Storage	Azure Blob Storage
Elastic Block Store (EBS)	Google Persistent Disks	Azure Disk Storage
S3 Glacier	Coldline	Azure Cool Blob Storage
Elastic File System (EFS)	Google Cloud Filestore	Azure Files
Amazon RDS	Cloud SQL	SQL Managed Instances
Amazon Aurora	Cloud Spanner	Azure SQL Database
Amazon DynamoDB	Cloud Bigtable	Azure Cosmos DB
Elastic MapReduce (EMR)	Dataproc	HDInsight

See also

You can check out this document from Google GCP for an entire list with comparison. Also one another document that you can check for a short description of AWS services.

8.7. Managed ML notebooks

- Sagemaker notebooks - AWS
  - How to setup sagemaker notebooks
  - How to connect to EMR
- Sagemake studio labs - A lightweight IDE for sagemaker.
- GCP Vertex AI Workbench / Google Cloud Datalab
- Azure Microsoft Azure Machine Learning Studio

8.8. Deployment

Important

Check out this conversation style notebook to help you with the topic.

See also

Tutorial by Daniel

8.9. 😁Thank you note!!😁