# Data Engineer Test

version 2020-06-07

## Engineering Basic

### OS

##### list known LINUX commands for the following purpose. Provide as much as commands to achieve the purpose.

|  |  |
| --- | --- |
| **purpose** | **commands** |
| Find which directory is using most of disk storage. | du -h / 2>/dev/null | sort -rh | head -1 |
| find a process that is listening on the port 80 | sudo lsof -n -i :80 | grep LISTEN |
| how many cpus and rams in the node | { lscpu | grep CPU\(s\)\: &  cat /proc/meminfo | grep MemTotal; } |
| kill multiple processes executed from /usr/local/bin/python3 | pkill /usr/local/bin/python3 |
| find location of the systemd service definition (file ends with .service) | find / -name '\*.service' |
| find all files ends with .py containing "http://dummy.com" | grep -rlnw '/path/to/search' -e ' http://dummy.com' | grep -e '.py$' |

##### list know GIT commands for the following purpose.

|  |  |
| --- | --- |
| purpose | commands |
| * exclude all files under /node\_modules and /vendors directory * commit changes with message "test commit" * push the changes to /feature/git101 | exclude all files under /node\_modules and /vendors directory   * { echo "/node\_modules" >> .gitignore & echo "/vendors" >> .gitignore; }   commit changes with message "test commit"   * git commit -m “test commit”   push the changes to /feature/git101   * echo "/feature" >> .gitignore * mkdir feature * mkdir feature/git101 * git init –-bare feature/git101 * git remote add origin feature/git101 * git push origin master |
| move HEAD of the current branch to upstream/develop without losing the changes made in the current branch | git stash  git checkout upstream/develop |
| Revert last 2 commits in the master branch without losing histories | git revert master~2 |
| Merge changes from upstream/master to origin/master | git checkout master  git pull upstream master  git push origin master |
| Save the current uncommitted changes in the current branch. Then restore those changes in another branch. | git checkout current\_branch  git stash  git checkout -b another\_branch  git stash pop |

##### How do you manage software on your servers? (e.g. manual, kubernetes, ansible , puppet)

|  |
| --- |
| Manual |

### Container Technology

##### What's the docker and kubernetes? And list top 5 command of docker and kubectl

|  |
| --- |
| Docker   * docker ps * docker image list * docker-compose up -d * docker-compose down -v * docker rm -f <<container\_id>>   I’ve never use the Kubectl, so the following answer is based on this [[1]](https://opensource.com/article/20/5/kubectl-cheat-sheet)  Kubectl   * kubectl get nodes * kubectl create ns <<name\_space>> * kubectl edit * kubectl delete * kubectl apply |

##### How do you deploy software in docker swarm or kubernetes cluster? Is it an automated process? How do you monitor the health of services?

|  |
| --- |
| I have never deployed the software in docker swarm or Kubernetes cluster. The following answer is summarized from [[1]](https://www.sumologic.com/glossary/docker-swarm/#:~:text=A%20Docker%20Swarm%20is%20a,join%20together%20in%20a%20cluster.&text=The%20activities%20of%20the%20cluster,are%20referred%20to%20as%20nodes.)   1. It could be an automated process if we set up the stack files for deploying to swarm properly [[2]](https://docs.docker.com/get-started/swarm-deploy/) 2. The health server can be monitored through the prometheus service [[3]](https://docs.docker.com/config/daemon/prometheus/). |

##### How do you send node and pod metrics to the metric server (prometheus, influxdb, elasticsearch)?

|  |
| --- |
| I have never sent node or pod metrics to the metric server. The following answer is summarized from   1. In short, setup the prometheus.yml and use the docker service command to create a prometheus server. [[1]](https://docs.docker.com/config/daemon/prometheus/) 2. For more detail about to use the prometheus as a monitored server please check this documentation. [[2]](https://prometheus.io/docs/introduction/overview/) |

### 

### Hadoop ecosystem

##### What are your preferred frameworks (e.g. HIve, Spark, Pig, Ooozie, Hive) to manage the ETL pipeline as well as why do you prefer them?

|  |
| --- |
| I prefer using a Spark framework, especially Pyspark, for manipulating the data in the ETL pipeline. The reason I choose because   1. It gives both `sparkSQL and `pyspark` syntax for manipulating the data, so it's easy to adapt whether you are good at python or SQL. 2. It's the first framework I have ever worked with as a data scientist, and I'm familiar with python the most, so it's the best framework for me to learn compared to others. 3. Spark is constantly updated, and the document is well written. 4. It gives you flexibility in ETL transformation, such as pivoting or loop over some variable. I found it is hard to do things in some ETL frameworks or traditional SQL. |

##### What are the use cases of HIVE external tables? What's the pros and cons of it?

|  |
| --- |
| Use cases of HIVE external table   1. When the data need to be stored in the specific location (outside of HDFS).    1. For example, I think about the highly confidential information such as PII (personal identifiable information) should be placed in where we have the highest security. It should not be placed in the HDFS where anyone in the cluster can access the data.    2. The second use cases could be when you have a configuration file that should be maintained by the business side. I think it’s hard for them to upload it directly to HDFS. Thus, we can provide them the external source such as SharePoint and just point the external table to it. 2. When you don’t want to delete the underlying data. For example, it could be a raw transaction with a massive volume, but you only use it for a few latest days. I think we can create a hive external table to point out to only latest few day.   Pros   1. The data still remains even if you drop the table from the hive command. If there are other’s table / view pointing to the same location, it should be fine. 2. You can drop the table and point to the new location if you change the underlying data location.   Cons   1. The data still remains even if you drop the table, so if it’s a confidential data it would be dangerous to leave it there. 2. There are some function that HIVE doesn’t support such as truncate, ACID, or query result caching.   To be honest, I have never used HIVE in my working life, so all the above answers are summarized from the following references   1. <https://data-flair.training/blogs/hive-internal-tables-vs-external-tables/> 2. <https://stackoverflow.com/questions/17038414/difference-between-hive-internal-tables-and-external-tables> |

##### 

##### Are there any ways to join tables between RDBMS (MySQL, Postgres) with HIVE in a single SQL statement?

|  |
| --- |
| I have only an experience with using sqoop import to import data from RDBMS to HDFS and then do the transformation (join) process onwards. I find a possible way here but it’s not a single SQL statement but it’s a sqoop statement please check it in [section 7.2.10](https://sqoop.apache.org/docs/1.4.0-incubating/SqoopUserGuide.html#id1770490)  I check that there is a way to do it with Presto as mentioned [here](https://stackoverflow.com/questions/27518828/can-i-join-mysql-with-hive)   * Presto allows querying data where it lives, including Hive, Cassandra, relational databases or even proprietary data stores. A single Presto query can combine data from multiple sources, allowing for analytics across your entire organization.   I also found the tutorial to join the data from RDBMS and HIVE [here](https://sungsoo.github.io/2015/09/19/presto-tutorial.html). In short, we use the presto connector for RDBMS and HIVE to connect with each data source then the 1 single statement can join the data together. |

##### How do you copy partitioned HIVE tables from user A and user B, when each user couldn't directly access data to each other?

|  |
| --- |
| Assume that I’m a data engineer receiving request from user A and B, (I’m not a user B trying to copy partitioned HIVE table from user A by myself).  I would   1. Create a HIVE table <<table name>> 2. Use the INSERT OVERWRITE TABLE <<table name>> PARTITION (<<partition\_column>>) SELECT \*, partition\_columns FROM <<target table name>> WHERE user IN (‘A’,’B’)   A short code snippet here from [stack overflow](https://stackoverflow.com/questions/48365816/hive-copy-from-partitioned-table) would be helpful.  When I was working in the hands-on test, I found out that the CREATE TABLE AS SELECT statement doesn’t support the partitioned table, so the above instruction would be a workaround for this purpose. |

##### What file format (ORC, Parquet, Avro etc.) do you think is the best for you? And why?

|  |
| --- |
| I usually use the csv, xlsx, and parquet format in my daily working life. The reasons are  CSV, XLSX   1. It’s easy for business side people to use it whether we give them for filling the information or receive the current report file from them. 2. It’s the file that business uses in a daily operation. I have to consume the data from it.   Parquet   1. It’s well compressed and can be partitioned by the specific column 2. It scans less data based on the partitioning, if we well specify it in the query. 3. It restores the exact data type when the file was written when we load it. No need to fix the data type after loading it into data frame. 4. It’s quite fast when loading it with pandas, spark compared to CSV, and XLSX.   Also, there is an update upon parquet which is a [delta-lake](https://delta.io/). It improves many functionalities of parquet file format.   1. One use case I used it is the UPSERT behavior. When you do the UPSERT behavior with parquet file format you have to read all the data and then select which row you need to update / insert and then write it back to the storage. With delta lake we can simplify the process and do it in a one single line. 2. It provides an ACID Transactions behavior. There is the time when I found an error in the writing operation, it reverts the change of underlying parquet file to the stage before we write it. |

##### When do you need to scale the HIVE metastore database? What's the major issue causing the workload in the metastore?

|  |
| --- |
| I have no idea about this question at all, so the answer is summarized from the following references [[1]](https://kb.databricks.com/metastore/hive-metastore-troubleshooting.html), [[2]](https://dzone.com/articles/cluster-diagnostics-troubleshoot-cluster-issues-us-2)  What’s the major issue causing the workload in the metastore.   1. When you have a connection more than 100 connections. It’s a connection limit. 2. When the data in metastore is too large and there are user using the intersive operation. It could lead to the process slower or timed out. Here is the [guide](https://dzone.com/articles/cluster-diagnostics-troubleshoot-cluster-issues-us-2) for scaling the HIVE metastore. |

### Programming Language

##### Rank and give the rating of your top 3 favorite programming languages. (5 high - 1 low)

|  |
| --- |
| Rank   1. Python (Highest preference) 2. SQL 3. JavaScript 4. R 5. MATLAB (Lowest preference) |

##### What programming language do you use to productionize models? How is it consumed by other services?

|  |
| --- |
| I use the python language to productionize the models. I had developed the model with MLflow versioning and save the best model with a pickle format. Then I use that model to score a new prediction and wrote it to csv.  The prediction result will be written to csv file with the agreed format attached with checksum file. Then the other services will grab it from the agreed location.  There is another use case where I create a table in a spark with underlying parquet file and then the other parties consumed it from the same cluster. |

## 

## Data Engineering

##### What's the difference between data lake and data warehouse?

|  |
| --- |
| The data lake concept seems to collection data for many sources together. It’s like a centralized data system ingesting every data source. There is no pre-requisite for data validation and data quality check. There is no schema or specific file format to be used. The data may be stored without any business use cases.  On the other hands, the data warehouse is well designed database with schema. The schema is well developed based on the business use cases. The objective of the data warehouse is to do an OLAP (online analytical processing), to slice and dice the data with ease. Only the target data can be loaded to the data warehouse system. |

##### What is the data pipeline and what tool do you use to build the whole pipeline?

|  |
| --- |
| The data pipeline is a step to transfer from one source to another. The components in data pipeline are   1. Connector – to connect with source data (e.g., JDBC, ODBC driver) 2. Storage Format – what kind of underlying file you would like to store (e.g., ORC, Parquet, Delta, AVRO, etc.) 3. ETL framework – the framework to extract transform and load the data (e.g., HIVE, Spark, AWS Glue, AWS Athena, etc.) 4. Orchestrator – the framework to orchestrate and connect each step together (e.g., Airflow) 5. Destination – the choice where the other parties can consume the data (e.g., S3, AWS Redshift, AWS RDS, PostgreSQL, MySQL, Cassandra, etc.)   For example, to build a hand-on test, I would use the following tools to complete the project   1. Apache spark 2. Apache parquet 3. Airflow 4. PostgreSQL 5. HIVE |

##### How do you deploy a new data flow in a different environment? How do you promote the pipeline to a higher environment with the minimal work of manual process?

|  |
| --- |
| 1. Using a docker to save the development environment and make a better deployment journey. 2. We can make a data pipeline as a docker-compose code and then running / deploying it to the docker swarm or kubenetes cluster. All the code can be placed in the github repository and use the Github CI/CD tools to minimal the manual work process. |

##### How do you monitor the quality of data produced in each stage of the pipeline?

|  |
| --- |
| We can add a data quality check function to verify the end-result of each stage in the data pipeline. The data quality check could be   1. Check that the primary key is not null 2. Check that the primary key is unique 3. Check that there is no duplicated row. 4. Check that the specific value is in the provided range.   The data quality check can be triggered by the airflow and log the result in the output. If it doesn’t meet the requirement, it’s our choice to choose how to fix it. |

##### How would you implement scalable realtime data ingestion to make it ready to use as fast as possible (< 1~2 mins)

|  |
| --- |
| I’ve never had an experience about the real time data ingestion. But to the best of my knowledge, I think we can use the [Apache Kafka](APACHE%20KAFKA) to build a data pipeline. They claimed in the page that HIGH THROUGHPUT - Deliver messages at network limited throughput using a cluster of machines with latencies as low as 2ms. |

##### How do you productionize models created by data scientists? What tool / framework you are using or recommended? How do you ensure the quality of the model before it's going to be released?

|  |
| --- |
| I recommend using [MLflow](https://mlflow.org/) . It’s a machine learning platform to manage machine learning life cycle. It helps a lot when you are developing the model. It can be logged every run of your model training. This led to a reproduceable result of the model. Once we are okay with it, we can deploy the model through the MLflow library with the [REST API endpoint](https://mlflow.org/docs/latest/models.html) (I’ve never used it before too). In the latest version of MLflow, they provide the [model registry](https://mlflow.org/docs/latest/model-registry.html) for managing the on-the-shelf model as well.  To ensure the quality of the model before it’s going to release.  For the model side,   1. We can split the data we have into a train and test set. I usually split the data in the time period manner. For example, if we have a data of 2020, we can split the training set into Jan – Sep and test on the Oct data. Then we shift the period 1 step to training from Jan – Oct then test of Nov data, so on until all data we have. 2. After that we can average the score over all folds to get the cross – validation score. 3. If the cross – validation score of the interesting metrics is met the requirement and it is consistent overtime, then I think it’s okay to release the model. Otherwise, the data scientist needs to retrain the model and find the sweet spot for retraining the model with the additional data (more data or shifting period of data). 4. When the model is on the shelf, we need to monitor the indicator overtime. To check that there is any diff in the metrics score.   For the input data side   1. We should profile the input data for the model and set the default statistical data range. When the new data come into the system, we need to check the basic statistics against the default data range to see whether there is any data drift before coming into the model. If it’s an importance feature it could lead the prediction to the other ways.   For the concept drift   1. We need to limit the training data set to reflect the change behavior of the customer. For example, we may change from 1-year historical training set into 3-month historical training set so that the data can reflect and capture the customer behavior changes as soon as possible.   Here is a [source of machine learning drift](https://databricks.com/blog/2019/09/18/productionizing-machine-learning-from-deployment-to-drift-detection.html) I think it is good to read. |

##### How do you manage secrets (password, credential) from being seen by people outside your team?

|  |
| --- |
| I usually put the secrets in the configuration file for the local development. For the enterprise implementation I saw that they used the KMS, IAM role and service to manage the access to each resource (I have never configured it myself).   1. Put it in the configuration file and keep it private. 2. Hash the secrets and keep the encryption key within team. 3. Use the IAM service and role provided by the cloud infrastructure (if we use any of them). |

##### What would you do if someone asked you to add / encrypt phone numbers to the data warehouse?

|  |
| --- |
| I would check with the company policy about the how to deal with the PII information even if it’s encrypted. If there is a way to do it, I will process as per the procedure. Otherwise, I will let them know that we can’t provide it in the data warehouse due to the legal issues.  By the way, I don’t understand why they need the phone number or even encrypted phone number in the data warehouse because it can’t be used in any data analytics use cases. And, if we hash them with salt, it will become a single unique key which is useless for any analytics use cases as well. I should discuss with them about their objective before doing any further action.  But if we have to store it, I think that we can treat it like we store the password. We can hash the phone number with the hash function and salt value. ([ref](https://www.patanasongsivilai.com/blog/password-hash-function-salt-values/)) |

#### 

##### What tools do you use for visualize and analyze data you have for each team below

|  |  |
| --- | --- |
| **team** | **tools** |
| Data Scientists | Matplotlib, Seaborn, PowerBI, Pandas, Spark |
| Data Analyst | Matplotlib, Seaborn, PowerBI, Pandas, Spark |
| Marketing | Excel, Power point, PowerBI, Pandas |
| Service Operations | Excel, Power point, PowerBI, Pandas |
| Higher/Middle management | PowerBI, Power point, Pandas |
| Software Development | Matplotlib, Seaborn, PowerBI, Pandas, Spark |

## 

## ML Engineering

##### What are your favorite use cases of machine learning models?

|  |
| --- |
| My favorite use case of machine learning model is a propensity model. Throughout my working life, the propensity model is the most benefit in term of revenue generating and impact with the business the most. The other models are good as well, but it wasn’t success like a propensity model. |

##### 

##### How would you design and implement the real time fraud detection system for the banking payment service? What algorithms are you going to use in the model and why?

|  |
| --- |
| I would go for a classification model with tree-based such as XGBoost or LightGBM. The reasons are the model is small and it’s fast in inferencing the prediction. The fraud detection is a use case that need a very few time-to-actions. So, we have to use the light-weight model.  In addition, the tree-based model is a non-linear model which is robust to the outlier. When come to the fraud application, I think that most of the data when there is fraud would have a outlier pattern compared to normal transaction. Also, we can understand how the model differentiate between fraud and non-fraud transaction because it’s a tree-based model. We can explain it with [SHAP](https://github.com/slundberg/shap).  The model can have 2 granularity level, the first one is in the transaction level whether it’s fraud or not. The second could be in a customer level when we aggregate the data together.  The fraud detection is a well known about the imbalanced in the target data (very few positive target) compared to negative target. We can solve this problem with up sampling, down sampling when training the model. Also, we should [calibrate the prediction probability](https://scikit-learn.org/stable/modules/calibration.html) after modeling with the actual fraud distribution. If we still face the problem with false positive, we can adjust the probability threshold later for alarming the fraud signal.  We can deploy the model with the MLFlow REST API. When the banking payment service call an API with the transaction data it can return the prediction result within a short period of time.  For the customer level the prediction can be pre-calculated at the end of the day and store within the database so that the API can look up and return result based on the customer id. |

##### How would you design and implement the food recommendation system for the food delivery service? What algorithms are you going to use in the model and why?

|  |
| --- |
| For the food recommendation system, the basic approaches would be a collaborative filtering or content-based filtering. It uses the past behavior for the customer, content to recommend what the user would likely to like based on the personal or content features. But to implement the food recommendation here I would propose a hierarchical-multilabel classification model. The concept is to separate the food into each granularity level. For example, the food category (noodles, rice, fast food, etc.), and the food-subcategory such as (noodels – yentaofo, natural, namtok, etc.). We will make a multi-label classification model for each of the positive label as a food category, food\_subcategory they bought in the past.  The feature could range from anything related to the customer, delivery, or food domains. Then we will have a prediction whether category, sub-category we should recommend to each user. We can map recommended sub-category to the highest ranking recommend merchant at that time.  The reason we don’t make the model for each merchant granularity because the number of positive labels would be very few for each one. And it’s hard for model to learn from this.  The algorithms could be a tree-based model like the previous example, XGBoost, LightGBM, your choices. The reason is it’s simple and it’s a structured data set. If I have a limit time to develop the model I would go for a tree-based model first. After that, we can try using the deep learning model for other approaches later.  The result could be produced and save to the database. For each user when they login to the application, we can look up those table and return the prediction result within a short time.  I purpose to make a batch prediction in a monthly or weekly manner, because I have the assumption that the customer preference doesn’t change so quickly that we have to make a real time prediction. Also, the business rule could be applied after we have the prediction result such as to add the ads merchant to the recommendation. |

## 

## Hands on test

### Problem Description

#### Source Data to process

**1. order\_detail.csv**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Note** |
| order\_created\_timestamp | timestamp | format YYYY-MM-DD HH:MM:SS |
| status | string |  |
| price | integer |  |
| discount | float |  |
| id | string |  |
| driver\_id | string |  |
| user\_id | string |  |
| restaurant\_id | string |  |

**2. restaurant\_detail.csv**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Name** | **Type** | **Note** |
| id | string |  |
| restaurant\_name | string |  |
| category | string |  |
| esimated\_cooking\_time | float |  |
| latitude | float |  |
| longitude | float |  |

#### 

#### Business requirements

* Create two tables in postgres database with the above given column types.
  + order\_detail table using **order\_detail.csv**
  + restaurant\_detail table using **restaurant\_detail.csv**
* Once we have these two tables in postgres DB, ETL the same tables to Hive with the same names and corresponding Hive data type using the below guidelines
  + Both the tables should be **external table**
  + Both the tables should have **parquet file format**
  + restaurant\_detail table should be partitioned by a column name **dt** (type string) with a static value **latest**
  + order\_detail table should be partitioned by a column named **dt** (type string) extracted from **order\_created\_timestamp** in the format **YYYYMMDD**

#### Example of dt column

order\_created\_timestamp: "2019-06-08 17:31:57"

dt: "20190608"

* After creating the above tables in Hive, create two new tables \_\_order\_detail\_new\_\_ and \_\_restaurant\_detail\_new\_\_ with their respective columns and partitions and add one new column for each table as explained below.

|  |  |  |
| --- | --- | --- |
| **Table Name** | **New Column Name** | **Logic** |
| order\_detail | discount\_no\_null | replace all the NULL values of discount column with 0 |
| restaurant\_detail | cooking\_bin | using esimated\_cooking\_time column and the below logic |

|  |  |
| --- | --- |
| **esimated\_cooking\_time** | **cooking\_bin** |
| 10-40 | 1 |
| 41-80 | 2 |
| 81-120 | 3 |
| greater than 120 | 4 |

Final column count of each table (including partition column):

1. order\_detail = 9

2. restaurant\_detail = 7

3. order\_detail\_new = 10

4. restaurant\_detail\_new = 8

#### SQL requirements

* Get the average discount for each category
* Row count per each cooking\_bin

#### CSV output requirements

Save the above query output to CSV files name discount.csv and cooking.csv.

#### Technical Requirements

* Use Apache Spark, Apache Sqoop or any other big data frameworks
* Use a scheduler tool to run the pipeline daily. Airflow is preferred
* Include a README file that explains how we can deploy your code
* (bonus) Use Docker or Kubernetes for up-and-running program

Question output

1. Source code
2. Docker, docker-compose, kubernetes files if possible.
3. README of how to test / run