

Date	25/10/2023
Project name	BIG DATA ANALYSIS WITH IBM CLOUD DATABASE

Phase 3

BIG DATA ANALYSIS WITH IBM CLOUD DATABASES

1. IBM Cloud Account Setup:

- If you don't have one, create an IBM Cloud account.
- Subscribe to the necessary services like IBM Cloudant, IBM Cloud Object Storage, IBM Cloud Data Engine, and IBM Watson Studio.

2. Data Ingestion:

- Ingest data from various sources (e.g., sensors, social media, customer databases).
- Implement data pipelines and connectors to extract data into a format suitable for analysis.
- Consider using IBM Cloud Data Integration for data ingestion and transformation tasks.

3. Data Storage:

- Set up IBM Cloud Object Storage for storing ingested data.
- Organize data into buckets and define access controls.
- Create a data storage strategy that considers data retention and archiving requirements.

4. Data Processing:

- Utilize IBM Cloud Data Engine (perhaps on Apache Spark) to process and analyze the stored data.
- Develop data processing pipelines and scripts for extracting valuable insights.
- Optimize the data processing workflows for efficiency and scalability.

5. Machine Learning:

- Use IBM Watson Studio to build, train, and deploy machine learning models.
- Select and implement machine learning algorithms for predictive analytics and pattern recognition.
- Monitor model performance and retrain models as needed.

6. Data Visualization:

- Integrate data visualization tools like Tableau or Power BI.
- Create interactive and informative dashboards to visualize insights.
- Ensure that visualization tools can connect to data stored in IBM Cloud databases.

7. Infrastructure Setup:

- Configure and optimize IBM Cloud Foundry environments for hosting applications, if needed.
- Ensure that the infrastructure can handle the demands of data processing and visualization.

8. Data Governance and Compliance:

- Implement data governance policies and access controls to ensure data security and compliance.
- Document data lineage and data usage for auditing and compliance purposes.

9. Scalability and Performance:

- Monitor the performance of your Cloud Foundry applications and data processing pipelines.
- Implement auto-scaling and load balancing as needed to handle fluctuations in data volume.

DATASET

<https://www.dofactory.com/sql/download-sample-database>

SOURCE CODE

```
import os
import ibm_boto3
from ibm_botocore.client import Config
from cloudant.client import Cloudant

# IBM Cloud Object Storage credentials
cos_credentials = {
    'api_key': os.environ.get('COS_API_KEY'),
    'service_instance_id': os.environ.get('COS_SERVICE_INSTANCE_ID'),
    'endpoint_url': os.environ.get('COS_ENDPOINT_URL'),
}

# IBM Cloudant credentials
cloudant_credentials = {
    'username': os.environ.get('CLOUDANT_USERNAME'),
    'password': os.environ.get('CLOUDANT_PASSWORD'),
    'host': os.environ.get('CLOUDANT_HOST'),
    'port': '443',
    'url': 'https://' + os.environ.get('CLOUDANT_HOST')
}

# Initialize IBM Cloud Object Storage client
cos_client = ibm_boto3.client('s3',
                              ibm_api_key_id=cos_credentials['api_key'],
                              ibm_service_instance_id=cos_credentials['service_instance_id'],
                              config=Config(signature_version='oauth'),
                              endpoint_url=cos_credentials['endpoint_url'])

# Initialize Cloudant client
cloudant_client = Cloudant(cloudant_credentials['username'],
                           cloudant_credentials['password'],
                           url=cloudant_credentials['url'], connect=True)

# Function to upload customer data to IBM COS
def upload_customer_data_to_cos(data, bucket_name, object_name):
```

```

cos_client.upload_fileobj(data, bucket_name, object_name)

# Function to store references to customer data in Cloudant
def store_customer_reference_in_cloudant(customer_id, bucket_name,
object_name):
    database = cloudant_client['customer_data']
    # Create a document in Cloudant that stores references to the customer
data in COS
if __name__ == '__main__':
    # Simulate customer data
    customer_data = {
        'customer_id': 'customer123',
        'name': 'Alice Johnson',
        'email': 'alice@email.com',
        'phone': '555-555-5555',
        'address': '789 Oak St, Village, Country',
    }

    # Upload customer data to IBM COS
    upload_customer_data_to_cos(str(customer_data), 'customer-bucket',
'customer123.json')

    # Store references in Cloudant
    store_customer_reference_in_cloudant('customer123', 'customer-bucket',
'customer123.json')

```

Conclusion

IBM Cloud databases offer a variety of services that can be used to build a complete big data analysis solution. By combining these services with other tools, such as data visualization tools and machine learning tools, businesses can gain valuable insights from their data.