**General Questions**

1. **Can you walk us through your experience with designing and implementing data pipelines in AWS?  
   *(This relates to your experience in the Spam Calls Data Processing Pipeline.)***
2. **How do you approach problem-solving when faced with complex data engineering challenges?  
   *(Your CV highlights problem-solving skills. You can elaborate on situations where you solved a difficult problem.)***
3. **Can you describe the differences between using AWS Glue and Apache Airflow for orchestrating ETL jobs? When would you choose one over the other?  
   *(Since you've worked with both, they may ask for practical examples of their use cases.)***
4. **How do you ensure data quality in your pipelines, and what tools have you used to monitor data integrity?  
   *(Mention how you used AWS CloudWatch and SNS in your project.)***

**SQL and Database Questions**

1. **Can you explain how you optimized SQL queries in Amazon Athena for reporting and analysis?  
   *(Your experience writing and optimizing complex SQL queries will be relevant here.)***
2. **What are the trade-offs between using a relational database like MySQL versus a NoSQL database like DynamoDB for storing large datasets?  
   *(This will test your understanding of database systems based on your listed experience.)***
3. **What techniques do you use for partitioning large datasets in Redshift, and why is partitioning important?**
4. **How do you handle schema changes or evolving data structures in your ETL pipeline?  
   *(This is important for evolving data sources, and you can reference schema management in AWS services.)***
   1. **Schema on read**
   2. **Adding new fields**
   3. **Keep versions of schemas**

**Python and Spark Questions**

1. **How have you used Python for data processing in your ETL workflows? Can you share an example where Python helped solve a specific data engineering challenge?  
   *(Discuss your Python experience in the Spam Calls Data Processing Pipeline.)***
2. **Can you explain how Spark’s data partitioning works and how you’ve optimized Spark jobs in your past projects?  
   *(This can be related to your experience with Spark in the Spam Calls project.)***
3. **How do you handle large-scale data processing with Spark, and what strategies have you used for optimizing Spark job performance?**

**AWS-Specific Questions**

1. **Can you explain the role of AWS Lambda in your data pipeline, and how did it help in automating the workflow?  
   *(Refer to your implementation of Lambda functions for triggering DAGs and Glue jobs.)***
2. **How would you handle a situation where AWS S3 files are arriving late or out of order? What impact would that have on your pipeline, and how would you mitigate it?**
3. **How do you ensure scalability in your ETL pipeline when processing growing datasets in AWS Glue?**
4. **Can you explain a situation where you used AWS Step Functions in a pipeline and why you chose them over other orchestration tools?**

**Machine Learning and Data Science (Relevant to Data Engineering)**

1. **In the E-commerce Platform Sales Analytics project, how did you handle missing data or outliers before applying Facebook’s Prophet model for forecasting?  
   *(This will touch on your data preprocessing skills.)***
2. **What considerations do you take into account when working with time-series data, and how does this affect your data engineering pipeline?  
   *(Mention your work with time-series forecasting and cross-validation techniques.)***
3. **How did you optimize the performance of your forecasting models in terms of accuracy and efficiency?  
   *(Discuss cross-validation and hyperparameter tuning for the Prophet model.)***
4. **How did you integrate your forecasted results into the dashboards for business insights? Can you describe the tools you used to achieve this?  
   *(Talk about your use of HoloViz Panel or QuickSight for creating interactive visualizations.)***

**Big Data and Tools**

1. **Can you explain how you’ve used Docker in your projects? How does containerization benefit data engineering workflows?  
   *(You’ve listed Docker in your skills section, so they may ask about containerization.)***
2. **What are some best practices you follow when setting up and managing data storage in AWS S3?**
3. **How do you handle large file sizes and optimize read/write operations in a cloud environment like AWS?**

**Scenario-Based Questions**

1. **Imagine you’re receiving streaming data from various APIs. How would you design a scalable architecture to ingest, store, and process this data in near real-time?  
   *(This question can relate to your experience in the Spam Calls project.)***
2. **How would you design a system to handle both batch and real-time data processing for an e-commerce platform's sales data? What AWS services would you use, and why?**
3. **How do you handle failure in your data pipeline? What strategies or tools do you use to ensure data pipeline resiliency and fault tolerance?**

**General Questions**

1. **Can you walk us through your experience with designing and implementing data pipelines in AWS?**
   * I have designed and implemented several data pipelines using AWS services like Glue, Lambda, and S3. For example, in the Spam Calls Data Processing Pipeline project, I built a robust pipeline to ingest data from multiple APIs, process it, and store the results in S3. I also used Lambda to trigger different components of the pipeline based on data size, ensuring efficiency and scalability.
2. **How do you approach problem-solving when faced with complex data engineering challenges?**
   * My approach starts with understanding the problem thoroughly by breaking it down into smaller parts. I then analyze the root cause and explore various solutions using both best practices and innovative approaches. I leverage tools like Spark for large-scale data processing or Python for scripting automation. When necessary, I collaborate with teammates to ensure that I’m considering different perspectives before finalizing a solution.
3. **Can you describe the differences between using AWS Glue and Apache Airflow for orchestrating ETL jobs? When would you choose one over the other?**
   * AWS Glue is a fully managed service focused on ETL jobs and data transformations, which is very useful for simpler or smaller-scale ETL pipelines. Airflow, on the other hand, provides much more flexibility and control over complex workflows. I choose Glue when I need a straightforward, scalable ETL solution integrated into AWS, and I use Airflow when I require more advanced task orchestration or need to integrate different types of jobs (e.g., Spark, Python scripts, external APIs).
4. **How do you ensure data quality in your pipelines, and what tools have you used to monitor data integrity?**
   * I use a combination of built-in tools like AWS CloudWatch for monitoring and SNS for sending alerts when data quality thresholds are breached. I also apply validation steps within the ETL pipeline, like checking for missing or malformed data and using assertions to validate business rules. Continuous monitoring ensures pipeline integrity, and if issues arise, they are addressed immediately.

**SQL and Database Questions**

1. **Can you explain how you optimized SQL queries in Amazon Athena for reporting and analysis?**
   * I optimize SQL queries by focusing on partitioning and filtering data early in the query, reducing the volume of data being processed. I also use Athena's support for columnar data formats like Parquet, which significantly reduces data retrieval times. Additionally, I ensure that only necessary fields are selected and avoid expensive operations such as large joins or aggregations on unindexed fields.
2. **What are the trade-offs between using a relational database like MySQL versus a NoSQL database like DynamoDB for storing large datasets?**
   * Relational databases like MySQL are excellent for structured data and complex queries with joins, whereas DynamoDB is better suited for high-velocity, large-scale, unstructured data with simple key-value access patterns. The trade-off lies in the need for complex querying vs. high scalability and performance under heavy load. I choose DynamoDB when scalability is a priority, and MySQL for structured, transactional data with complex relationships.
3. **What techniques do you use for partitioning large datasets in Redshift, and why is partitioning important?**
   * I use techniques like distribution keys and sort keys to partition data in Redshift, which improves query performance by localizing data access patterns. Partitioning reduces the amount of data scanned during queries, leading to faster execution times. For instance, partitioning based on date ranges helps limit queries to relevant time windows.
4. **How do you handle schema changes or evolving data structures in your ETL pipeline?**
   * To handle schema changes, I version my ETL processes, ensuring backward compatibility where possible. I also adopt schema-on-read approaches, especially with semi-structured data, so that schema evolution doesn’t disrupt existing pipelines. When major changes occur, I implement data validation steps and alert monitoring to catch any issues early on.

**Python and Spark Questions**

1. **How have you used Python for data processing in your ETL workflows? Can you share an example where Python helped solve a specific data engineering challenge?**
   * Python is my go-to language for handling data manipulation, cleaning, and automation in ETL workflows. For example, in the Spam Calls project, I wrote Python scripts to interface with external APIs, process JSON responses, and convert the data into a format suitable for downstream analysis. Python’s versatility allowed me to automate the entire process, improving efficiency and reducing errors.
2. **Can you explain how Spark’s data partitioning works and how you’ve optimized Spark jobs in your past projects?**
   * Spark’s data partitioning divides the data across different nodes in the cluster, allowing parallel processing. I optimize Spark jobs by ensuring that data is evenly distributed across partitions and minimizing data shuffling between nodes, which can slow down performance. I also tune settings like executor memory and adjust partition sizes to align with the dataset and cluster configuration.
3. **How do you handle large-scale data processing with Spark, and what strategies have you used for optimizing Spark job performance?**
   * For large-scale data processing, I focus on minimizing data shuffling by using narrow transformations like map and filter where possible. I also cache intermediate datasets when they are reused in multiple stages of a job, reducing redundant computations. Additionally, I optimize Spark configurations like memory management and executor settings based on the specific workload.

**AWS-Specific Questions**

1. **Can you explain the role of AWS Lambda in your data pipeline, and how did it help in automating the workflow?**
   * In the Spam Calls project, I used AWS Lambda to automatically trigger Airflow DAGs when certain conditions were met, such as a specific file size in S3. This automation reduced the need for manual intervention and ensured that the pipeline remained efficient and scalable. Lambda also helped in connecting different services like Glue and S3 seamlessly.
2. **How would you handle a situation where AWS S3 files are arriving late or out of order? What impact would that have on your pipeline, and how would you mitigate it?**
   * I would use a timestamp-based processing approach to ensure data is processed in the correct order. Late-arriving files could be handled using event-driven triggers in S3, combined with a buffering mechanism to wait for a complete batch of files before starting the next ETL process. If necessary, I would set up alerts to notify me of such delays.
3. **How do you ensure scalability in your ETL pipeline when processing growing datasets in AWS Glue?**
   * I ensure scalability by using partitioning in the data storage layer (S3 or Redshift), using Glue’s dynamic frame-based processing for large datasets, and adjusting worker configurations in Glue to handle increasing data volumes. I also design pipelines to be modular and scalable by breaking them into smaller, reusable components.
4. **Can you explain a situation where you used AWS Step Functions in a pipeline and why you chose them over other orchestration tools?**
   * In the Spam Calls project, I use d AWS Step Functions to manage the orchestration of various AWS services (Lambda, Glue, S3) because Step Functions provide a clear, visual workflow and allow fine-grained control over retries, error handling, and conditional branching. This was essential for ensuring the reliability of the pipeline, especially when dealing with different API response times.

**Machine Learning and Data Science**

1. **In the E-commerce Platform Sales Analytics project, how did you handle missing data or outliers before applying Facebook’s Prophet model for forecasting?**
   * I handled missing data by using interpolation methods for small gaps and more sophisticated imputation techniques when needed. For outliers, I applied z-score thresholds to identify and remove anomalies, ensuring that the Prophet model wasn't skewed by extreme values. Prophet also has built-in mechanisms to handle missing data effectively, which helped streamline the process.
2. **What considerations do you take into account when working with time-series data, and how does this affect your data engineering pipeline?**
   * When working with time-series data, I ensure that the data is consistently formatted, free from missing timestamps, and properly resampled if necessary. I also account for seasonality and trend components, as these factors affect how data should be preprocessed and modeled. In the pipeline, I ensure data arrives in sequential order to avoid inaccurate forecasts.
3. **How did you optimize the performance of your forecasting models in terms of accuracy and efficiency?**
   * I optimized model performance by tuning hyperparameters through cross-validation and using Prophet's built-in seasonalities to capture regular patterns. I also employed early stopping techniques during model training to avoid overfitting and used error metrics like MAE and RMSE to select the best model configuration.
4. **How did you integrate your forecasted results into the dashboards for business insights? Can you describe the tools you used to achieve this?**
   * I integrated forecast results into dashboards by using HoloViz Panel for real-time, interactive visualizations. The data from the Prophet model was transformed into business insights like future sales predictions, inventory recommendations, and marketing strategies. This allowed decision-makers to explore different scenarios and make data-driven decisions.

**Big Data and Tools**

1. **Can you explain how you’ve used Docker in your projects? How does containerization benefit data engineering workflows?**
   * I’ve used Docker to containerize Python scripts and Spark jobs, making it easy to replicate environments across different machines. Containerization ensures that dependencies are consistent, making it easier to deploy and scale data engineering workflows across different cloud environments without worrying about compatibility issues.
2. **What are some best practices you follow when setting up and managing data storage in AWS S3?**
   * I follow practices such as organizing data into meaningful, hierarchical folder structures, enabling S3 versioning for backup purposes, and using lifecycle policies to transition infrequently accessed data to Glacier. I also ensure proper encryption and access control with IAM policies to maintain security and compliance.