Staff DataOps Engineer – Panel Interview Instructions

For the panel round of interviews, we would like you to:

1. **Read a bit more about Lore** (see, but do not share, "Resilience and Lore" at https://careers.lore.co/resilience-lore- password: lore-careers).
2. **Read a bit about the people** you will be interviewing with (see below)
3. **Select one of the problem statements** that resonates with you and that you would like to discuss. Let me know which one you select!
4. **Prepare a working prototype** to illustrate your approach to solving the chosen problem. You can also use diagrams, flowcharts, bullet points, or any other visual or textual representation that helps illustrate your thought process.
5. **Be prepared to discuss your approach** in detail during each interview, including the rationale behind your choices, any potential challenges you foresee, and how you would address them.
6. Read **Working at Lore** on https://careers.lore.co/

Interviewers

You'll be meeting with a selection of three of these folks

Adam Ameele is a health service psychologist focused on helping build explorations in Lore to create ongoing resilience-enhancing conversations with users. These dynamic, personalized, and authentic dialogues lead to insights that grow agency and action-oriented mindsets in users.

Brian Kerschner is a Data Engineer working on our Explorations product. With a background in designing data architecture to uncover decision-making patterns, he's passionate about building systems that help users surface meaningful insights—enabling them to rediscover and build on their own thinking over time.

Daniel Kunnath is a software engineer and technical leader at Lore. He blends hands-on engineering with strategic leadership and loves tackling complex problems head-on. At Lore, he's working on high-leverage infrastructure and automation to accelerate delivery and empower teams, all with a focus on precision, security, and user privacy.

Danielle Martin is a writer and prompt engineering expert at Lore. She's widely considered the conscience of our generative AI bot and is relentless in her efforts to get each of us to communicate more clearly.

Edward Sarker is a staff software engineer at Lore, with extensive experience in infrastructure, AI, and multi-cloud environments. He leverages his passion for complex challenges and cutting-edge development to architect innovative systems that create powerful automation and intelligence.

Jessyca Duerr is a storytelling expert at Lore. Recently she's been engaged in ensuring the "why" behind Lore is maximally engaging and clear to our employees, investors, and external partners.

Jonathon Gaff is a Data Engineer working on our Generative AI agent, LoreBot. He's been instrumental in imagining and architecting the data and software systems to make conversations with LoreBot trustworthy and private. In addition to Data Engineering, Jonathon brings experience as a Backend Software Engineer.

Victoria Shapiro, J.D. is a regulatory intelligence and strategy expert at Lore. Recently, she's been engaged in shaping policy for us across the fast-moving areas of artificial intelligence, data privacy, and data governance, helping us stay out in front of emerging regulatory trends.

Problem Statements

We are not looking for a fully production-ready solution, but rather a clear demonstration of your problem-solving skills, your ability to think critically about DataOps challenges, and how you would strategize and define requirements to address them. Your solution should include concrete examples of code, schema definitions, or technical configurations where appropriate, demonstrating your ability to translate strategy into actionable technical designs. We look forward to discussing your ideas with you during the interview.

These scenarios are reflective of challenges we have encountered in the past and you might encounter in the course of a role at Lore. We recognize the scenario cannot be sufficiently detailed in this brief, but we are unable to answer questions prior to the call. However, you are encouraged to make assumptions based on your experience to provide a comprehensive solution strategy, including the detailed requirements derived from your approach.

Below are five problem statements. Please **select one** that you feel best demonstrates your skills and experience to provide a comprehensive solution strategy.

Problem Statement 1: Establishing a Global Data Platform CI/CD and Automation Framework

Scenario: Our global data platform is growing rapidly, encompassing numerous microservices, data pipelines, and analytical applications across different cloud environments (e.g., AWS, GCP). Currently, deployment processes for new features, bug fixes, and infrastructure changes are largely manual, inconsistent across teams, and prone to errors. This significantly slows down our innovation cycle, increases operational overhead, and makes it difficult to ensure the reliability and security of our data assets. We need a strategic initiative to design and implement a unified, highly automated CI/CD framework for all data platform components, emphasizing Infrastructure as Code (IaC) and continuous delivery.

Task: Describe your strategic vision for addressing this challenge. Your response should outline how you would holistically investigate the current state, identify key bottlenecks, and define the strategic opportunities for automation across the data platform lifecycle. Propose your future-state architecture principles for a unified CI/CD framework, detailing its conceptual design, key components (e.g., version control, CI servers, artifact repositories, deployment tools), and how it would support multi-cloud deployments and microservices. Define the key functional and non-functional requirements (e.g., deployment frequency, rollback capabilities, security compliance, auditability, speed of iteration, resource utilization) that would guide implementation and define success. As a hands-on component, provide a simplified IaC configuration code snippet (e.g., using Terraform or CloudFormation syntax for a data processing component or a basic data storage resource) that demonstrates idempotency and modularity within your proposed framework, along with a high-level CI/CD pipeline definition code snippet (e.g., using YAML for a Jenkinsfile or GitLab CI/CD pipeline) for a typical data microservice. Finally, provide a high-level phased approach for implementing and rolling out this framework across different data teams.

Problem Statement 2: Designing an Intelligent Data Quality & Observability System for Critical Data Products

Scenario: As Lore scales, the volume and criticality of our data products are increasing. While individual teams implement some data quality checks, there is no centralized, automated, and proactive system for monitoring data quality and ensuring data product reliability across the entire data platform. This leads to late detection of data anomalies, inconsistent data for downstream consumers, and a reactive approach to data issues. We need to establish a comprehensive data quality and observability system that provides real-time insights into the health and accuracy of our critical data products, enabling proactive issue resolution and building trust in our data.

Task: Outline your strategic approach to designing and implementing an intelligent data quality and observability system. Your response should detail how you would identify and prioritize critical data products for monitoring, define key data quality dimensions (e.g., completeness, accuracy, consistency, timeliness, validity), and establish clear Service Level Objectives (SLOs) and Service Level Indicators (SLIs) for data reliability. Propose a conceptual architecture for this system, detailing how it would integrate with existing data pipelines and storage (e.g., data lakes, warehouses), and specifying key components (e.g., data profiling, anomaly detection, alerting mechanisms, lineage tracking, dashboarding). For your proposed system, describe how it would automate data quality checks, detect

and alert on data anomalies, and provide comprehensive observability into data freshness, pipeline health, and resource utilization. As a hands-on component, provide a simplified code example (e.g., Python with a data quality library like Great Expectations or a custom SQL script) demonstrating an automated data quality check for a specific data product attribute, along with an abstract configuration for an alert (e.g., PagerDuty or Slack integration) triggered by a data quality anomaly. Finally, describe your plan for continuously improving and evolving this system based on operational feedback and new data product requirements.

Problem Statement 3: Optimizing Real-Time Data Flow and Operational Efficiency for a High-Scale Streaming Platform

Scenario: Lore is expanding into new real-time analytics use cases, requiring our data platform to ingest, process, and serve data streams at extreme scale (e.g., millions of events per second) with low latency. Our current streaming infrastructure, while functional, faces challenges with operational complexity, resource inefficiency, and difficulty in quickly diagnosing and resolving issues in a high-throughput environment. We need a strategic approach to optimize our real-time data flow, focusing on operational efficiency, cost-effectiveness, and ensuring resilience and scalability for mission-critical streaming applications.

Task: Outline your strategic vision for optimizing our real-time data flow and enhancing its operational efficiency. Your response should detail how you would assess the current streaming architecture (e.g., Kafka, Spark Streaming, Flink) to identify performance bottlenecks and areas for operational improvement. Propose a refined conceptual architecture for the high-scale real-time data platform, emphasizing principles of operational excellence, auto-scaling, fault tolerance, and cost optimization. Define key operational requirements (e.g., monitoring granularity, logging standards, disaster recovery, security of data in transit/at rest, incident response playbooks, cost per transaction/event) and propose metrics for measuring operational efficiency and system health. As a hands-on component, provide a simplified configuration example (e.g., Kubernetes YAML for a streaming application, or a Kafka Streams/Flink job configuration snippet) that demonstrates a best practice for operational resilience (e.g., consumer group rebalancing, state management, error handling) and a code snippet (e.g., Python or shell script) for an automated operational task (e.g., scaling a consumer group, purging old topics, or a health check). Throughout, explain your technical and architectural justifications, considering factors like message queuing strategies, stateful vs. stateless

processing, distributed tracing, and infrastructure automation for rapid provisioning and scaling.

Problem Statement 4: Automating GCP Cloud SQL Provisioning and Data Integration for Analytics

Scenario: Our product teams frequently require new Cloud SQL instances (e.g., PostgreSQL, MySQL) for various applications, ranging from transactional databases to specialized microservices. Data from these instances is crucial for central analytical purposes and needs to be replicated to our BigQuery data warehouse. Currently, provisioning new Cloud SQL instances, setting up change data capture (CDC) with DataStream to BigQuery, and ensuring seamless connectivity to analytical tools like Looker Studio, is a manual, time-consuming process. This leads to delays in data availability for analysis, inconsistent configurations, and potential security vulnerabilities. We need to implement a fully automated, scalable, and secure DataOps process for managing the lifecycle of these Cloud SQL instances and their data integration.

Task: Describe your strategic vision for addressing this challenge. Your response should outline how you would holistically design an automated process for provisioning new Cloud SQL instances, integrating them with DataStream for CDC to BigQuery, and ensuring their data is consumable by Looker Studio. Propose your future-state architecture principles for this automated pipeline, detailing the components involved (e.g., IaC tools, CI/CD pipelines, networking configurations, monitoring). Define the key functional and non-functional requirements (e.g., self-service provisioning, deployment speed, data freshness SLAs, security compliance, network isolation, cost optimization, observability of data flow) that would guide implementation and define success. As a hands-on component, provide a simplified IaC configuration (e.g., using Terraform) for provisioning a Cloud SQL instance and configuring its basic network access. Additionally, provide an abstract configuration snippet (e.g., a pseudo-YAML or JSON structure) for a DataStream stream, demonstrating how you would automate its setup for replication to BigQuery. Finally, describe your approach to continuously monitor the health and connectivity of these integrated systems, and how you would troubleshoot common issues.

Problem Statement 5: Automating Data Access Management & Security Governance

Scenario: As our data platform grows, managing access to sensitive data across various tools and databases has become a significant operational and security challenge. We

currently face manual, ad-hoc processes for granting data access, leading to delays, inconsistent security postures, and difficulty in auditing who has access to what, and why. Specifically, our data consumers frequently request access to Looker Studio dashboards and underlying Cloud SQL (PostgreSQL/MSSQL) databases. We need a strategic, automated solution that streamlines access requests, enforces security policies, and provides a clear audit trail for all data access grants, ensuring compliance and data governance.

Task: Outline your strategic vision for implementing an automated and scalable data access management solution that covers the entire lifecycle of data access requests, from user initiation to automated provisioning and de-provisioning. Your response should detail a conceptual design for the automated workflow, identifying key stakeholders and their roles in the approval process, and proposing the technical components and technologies required for scalability, security, and maintainability. Focus on how approved access would be automatically provisioned for both Looker Studio dashboards and custom roles/user permissions within Cloud SQL instances (PostgreSQL/MSSQL), adhering to the principle of least privilege, and ensuring comprehensive auditability. As a hands-on component, provide a simplified code snippet (e.g., Python or a shell script leveraging a cloud SDK/API) illustrating how an automated Cloud SQL custom role creation and user grant might be implemented based on an approved request. Additionally, include an abstract YAML/JSON configuration code snippet demonstrating a part of your automated workflow or access policy for either Looker Studio or Cloud SQL access. Finally, describe how you would measure the success and security posture of this automated access management solution.