Senior / Staff ML Engineer – Mock Interview Checklist & Prompts

Use this checklist before each mock interview session. Then use the provided prompts to request a specific interview simulation. This is designed for ML Engineers specializing in Distributed Systems aiming for Senior/Staff roles.

1. ML System Design Readiness

- Can I clearly frame the business problem and define the target label?
- Can I explain online vs offline pipelines and why each exists?
- Can I design low-latency (<50ms) prediction paths end-to-end?
- Do I know where feature engineering happens (streaming vs batch)?
- Can I discuss model versioning, rollback, and safe deploys?
- Can I explain tradeoffs (latency vs accuracy, freshness vs stability)?

2. Feature Store & Data Architecture

- Can I explain offline store (S3/HDFS/BigQuery) vs online store (Redis/Cassandra)?
- Do I understand point-in-time correctness and data leakage prevention?
- Can I explain feature freshness, TTL, and backfills?
- Do I know how to handle schema evolution?
- Can I discuss joins at training vs inference time?

3. Distributed Systems Fundamentals

- Can I explain CAP theorem and tradeoffs?
- Do I understand partitioning strategies (hash, range, consistent hashing)?
- Can I reason about hot keys and rebalancing?
- Do I understand exactly-once vs at-least-once semantics?
- Can I explain idempotency in pipelines?

4. Training at Scale

- Can I explain data parallelism vs model parallelism?
- Do I understand sharding and distributed training?
- Can I reason about GPU/CPU utilization?
- Do I know how to reduce training time (caching, sampling, pipelines)?

5. Model Serving at Scale

- Can I explain batching, async inference, and caching?
- Do I understand model loading strategies and memory pressure?
- Can I reason about horizontal scaling and autoscaling?
- Do I know how to handle multi-model serving?

6. Reliability & Failure Handling

- Can I walk through debugging latency spikes step-by-step?
- Do I know how to design circuit breakers and fallbacks?
- Can I explain blast radius control (canary, shadow, rollback)?
- Do I understand failure isolation between components?

7. Monitoring & Drift

- Can I explain data drift vs concept drift?
- Do I know statistical methods for drift detection?
- Can I design alerting without noise?
- Do I know how to handle delayed labels?

8. Performance & Optimization

- Can I break down a latency budget?
- Do I know where bottlenecks usually occur?
- Can I suggest optimizations without adding hardware?

9. Architecture Tradeoffs

- Can I argue build vs buy with long-term cost reasoning?
- Do I understand monolith vs microservices tradeoffs for ML?
- Can I explain org structure impact on architecture?

10. Leadership & Staff-Level Thinking

- Can I handle cross-team conflicts with data and empathy?
- Do I think in terms of ROI and business impact?
- Can I articulate a 6-month technical vision?
- Can I prioritize and say no when needed?

Mock Interview Prompts (Use These With ChatGPT)

- Run a Senior MLE system design interview on real-time prediction system.
- Simulate a Staff MLE interview focused on feature store design.
- Give me a distributed systems deep dive interview for ML serving.
- Act as Meta interviewer and test my ML platform design.
- Run a failure-debugging scenario interview for ML production system.
- Test me on leadership and cross-team conflict as a Staff MLE.
- Do a brutal Senior-level feedback on my ML system design answers.

How to Use This Checklist

- Before each session, scan all sections and ensure you can speak confidently on each point.

- During mock interviews, think in terms of architecture, tradeoffs, and business impact – not just code.
- After each mock interview, note weak areas and revisit them.
- This checklist is designed to move you from Senior to Staff thinking.