Staff Software Engineer

Behavioral Interview Stories - Part 2

Candidate:	Luv Saxena
Position:	Staff Software Engineer
Date:	February 14, 2026
Stories Covered:	2 Behavioral Interview Stories

Story 1: Building a Data Lake with Apache Cassandra

Interview Question:

Tell me about a time when you had to make a technical decision that you weren't fully confident about, but you had to move forward anyway due to time or business pressure. How did you handle it? What was the outcome?

Story Context:

Project: Building a centralized real-time data lake at a startup (WinZo) to support analytics and decision-making across multiple teams

The Challenge:

• **Business pressure:** Multiple teams (6+ teams) needed access to real-time data for analytics

• **Requirements:** Needed horizontally scalable, distributed database to handle high write throughput (15-20K writes/sec)

• **Query pattern:** User ID + date range queries (partition key + clustering key)

• **Initial uncertainty:** Not 100% confident about the choice but had to move forward with limited time

Your Decision-Making Process:

Step 1: Evaluated Multiple Options

<b>Database</b>	<b>Pros</b>	<b>Cons / Why Not Chosen</b>
<b>Cassandra</b>	• High write throughput<br/>• Horizontally sca[illegible]	• Never used in production before<br/>• Eventual consistency (availability focused)<br/>• Some uncertainty about operations [illegible] Partition key
<b>HBase</b>	• Columnar distributed DB<br/>• Previous exp[illegible]	• Favors consistency over availability<br/>• Good for row key queries<br/>• Not used for date range querie
<b>MongoDB</b>	• Popular NoSQL option<br/>• Good documen	• No experience with distributed cluster<br/>• Not ideal for write-heavy wor

Step 2: Data-Driven Validation

• **Researched benchmarks:** Found reliable sources showing Cassandra could handle 15-20K writes/sec with p99 <10ms

• **Built proof of concept:** Created middleware library and tested locally with real data

• **Validated query patterns:** Confirmed partition key (user ID) + clustering key (date) met requirements

• **Made the call:** Despite some uncertainty, benchmarks + POC gave enough confidence to proceed

Implementation & Architecture:

Architecture Components: Kafka → Apache Spark Streaming → Cassandra 3-node cluster on AWS

• **Cluster setup:** 3-node Cassandra cluster with replication factor of 2

• **Consistency tuning:** Write quorum = 1, Read quorum = 2 (strong consistency with tunable guarantees)

• **Middleware library:** Built TypeScript Kafka producer library adopted by 6 teams

• **Data flow:** Real-time streaming from multiple systems into centralized data lake

Quantified Outcomes:

<b>Metric</b>	<b>Result</b>
Write throughput (normal)	15,000 writes/sec
Write throughput (peak)	30,000-40,000 writes/sec
Read throughput	5,000 reads/sec (date range queries)
P99 latency	<10ms

CPU utilization (normal)	15-20%
CPU utilization (peak)	Handled spikes gracefully, no issues
Production issues	Zero issues - never witnessed any cluster problems
Storage utilization	700-800MB out of 2-3GB (efficient)
Availability improvement	Monolith availability: 96-97% → 99% (offloaded queries)
Teams using data lake	6+ teams adopted the system

Business Impact:

✓ **Offloaded analytical queries:** Moved analytics from monolithic database to Cassandra

✓ **Enabled data-driven decisions:** Teams could query player history, activity patterns for offers/promotions

✓ **Improved system availability:** Monolith availability increased from 96-97% to 99%

✓ **Org-wide adoption:** 6+ teams relied on this data lake for analytics and decision-making

✓ **No production incidents:** Zero issues despite high throughput and multiple dependent teams

What You Learned / Would Do Differently:

• **Benchmark competitors too:** Should have done performance benchmarking with HBase/MongoDB to have stronger comparative data

• **Data-driven confidence:** Even with uncertainty, benchmarks + POC provided enough validation to move forward

• **Operational excellence matters:** Monitoring, tuning, and proper configuration (quorum settings, replication) were key to success

Interview Feedback & Rating:

Overall Rating: 9/10

Strengths:

✓ **Clear decision-making under uncertainty** - admitted you weren't 100% confident but moved forward with data

✓ **Data-driven approach** - researched benchmarks, built POC, validated assumptions

✓ **Quantified outcomes** - 15K-40K writes/sec, p99< 10ms, availability 96% → 99%

✓ **Technical depth** - discussed partition keys, clustering keys, quorum settings, replication factors

✓ **Org-level impact** - 6+ teams adopted the system, improved overall system availability

✓ **Humility and learning** - acknowledged what you could have done better (HBase benchmarking)

✓ **Real production validation** - zero issues, handled spikes beautifully, never crashed

Minor Areas for Improvement (to reach 9.5/10):

• Could emphasize stakeholder alignment more - did you present this decision to leadership/architects?

• Could discuss failure scenarios upfront - what if Cassandra had failed? Rollback plan?

• Answer was quite long (~8-10 mins) - in real interviews, keep to 3-4 minutes

This Story Maps Well To:

✓ Tell me about a tough technical decision you made

✓ How do you make decisions with incomplete information?

✓ Tell me about a high-impact project you led

✓ How do you evaluate trade-offs between different technologies?

✓ Tell me about scaling challenges you've faced

✓ How do you handle ambiguity and uncertainty?

Story 2: Multi-Cloud Migration - AWS to Azure

Interview Question:

Tell me about a time when you had to deliver a critical project on a very tight deadline, and something unexpected happened that threatened the timeline. How did you handle it? What trade-offs did you make?

Story Context:

Project: Multi-cloud migration at Tekion - migrating 8-10 backend services from AWS to Azure to enable dual-cloud operation

The Challenge:

• **Business driver:** Company secured a favorable deal with Microsoft, needed to onboard new customers on Azure

• **Technical scope:** Migrate 8-10 microservices to work seamlessly on both AWS and Azure

• **Tight timeline:** Aggressive deadline to start onboarding new customers

• **Complexity:** Replace AWS-native services (S3, SQS, SNS, Lambda) with Azure equivalents (Blob Storage, Event Bus, Azure Functions)

• **Your role:** Led the entire migration as project lead

Technical Architecture:

<b>AWS Service</b>	<b>Azure Replacement</b>
S3 (Object Storage)	Azure Blob Storage
SQS (Queue)	Azure Event Bus
SNS (Pub/Sub)	Azure Event Hub
Lambda (Serverless)	Azure Functions

Solution Approach: Configuration-driven cloud abstraction - services check 'cloud' variable at runtime to determine which native services to use (AWS or Azure)

Unexpected Challenges That Threatened Timeline:

• **Azure Functions not deploying:** Infrastructure issues prevented Azure Functions from coming up - had to coordinate with infra team for resolution

• **Dependent services not working:** Platform layer services and other dependencies were not functional in Azure environment - required proactive follow-up with multiple teams

• **UI issues:** Frontend wasn't working properly in Azure setup - needed coordination with UI team

• **Library upgrades required:** S3 upload library needed upgrade to 'media upload library' to support both clouds - dependency on platform team

• **Integration testing blocked:** Couldn't do proper end-to-end testing because of missing dependencies

• **Multiple teams involved:** Had to coordinate across infra team, platform team, QA team, services teams, UI team

How You Handled It:

• **Proactive coordination:** Continuously followed up with all dependent teams to get ETAs and unblock issues

• **Escalation when needed:** Coordinated with infra team to fix Azure Functions deployment issues

• **Testing rigor:** Ensured critical flows (Excel generation, PDF generation using Lambda/Azure Functions) were thoroughly tested

• **Stakeholder communication:** Kept leadership informed of all dependencies, blockers, and risks throughout the project

• **Risk mitigation strategy:** Extensively tested AWS flows before merging to ensure zero impact to existing customers

Risk Mitigation - Phased Rollout:

To minimize risk, you implemented a **phased merge strategy** to protect existing AWS customers:

• **Phase 1: Test AWS flows extensively** - Before any code merge, validated that AWS functionality remained intact

• **Phase 2: Merge less critical services first** - Started with services that had lower business impact

• **Phase 3: Validate Azure functionality** - Ensured Azure-specific code worked correctly with integration tests

• **Phase 4: Merge mission-critical services** - Only after confidence was high, merged accounting and financial statement services (highest risk)

• **Result: Zero impact to existing AWS customers** - Seamless migration with no downtime or issues

Quantified Outcomes:

✓ **Successfully migrated 8-10 microservices** to multi-cloud architecture

✓ **Zero impact to existing AWS customers** - extensive testing prevented any disruption

✓ **Enabled new revenue stream** - company could now onboard customers on Azure

✓ **$60-70K cost savings** for 100 new customers across 6-7 organizations

✓ **Org-wide benefit** - accounting, financial statements, and other critical services now multi-cloud

✓ **Seamless dual-cloud operation** - services work on both AWS and Azure based on configuration

Leadership & Stakeholder Management:

• **Cross-functional coordination:** Worked with infra team, platform team, QA lead, services teams, UI team

• **Continuous communication:** Kept leadership informed of blockers, dependencies, and timeline risks

• **Proactive problem-solving:** Didn't wait for issues to resolve themselves - actively followed up and escalated

• **Team collaboration:** Worked closely with QA lead for integration testing strategy

Interview Feedback & Rating:

Overall Rating: 8.5/10

Strengths:

✓ **High-impact project leadership -** migrated 8-10 services, enabled new revenue stream

✓ **Handled unexpected challenges -** multiple blockers (Azure Functions, dependencies, libraries) but drove resolution

✓ **Cross-functional coordination -** worked with infra, platform, QA, services, UI teams

✓ **Risk mitigation** - phased rollout protected existing AWS customers (zero impact)

✓ **Stakeholder communication** - kept leadership informed of dependencies, blockers, timeline risks

✓ **Quantified business impact** - $60-70K savings, enabled new customer onboarding

✓ **Proactive problem-solving** - didn't wait for issues to fix themselves, escalated and followed up

Areas for Improvement (to reach 9/10):

• Could emphasize **specific escalation techniques** - HOW did you get teams to prioritize your blockers?

• Could discuss **trade-offs more explicitly** - what did you sacrifice to meet timeline? Features deferred?

• Could mention **monitoring/observability** - how did you validate zero AWS customer impact post-merge?

• Answer was quite long (~6-7 mins) - in real interviews, keep to 3-4 minutes

This Story Maps Well To:

✓ Tell me about a complex project with multiple dependencies

✓ How do you handle blockers and keep projects moving?

✓ Tell me about a time you had to balance speed and safety

✓ How do you manage stakeholders and communicate risks?

✓ Tell me about a high-impact project you led

✓ How do you coordinate across multiple teams?

✓ Tell me about handling unexpected challenges under tight deadlines

Summary & Interview Readiness

Your Behavioral Story Portfolio:

<b>Story</b>	<b>Rating</b>	<b>Key Theme</b>
Cassandra Data Lake Decision	9/10	Technical decision-making under uncertainty
AWS → Azure Multi-Cloud Migration	8.5/10	Cross-functional project leadership
Flipkart Monolith → Microservices	8.5/10	Stakeholder alignment & org-wide impact
RingCentral Engineer Mentoring	8.5/10	Developing talent & mentorship

Overall Assessment:

You have **strong, well-rounded behavioral stories** that demonstrate Staff-level leadership across multiple dimensions: technical decision-making, cross-functional collaboration, risk management, stakeholder communication, and mentorship. Your stories are backed by quantified outcomes and show real org-level impact.

Key Strengths Across All Stories:

✓ **Data-driven decision-making** - you use benchmarks, POCs, and testing to validate choices

✓ **Quantified impact** - every story has metrics (throughput, latency, cost savings, availability)

✓ **Org-level thinking** - you focus on broader impact, not just team-level execution

✓ **Risk management** - phased rollouts, extensive testing, monitoring, failure planning

✓ **Stakeholder management** - you coordinate across teams and communicate with leadership

✓ **Humility & learning** - you acknowledge what you could have done better

Areas to Refine:

• **Keep answers concise** - aim for 3-4 minutes per story in real interviews

• **Emphasize YOUR decisions** - use 'I decided' vs. 'we decided' to show ownership

• **Discuss failure scenarios upfront** - what if X failed? What was the rollback plan?

• **Add conflict/disagreement elements** - did anyone push back? How did you convince them?

Interview Readiness:

You are **interview-ready for Staff Engineer roles**. Your behavioral stories demonstrate:

✓ Technical leadership (Cassandra decision, multi-cloud architecture)

✓ Project management under complexity (Azure migration with blockers)

✓ Org-wide influence (multiple teams adopted your solutions)

✓ Mentorship & talent development (promoted engineers)

✓ Risk mitigation & operational excellence (phased rollouts, zero downtime)

Recommended Next Steps:

1. **Practice condensing** - rehearse these stories in 3-4 minute versions

2. **Add conflict elements** - think about who pushed back and how you convinced them

3. **Prepare 1-2 'failure' stories** - what didn't go well initially? How did you recover?

4. **Mock interview practice** - do 1-2 more mocks right before actual interviews to stay sharp

5. **Rest & let practice settle** - you've done solid prep, trust your preparation

Final Note:

These stories show genuine Staff-level thinking and impact. Combined with your system design skills (rate limiter 8.5/10, Rippling news aggregator 8.5-9/10), you're well-positioned for Staff engineer interviews at top companies. Focus on delivery, stay confident, and trust your preparation.