interview101

PERSONALIZED INTERVIEW PLAYBOOK

Your Roadmap to Landing This Role

Everything you need to walk into your interview confident, prepared, and ready to win.

PREPARED FOR
Jordan M. Chen

TARGET ROLE
SDE

COMPANY
Amazon

GENERATED
May 04, 2026

1

Your Interview Prep Starts Here

A quick snapshot of where you stand and how to use this report.

YOUR QUICK ASSESSMENT

Your background demonstrates exceptional technical depth that directly aligns with Amazon's engineering standards, particularly your proven ability to architect scalable solutions—a critical capability for success as an SDE at the company.

To succeed as an Amazon SDE, prioritize strengthening your system design skills—particularly around scalability and distributed systems trade-offs. Focused practice on these areas, combined with mock interviews, will significantly boost your readiness.

What's Inside

Short on Time?
30-minute prep path

- → Read Where You Stand (Section 2)
- → Memorize your pitches (Section 4)
- → Review your top 3 stories (Section 5)
- → Grab the cheat sheet (Section 10)

Full Preparation
2-hour deep dive

- → Read the full report front-to-back
- → Practice all stories out loud
- → Role-play the top 5 questions
- → Customize your questions to ask

2

Where You Stand

An objective assessment of your fit for this role, based on your resume and the job requirements.

YOUR FIT ASSESSMENT

Strong Match

Your distributed systems expertise and mentoring track record make you a strong technical leader for this SDE role. Sharpening your algorithms foundation before onboarding would set you up for immediate impact.

84 / 100

Skills match — 37 / 40

You match nearly all required skills including Java, Python, distributed systems, microservices, and both SQL/NoSQL databases, with strong AWS expertise.

Experience alignment — 35 / 35

Your 7 years and senior-level experience at top tech companies directly aligns with SDE 2 requirements for distributed systems development.

Culture & role fit — 12 / 25

Your resume doesn't yet signal Amazon's leadership principles clearly; adding explicit customer obsession and ownership language would strengthen cultural alignment.

Your Strengths

Proven High-Scale Distributed Systems Architecture

Your experience designing microservices handling 500K+ requests/second at Microsoft Azure directly addresses Amazon's requirement to "design, build, and maintain highly scalable distributed systems." You've architected event-driven pipelines processing 2B+ events daily with 99.99% uptime using AWS services like SQS, Lambda, and DynamoDB—the exact tech stack Amazon uses. **This hyperscale experience positions you to immediately contribute to Amazon's infrastructure demands.**

Technical Leadership with Mentoring Impact

Your role leading technical design reviews for 8 engineers at Microsoft and mentoring 3 junior engineers to promotion perfectly aligns with Amazon's expectation to "lead technical design reviews and mentor junior engineers." Combined with your cross-functional collaboration experience at top-tier companies like Microsoft and Expedia, you demonstrate the leadership depth Amazon seeks. **Your proven ability to develop talent while driving technical decisions makes you an ideal SDE 2 candidate.**

Customer-Obsessed Operational Excellence Champion

Your customer obsession initiative that reduced search errors by 62% at Microsoft Azure exemplifies Amazon's core "Customer Obsession" principle. You've owned full on-call rotations and improved MTTD from 14 to 4 minutes through enhanced CloudWatch alerting, demonstrating the "operational excellence through monitoring, alerting, and on-call practices" Amazon requires. **This combination of customer focus and operational rigor reflects Amazon's cultural DNA perfectly.**

Gaps to Address

Missing Algorithms and Data Structures Depth

While your resume demonstrates strong system architecture skills, the job description explicitly requires "data structures and algorithms" expertise, which isn't evidenced in your experience bullets. Amazon's technical interviews heavily emphasize algorithmic problem-solving, and your background focuses on distributed systems rather than **core computer science fundamentals**. Your microservices and pipeline work shows engineering maturity, but doesn't demonstrate the coding interview skills Amazon prioritizes. You'll need to refresh leetcode-style problem solving and be ready to discuss algorithmic complexity in your solutions.

→ See Section 5 for stories to bridge this

Limited Cross-Team Leadership Evidence

The role requires "collaborate cross-functionally with product managers, designers, and partner teams," but your resume primarily shows technical mentorship within engineering teams. While you mention "cross-functional collaboration" in skills and led design reviews for 8 engineers, there's **no specific evidence of leading initiatives across product, design, or business stakeholders**. Amazon SDE2s frequently drive feature decisions with non-technical partners, requiring demonstrated influence beyond your engineering organization. You should prepare examples of working with product managers or driving customer-facing decisions. → See Section 7 for scripts

Weak Amazon Leadership Principles Alignment

Your "customer obsession initiative" shows promise, but your resume lacks depth around Amazon's core leadership principles like "Invent and Simplify," "Bias for Action," and "Ownership." Your experience emphasizes technical excellence but doesn't clearly demonstrate **innovative problem-solving or simplification of complex systems**. Amazon expects SDE2s to challenge existing approaches and drive simplification, but your bullets focus more on performance optimization than fundamental innovation. You need stronger examples of questioning assumptions, inventing new solutions, or dramatically simplifying complex workflows to match their cultural expectations. → See Section 5 for stories to bridge this

YOUR PREP PRIORITY

Here's how to use this report

You're walking in with exceptional technical depth and proven scale—your real advantage is the distributed systems credibility that Amazon deeply respects. Priority one is anchoring your culture fit gap by building five STAR stories explicitly mapped to Amazon's Leadership Principles in Section 5, because your 48% culture score will come up in every behavioral round and interviewers need to see you own customer obsession and bias for action, not just technical excellence. Priority two is weaponizing Section 6's twelve likely questions to practice your cross-team leadership narrative, since your mentoring impact is real but currently buried—surface it early and often. Start by reading Section 3 to decode exactly what this SDE level role demands, then jump straight to Section 5 to craft your stories. You've got the technical chops; now just translate them into Amazon's language.

3

What They Actually Want

The job description tells part of the story. Here's what's really driving this hire.

Reading Between the Lines

WHAT THEY SAID	WHAT THEY MEAN	HOW YOU DEMONSTRATE IT
"highly scalable distributed systems and microservices"	Amazon tests Invent and Simplify through system design depth — can you architect solutions that handle Amazon-scale traffic while maintaining simplicity?	Your **500K+ requests/second microservices** at Microsoft Azure directly demonstrates this scale. Emphasize the architectural decisions that enabled this throughput.
"Lead technical design reviews and mentor junior engineers"	Amazon evaluates Earn Trust and Learn and Be Curious — can you influence without authority and develop talent at their leadership bar?	Your experience **leading technical design reviews for 8 engineers** and mentoring 3 junior engineers to promotion shows both technical influence and talent development.
"operational excellence through monitoring, alerting, and on-call practices"	Amazon tests Ownership and Deliver Results — do you take full accountability for production systems and proactively prevent customer impact?	Your **owned full on-call rotation** experience and reducing MTTD from 14 to 4 minutes through improved CloudWatch alerting demonstrates operational ownership.
"Own features end-to-end, from architecture through deployment"	Amazon evaluates Ownership and Bias for Action — can you drive complex initiatives independently without needing constant direction or support?	Your **customer obsession initiative** to rewrite the search indexing service shows end-to-end ownership, but emphasize the full lifecycle from design to deployment.
"Search technologies (Elasticsearch, Solr)"	Amazon prioritizes Customer Obsession through search relevance — can you build discovery experiences that help customers find what they need efficiently?	Your **Elasticsearch integration for property search** at Zillow, improving relevance score by 28%, directly demonstrates search technology expertise and customer impact focus.

Why This Role Exists

Amazon's massive scale requires engineers who can build systems serving millions of customers while maintaining the company's obsession with operational excellence. The emphasis on distributed systems, microservices, and AWS technologies reflects Amazon's need to continuously innovate across their e-commerce platform, cloud infrastructure, and numerous product lines while maintaining their 'Day 1 mentality' of rapid iteration.

The combination of technical leadership expectations, mentoring responsibilities, and end-to-end ownership signals a team transitioning from individual contributors to a more mature engineering organization. The specific emphasis on operational excellence, monitoring, and on-call practices suggests recent scaling challenges. The focus on cross-functional collaboration indicates they've experienced coordination friction between engineering and product teams.

The pain they're solving: They need someone who can bridge the gap between **technical execution and organizational maturity**. Previous engineers likely delivered features but struggled with system reliability, team coordination, or developing junior talent. You should position yourself as someone who doesn't just code but elevates team standards, owns operational excellence, and can mentor while driving results.

Company Intelligence That Matters

LEADERSHIP PRINCIPLES IN PLAY

Every interview round evaluates alignment to these values. For this role, focus on:

- Customer Obsession
- Ownership
- Invent and Simplify
- Are Right, A Lot
- Learn and Be Curious
- Hire and Develop the Best
- Insist on Highest Standards
- Think Big
- Bias for Action
- Frugality
- Earn Trust
- Dive Deep
- Have Backbone; Disagree and Commit
- Deliver Results
- Strive to be Earth's Best Employer
- Success and Scale Bring Broad Responsibility

Have a STAR story ready for each. Vague answers are the #1 reason qualified candidates fail.

INTERVIEW PROCESS

Expect phone/video screening, online work style assessments, then 4-6 hour virtual loop with multiple engineers. Each interviewer owns 2-3 Leadership Principles using STAR behavioral questions. The Bar Raiser interview is conducted by a trained interviewer ensuring hiring standards. Technical rounds cover system design, coding, and operational excellence.

CULTURE SIGNAL

Amazon operates through written narratives instead of PowerPoint, requiring deep thinking before meetings. The 'Day 1 mentality' drives everything—customer obsession over competitor focus, long-term thinking over short-term wins. High standards aren't negotiable, and 'peculiar' culture means doing things differently from other tech companies.

WHAT SUCCESS LOOKS LIKE

SDE 2s own features end-to-end while mentoring junior engineers. They write narrative design documents, lead technical reviews, and drive operational excellence through monitoring and on-call practices. They balance speed with quality, make reversible decisions quickly, and communicate technical strategy across teams through written memos.

What Makes This Interview Different

UNIQUE TO AMAZON

Amazon's Bar Raiser is a senior employee from a completely different team who has veto power over the hiring decision. They are not evaluating you for this specific team — they are evaluating whether you raise the bar for Amazon overall. Their questions will be harder, their follow-ups more probing, and their LP expectations more exacting than any other interviewer.

Hidden Priorities (What the JD Reveals)

1. **Technical Leadership Beyond Individual Contribution** JD signal

   The JD emphasizes 'Lead technical design reviews' and 'mentor junior engineers' alongside 'technical project leadership' in preferred skills. This signals they want someone who can **influence technical decisions** across teams, not just write code well.

2. **End-to-End Operational Ownership Mindset** JD signal

   Notice how 'Own features end-to-end' appears last in responsibilities but combines with 'operational excellence through monitoring, alerting, and on-call practices.' They want someone who thinks like a **service owner**, not just a feature developer.

3. **Bar Raiser Interview Will Test Standards** Company intel

   Amazon's hiring process includes a Bar Raiser interview conducted by someone specifically trained to ensure you meet hiring standards. This person will focus on whether you **raise the bar or lower it** across all Leadership Principles, not just technical skills.

Watch Out For These Mistakes

Underselling Your Customer Obsession Impact

Amazon interviewers will probe deeply on Customer Obsession examples, and your search indexing rewrite that cut errors by 62% is perfect material. Don't just state metrics—explain **how you identified customer pain points** and prioritized their needs over technical convenience. Use Section 5 stories to detail your decision-making process and what you learned about customer impact from this initiative.

Avoiding System Design Deep Dives

Your microservices handling 500K+ requests/second will trigger detailed architecture questions, but the job requires explicit distributed systems design knowledge. When discussing your Azure pipeline processing 2B+ events daily, **explain partition strategies, consistency models, and failure handling** in detail. Reference Section 7 gap scripts to prepare for questions about CAP theorem, consensus algorithms, and data consistency patterns.

Generic Leadership Stories Without Ownership

Amazon's Ownership principle demands you treat every project like your own company. Your mentoring of 3 junior engineers and leading 8-person design reviews shows leadership, but frame these with **long-term business ownership thinking**. Explain how you made decisions considering future maintenance costs, team scalability, and business impact. Use Section 5 to craft stories showing you optimize for long-term success over short-term wins.

WHAT THIS MEANS FOR YOUR INTERVIEW

- → **Prepare STAR examples** — Have 2-3 detailed stories ready for each Leadership Principle, focusing on Customer Obsession, Ownership, and Deliver Results.
- → **Emphasize end-to-end ownership** — Describe projects where you owned architecture, implementation, deployment, and operational monitoring, not just coding tasks.
- → **Demonstrate written communication** — Mention technical design documents, post-mortems, or decision memos you've written to show alignment with narrative culture.
- → **Show mentorship impact** — Provide specific examples of developing junior engineers, conducting code reviews, or raising team technical standards.
- → **Quantify customer impact** — Connect your technical decisions to customer outcomes, performance improvements, or business metrics rather than just technical achievements.

4

Your Story, Interview-Ready

Polished answers to "Tell me about yourself" — the most predictable question, and the easiest to fumble.

The 30-Second Pitch

~30 seconds

> I'm Jordan Chen, a **Senior Software Engineer with 7 years** building distributed systems at Microsoft, Expedia, and Zillow. I've architected microservices handling **500K+ requests per second** and event-driven pipelines processing 2B+ daily events with 99.99% uptime on AWS. My experience leading technical design reviews and driving **customer obsession initiatives** that reduced errors by 62% aligns perfectly with Amazon's leadership principles. I'm excited to bring this ownership mindset to scale Amazon's mission-critical systems.

1. **Hook:** Establishes credibility immediately with specific experience
2. **Proof:** Concrete numbers make it real and memorable
3. **Bridge:** Connects your past to their specific opportunity
4. **Close:** Shows ambition without sounding desperate

The 2-Minute Version

~2 minutes

Past: I started my career at Zillow building backend services in Python for property search, then progressed to Expedia where I migrated monolith to microservices architecture serving 50M monthly users.

Present: At Microsoft Azure, I've scaled to new heights — designing distributed microservices handling 500K+ requests per second and architecting event-driven pipelines processing 2B+ events daily while leading technical reviews for 8 engineers.

Pivot: I'm ready to take on broader ownership and think bigger about customer impact at massive scale.

Future: Amazon's customer obsession and operational excellence culture aligns perfectly with my experience driving customer-facing improvements and owning full operational responsibility for high-scale distributed systems.

"Why This Company?"

I'm excited by Amazon's $100B AWS AI infrastructure investment through 2027. My experience building distributed systems handling 500K+ requests per second at Microsoft directly aligns with this massive scaling challenge. Amazon's **customer obsession** resonates deeply—I led similar initiatives reducing customer-facing errors by 62%. I'm energized to help build the infrastructure powering AI at unprecedented scale.

Tip: This connects the $100B AWS AI infrastructure news to your distributed systems experience, showing immediate relevance to Amazon's current priorities.

"Why this role specifically?"

What excites me most is the opportunity to **"own features end-to-end, from architecture through deployment and operational excellence"** - exactly what I did at Microsoft Azure, where I architected event-driven pipelines processing 2B+ events daily while owning full on-call rotation. The focus on **"operational excellence through monitoring, alerting, and on-call practices"** aligns perfectly with my experience reducing MTTD from 14 to 4 minutes through improved CloudWatch alerting.

Tip: Mirror exact JD language like "end-to-end ownership" and "operational excellence" while backing each phrase with specific resume metrics and achievements.

Delivery Tips

- ✓ **Practice out loud** — reading silently isn't the same. Record yourself and listen back.
- ✓ **Pause after key points** — let your numbers land. Important stats deserve a beat.
- ✓ **End with forward energy** — your last sentence should make them want to hear more.
- ✗ **Don't memorize word-for-word** — know the beats, not the script. Robotic delivery kills credibility.
- ✗ **Don't rush the "why here"** — this is where you show genuine interest. Slow down.
- ✗ **Don't apologize for gaps** — not here. Save objection handling for when they ask directly.

5

8 Stories That Win Interviews

Your proof points — built from your resume, ready to personalize and deliver.

How These Stories Work

We've built STAR stories from your resume — but **only you know the full details**. Each story has three parts:

✓ **Verified** = Facts directly from your resume (company, role, metrics)
◂ **Draft** = Plausible details we've inferred — **review and correct these**
○ **You fill in** = Details only you know — add these before your interview

From your resume Draft — verify this You fill in

Before your interview: Read each story, correct anything we got wrong, and fill in the blanks. Practice telling each story in 2-3 minutes. The goal isn't to memorize — it's to know the beats so you can deliver naturally.

YOUR 8 STORIES

1. High-Scale Microservices Performance
2. Billion-Scale Data Pipeline
3. Team Leadership and Development
4. Operational Excellence Ownership
5. Customer-Driven Service Redesign
6. Monolith to Microservices Migration
7. Quality Engineering Standards
8. Search Technology Integration

1 High-Scale Microservices Performance

Invent and Simplify | Deliver Results

✓ FROM RESUME What we know for certain

"Designed and built distributed microservices handling 500K+ requests/second using Java and Python, reducing p99 latency by 38%"

USE THIS STORY FOR

Tell me about a time you optimized system performance / Describe a complex technical project you led / Give an example of when you simplified a complex system

DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company name]*, our distributed microservices architecture was struggling with performance issues that were impacting user experience. *[What specific symptoms were users experiencing? Timeouts? Slow responses?]* The system was handling significant traffic volume, but our p99 latency was unacceptable for *[what type of application/service?]*.

TASK VERIFY

> I needed to identify the root causes of the latency issues and redesign the microservices to handle 500K+ requests per second while dramatically improving response times. *[What was the deadline or business pressure driving this work?]*

ACTION VERIFY

> I conducted a comprehensive performance analysis using *[what specific profiling tools?]* and identified bottlenecks in *[which components - database calls, network, CPU?]*. I then redesigned the services using *[what specific Java/Python frameworks and patterns?]*, implemented *[what caching strategy?]*, and optimized *[what specific code paths or algorithms?]*. *[What was your testing strategy to validate the improvements?]*

RESULT FROM RESUME

> The redesigned microservices successfully handled 500K+ requests/second with a 38% reduction in p99 latency. *[What was the business impact? User satisfaction improvements? Revenue impact? Did this work lead to recognition or expanded responsibilities?]*

Before You Use This Story

- [ ] Add the 'before' p99 latency number to quantify the 38% improvement
- [ ] Specify which performance bottlenecks you identified and how
- [ ] Detail the specific technologies and optimization techniques used

Likely Follow-up Questions — Prepare Your Answers

"How did you identify which microservices were causing the bottlenecks?"

> *Walk through your debugging methodology - monitoring tools, metrics you looked at, how you traced requests across services*

"What tradeoffs did you consider when redesigning for performance?"

> *Discuss memory vs CPU, consistency vs performance, complexity vs maintainability - show systems thinking*

"How did you ensure the changes didn't break existing functionality?"

> *Describe your testing strategy, rollout plan, monitoring during deployment*

2 Billion-Scale Data Pipeline

Think Big | Insist on the Highest Standards

✓ FROM RESUME What we know for certain

"Architected event-driven data pipeline on AWS (SQS, Lambda, DynamoDB) processing 2B+ events per day with 99.99% uptime"

USE THIS STORY FOR

Tell me about a time you thought big / Describe a system you built from scratch / Give an example of when you insisted on high standards

‹ DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company name]*, we needed to process massive volumes of event data in real-time for *[what business purpose - analytics, recommendations, monitoring?]*. The existing data processing infrastructure couldn't scale to handle the growth we were experiencing, and *[what problems was this causing - data loss, delays, system crashes?]*.

TASK VERIFY

> I was tasked with architecting a completely new event-driven data pipeline that could reliably process billions of events per day while maintaining near-perfect uptime for *[what critical business functions depended on this?]*.

ACTION VERIFY

> I designed an event-driven architecture using AWS SQS for message queuing, Lambda for serverless processing, and DynamoDB for storage. *[What was your approach to handling backpressure and ensuring exactly-once processing?]* I implemented comprehensive error handling with *[what retry and dead letter queue strategies?]* and built monitoring dashboards using *[what AWS monitoring tools?]*. *[How did you test this at scale before production deployment?]*

RESULT FROM RESUME

> The pipeline successfully processes 2B+ events per day with 99.99% uptime. *[What was the business impact? Cost savings from the serverless approach? New capabilities this enabled? How did stakeholders react to this achievement?]*

Before You Use This Story

- [ ] Clarify what types of events were being processed and their business value
- [ ] Add details about the cost and performance comparison to the old system
- [ ] Specify how you achieved 99.99% uptime - what failure modes did you account for?

Likely Follow-up Questions — Prepare Your Answers

"How did you handle data consistency and ordering across this distributed pipeline?"

> *Discuss your understanding of eventual consistency, partitioning strategies, and ordering guarantees in distributed systems*

"What was your strategy for monitoring and alerting on a system processing 2B events daily?"

> *Show how you think about meaningful metrics vs noise, setting thresholds, and operational excellence*

"How did you test this system's behavior under failure conditions?"

> *Describe chaos engineering, load testing, and failure scenario planning - show proactive thinking*

3 Team Leadership and Development

Hire and Develop the Best | Earn Trust

✓ FROM RESUME What we know for certain

"Led technical design reviews for a team of 8 engineers; mentored 3 junior engineers to promotion"

USE THIS STORY FOR

Tell me about a time you developed someone / Describe your leadership style / Give an example of when you had to earn trust

‹ DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> As a *[your title]* at *[company name]*, I was leading technical design reviews for a team of 8 engineers working on *[what product/system?]*. The team had varying levels of experience, and we had 3 junior engineers who were *[what challenges were they facing - struggling with system design, code quality, career progression?]*.

TASK VERIFY

> I needed to ensure our design reviews were effective for maintaining code quality while also creating development opportunities for the junior engineers. Additionally, I took on the goal of mentoring the 3 junior engineers toward promotion within *[what timeframe?]*.

ACTION VERIFY

> I restructured our design review process to be more educational, having junior engineers *[present their designs? shadow senior reviews?]*. For mentoring, I *[what specific approach - weekly 1:1s, code review partnerships, project assignments?]*. I created individual development plans focusing on *[what skills - system design, coding best practices, communication?]* and gave them ownership of *[what types of projects or responsibilities?]*. *[How did you measure their progress and adjust your approach?]*

RESULT FROM RESUME

> All 3 junior engineers achieved promotion to the next level. *[What was the timeline? What specific growth did you observe? How did this impact team dynamics and productivity? Did you receive recognition for this mentorship?]*

Before You Use This Story

- [ ] Specify the promotion timeline and what levels they moved between
- [ ] Add concrete examples of how the engineers grew under your mentorship
- [ ] Detail the improvements you made to the design review process and their impact

Likely Follow-up Questions — Prepare Your Answers

"How did you tailor your mentoring approach to each individual's needs?"

> *Show you understand people are different - discuss assessing learning styles, strengths/weaknesses, career goals*

"What was the most challenging aspect of leading design reviews with mixed experience levels?"

> *Discuss balancing thorough review with learning opportunities, managing time, ensuring participation*

"How did you measure the success of your mentoring beyond promotions?"

> *Think about code quality, confidence, independence, knowledge sharing - show holistic view of development*

4 Operational Excellence Ownership

Ownership · Dive Deep

✓ FROM RESUME What we know for certain

"Owned full on-call rotation; reduced MTTD from 14 minutes to 4 minutes through improved CloudWatch alerting"

USE THIS STORY FOR

Tell me about a time you took ownership / Describe when you improved a process / Give an example of going above and beyond your role

DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company name]*, I took full ownership of the on-call rotation for *[what system/service?]* that was critical for *[what business function?]*. The team was experiencing *[what pain points - frequent pages, long resolution times, unclear alerts?]*, and our Mean Time to Detection was 14 minutes, which was impacting *[customer experience/SLA compliance?]*.

TASK VERIFY

> I needed to completely overhaul our monitoring and alerting strategy to reduce MTTD while also making the on-call experience more manageable for the team. *[Was there a specific incident or deadline driving this urgency?]*

ACTION VERIFY

> I conducted a thorough analysis of our existing CloudWatch alerts to identify *[what problems - false positives, missing coverage, unclear signals?]*. I redesigned the alerting strategy by *[what specific changes - new metrics, better thresholds, alert prioritization?]* and implemented *[what automation or runbook improvements?]*. I also *[how did you improve the on-call process beyond just alerting - dashboards, documentation, escalation procedures?]*. *[How did you test and validate these improvements?]*

RESULT FROM RESUME

> MTTD improved from 14 minutes to 4 minutes through the enhanced CloudWatch alerting. *[What was the impact on system reliability? Team morale? Customer satisfaction? Did you share these improvements with other teams? Any recognition received?]*

Before You Use This Story

- [ ] Add the improvement in Mean Time to Recovery (MTTR) in addition to MTTD
- [ ] Specify what types of issues the improved alerting helped catch faster
- [ ] Quantify the reduction in false positive alerts or on-call burden

Likely Follow-up Questions — Prepare Your Answers

"What was your methodology for determining which alerts were actually valuable?"

> *Discuss analysis of historical incidents, alert fatigue, signal vs noise - show data-driven approach*

"How did you balance comprehensive monitoring with alert fatigue?"

> *Show understanding of operational excellence principles - meaningful alerts, proper escalation, team sustainability*

"What other operational improvements did this work enable for your team?"

> *Think beyond just alerting - incident response, system reliability, team confidence, proactive vs reactive work*

5 Customer-Driven Service Redesign

Customer Obsession | Deliver Results

✓ FROM RESUME What we know for certain

"Drove customer obsession initiative to rewrite search indexing service, cutting customer-facing errors by 62%"

USE THIS STORY FOR

Tell me about a time you were customer obsessed / Describe when you delivered results under pressure / Give an example of improving customer experience

DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company name]*, our search indexing service was generating significant customer-facing errors that were *[what was the customer impact - complaints, lost transactions, etc?]*. *[What triggered your awareness of this issue - metrics dashboard, customer feedback, incident reports?]*

TASK VERIFY

> As *[your role]*, I needed to lead a complete rewrite of the search indexing service with customer obsession as the driving principle. *[What was your specific mandate - timeline, constraints, team size?]*

ACTION VERIFY

> I started by *[how did you gather customer pain points - interviews, data analysis, support tickets?]*. Then I *[what was your technical approach - new architecture, different algorithms, infrastructure changes?]*. Throughout the project, I *[how did you keep customer needs central - regular feedback loops, A/B testing, customer advisory input?]*

RESULT FROM RESUME

> We successfully cut customer-facing errors by 62%. *[What was the immediate customer reaction? Did this lead to recognition, promotion, or expanded responsibilities? How did this impact the broader product strategy?]*

Before You Use This Story

- [ ] What was the baseline error rate before your initiative?
- [ ] Add specific customer pain points that drove your obsession
- [ ] Include the timeline - how long did this rewrite take?
- [ ] Quantify team size and stakeholders you coordinated with

Likely Follow-up Questions — Prepare Your Answers

"How did you prioritize which customer pain points to address first?"

> *Think about your decision-making framework - data analysis, customer impact severity, technical feasibility, or business value*

"What resistance did you face to doing a complete rewrite versus incremental fixes?"

> *Focus on how you built the business case and convinced stakeholders that customer obsession required bold action*

"How did you ensure the rewrite wouldn't introduce new customer-facing issues?"

> *Discuss your risk mitigation strategies - testing approaches, rollout plan, monitoring, rollback procedures*

6 Monolith to Microservices Migration

Invent and Simplify | Think Big

✓ FROM RESUME What we know for certain

"Migrated monolith to Kubernetes-orchestrated microservices architecture, reducing deployment time by 70%"

USE THIS STORY FOR

Tell me about a time you simplified a complex system / Describe a major technical transformation you led / Give an example of driving architectural change

‹ DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company]*, we were operating with a monolithic architecture that was *[what specific problems - slow deployments, scaling issues, team bottlenecks?]*. *[What was the business impact - delayed features, increased costs, developer frustration?]*

TASK VERIFY

> I was tasked with *[were you assigned this or did you propose it?]* migrating our monolith to a Kubernetes-orchestrated microservices architecture. *[What was the scope - how many services, which parts of the system, what timeline?]*

ACTION VERIFY

> I began by *[how did you plan the migration - service boundaries analysis, dependency mapping, phased approach?]*. For the technical execution, I *[what was your containerization strategy, Kubernetes setup, data migration approach?]*. To manage complexity, I *[how did you coordinate across teams, handle service communication, ensure data consistency?]*

RESULT FROM RESUME

> The migration successfully reduced deployment time by 70%. *[What other benefits emerged - improved scalability, team velocity, system reliability? How did this change your team's development practices? Did this become a model for other teams?]*

Before You Use This Story

- [ ] What was the original deployment time before the migration?
- [ ] Add the number of microservices you broke the monolith into
- [ ] Include the migration timeline and any major milestones
- [ ] Specify your role - were you the technical lead, architect, or implementer?

Likely Follow-up Questions — Prepare Your Answers

"How did you decide where to draw service boundaries when breaking up the monolith?"

> *Explain your analysis process - domain modeling, data flow analysis, team structure, or business capabilities*

"What was the biggest technical challenge during the migration?"

> *Think about data consistency, service communication, transaction management, or operational complexity*

"How did you handle the operational overhead of managing multiple services?"

> *Discuss monitoring, logging, debugging distributed systems, and the tooling/processes you put in place*

7 Quality Engineering Standards

Insist on the Highest Standards | Deliver Results

✓ FROM RESUME What we know for certain

"Implemented CI/CD pipelines using Jenkins and GitHub Actions; achieved 95% test coverage across owned services"

USE THIS STORY FOR

Tell me about a time you insisted on high standards / Describe when you improved code quality / Give an example of implementing best practices

‹ DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company]*, our code quality and deployment processes were *[what specific issues - low test coverage, manual deployments, frequent production bugs?]*. *[What was driving the need for change - incidents, technical debt, team growth?]*

TASK VERIFY

> I took ownership of implementing comprehensive CI/CD pipelines and establishing quality standards across *[how many services or what scope?]*. My goal was to achieve high test coverage and reliable automated deployments.

ACTION VERIFY

> I designed and implemented CI/CD pipelines using *[why did you choose Jenkins and GitHub Actions specifically?]*. For test coverage, I *[what was your testing strategy - unit tests, integration tests, end-to-end tests?]*. To drive adoption, I *[how did you get team buy-in, handle resistance, provide training?]*

RESULT FROM RESUME

> I successfully achieved 95% test coverage across owned services. *[What was the impact on deployment frequency, bug rates, or developer productivity? Did other teams adopt your approach? How did leadership recognize this work?]*

Before You Use This Story

- [ ] What was the starting test coverage percentage before your initiative?
- [ ] Add specific metrics on deployment frequency improvement
- [ ] Include the number of services or codebase size this covered
- [ ] Specify what types of tests made up the 95% coverage

Likely Follow-up Questions — Prepare Your Answers

"How did you convince developers to write more tests when they were focused on feature delivery?"

> *Think about how you made the business case, addressed time concerns, and created incentives or accountability*

"What was your strategy for achieving 95% coverage without just writing meaningless tests?"

> *Discuss code quality metrics, test effectiveness, coverage types, and how you balanced quantity with quality*

"How do you maintain these high standards as new team members join?"

> *Consider onboarding processes, code review practices, automated enforcement, and cultural elements*

8 Search Technology Integration

Learn and Be Curious | Deliver Results

✓ FROM RESUME What we know for certain

"Integrated Elasticsearch for full-text property search, improving search relevance score by 28%"

USE THIS STORY FOR

Tell me about a time you learned a new technology / Describe when you improved an existing system / Give an example of solving a technical challenge

◦ DRAFT Plausible story — review and personalize

We've written a realistic version based on your resume. Read through it, correct anything that's wrong, and fill in the blanks marked with [brackets].

SITUATION VERIFY

> At *[company]*, our property search functionality was *[what specific problems - poor relevance, slow performance, limited search capabilities?]*. Users were *[what was the user experience issue - couldn't find properties, irrelevant results, frustration?]*

TASK VERIFY

> I needed to integrate Elasticsearch to implement full-text property search and significantly improve search relevance. *[Were you familiar with Elasticsearch beforehand, or was this a learning challenge? What was your timeline?]*

ACTION VERIFY

> Since *[was this new technology for you?]*, I started by *[how did you learn Elasticsearch - documentation, tutorials, experimentation?]*. For the integration, I *[what was your implementation approach - data modeling, indexing strategy, search algorithms?]*. To optimize relevance, I *[how did you tune scoring, handle property-specific search requirements, test relevance?]*

RESULT FROM RESUME

> The integration improved search relevance score by 28%. *[How did you measure relevance - user behavior, click-through rates, search success metrics? What was the user feedback? Did this lead to other search improvements or expanded responsibilities?]*

Before You Use This Story

- [ ] Define how search relevance score was measured - what specific metric?
- [ ] Add the baseline relevance score before Elasticsearch integration
- [ ] Include how long it took you to learn and implement Elasticsearch
- [ ] Specify the size and complexity of the property dataset you indexed

Likely Follow-up Questions — Prepare Your Answers

"How did you approach learning Elasticsearch quickly enough to implement this successfully?"

> *Discuss your learning strategy - hands-on experimentation, documentation study, seeking mentorship, or building proof-of-concepts*

"What specific Elasticsearch features did you leverage to improve property search relevance?"

> *Think about scoring algorithms, analyzers, filters, boosting, or custom relevance tuning you implemented*

"How did you validate that the 28% improvement in relevance score actually translated to better user experience?"

> *Consider A/B testing, user behavior metrics, conversion rates, or qualitative feedback you gathered*

Build your own story

Your 8 stories cover your strongest proof points — use this template whenever a new experience comes to mind before your interview.

Start with a real resume bullet or achievement. Don't start with a story idea — start with a fact. A metric, a deliverable, a result you can stand behind. Everything else builds from there.

✓ ANCHOR The real achievement — copy from your resume or write it in one sentence

STORY TITLE

COMPETENCY TAGS (PICK 1–2)

USE THIS STORY FOR — WHAT INTERVIEW QUESTIONS DOES IT ANSWER?

WHICH AMAZON VALUE DOES THIS BEST DEMONSTRATE?

Customer Obsession | Ownership | Invent and Simplify | Are Right, A Lot | Learn and Be Curious | Hire and Develop the Best | Insist on the Highest Standards | Think Big | Bias for Action | Frugality | Earn Trust | Dive Deep | Have Backbone; Disagree and Commit | Deliver Results | Strive to be Earth's Best Employer | Success and Scale Bring Broad Responsibility

Circle one — be honest. If it's split between two, pick the one the story demonstrates most clearly.

SITUATION Draft

Set the context. What was the state of things before you acted? Keep to 2–3 sentences. Use [brackets] for anything you're not 100% certain about yet.

TASK Draft

What were you specifically responsible for? Why you, not someone else?

ACTION Draft

What did you specifically do? Name your decisions, not just activities. Every claim needs a "why I chose this" — that's where interviewers probe hardest.

RESULT ✓ Anchor here

Start with the metric from your anchor above — that's your verified fact. Then add business impact, recognition, or follow-on effects.

STRESS-TEST WITH AI

Once you've drafted your story, paste this into Claude or ChatGPT:

> *"Act as a Amazon interviewer. I'm going to tell you a STAR story. After I finish, push back with 3 follow-up questions that test whether my answer is specific, credible, and genuinely demonstrates strong performance for this company. Be tough."*

Before you use this story

- [ ] Can you state the result metric from memory, without checking notes?
- [ ] In the Action, can you explain every decision and why you made it — not just what you did?
- [ ] Have you practiced this out loud at least once, timing it at 2–3 minutes?
- [ ] If the interviewer asks "what would you do differently?" — do you have an honest answer ready?

Interviewers won't ask the exact questions we prepared for — but if your story is solid, you can answer any version of the question. The goal isn't to memorise. It's to know the beats so you can deliver naturally.

6

What They're Testing — And How to Answer It

12 question patterns decoded — what's really being assessed, and how to answer any version of each question.

COMPANY Derived from Amazon interview structure

ROLE Standard for Software Engineer interviews

JD Derived from your job description

HOW TO USE THIS SECTION

These 12 questions were built specifically for your Amazon SDE interview. The distribution — 4 coding / 1 operational / 4 behavioral / 3 system design — reflects how Amazon actually structures this interview at your level, based on their published evaluation criteria and hiring patterns.

Each question includes three layers of prep intelligence: what the interviewer is actually evaluating beneath the surface, what a strong answer demonstrates, and the patterns that cause candidates to fall short.

Work through every question before your interview. For behavioral questions, draft your answer using the Story Builder in Section 5 — then practice saying it out loud until the delivery feels natural.

1 Technical — System Design

ROLE Probing candidate's actual distributed systems experience

> **"I see you architected an event-driven data pipeline processing 2B+ events per day at Microsoft. Walk me through your design decisions for handling backpressure when downstream services couldn't keep up with the event volume. How did you ensure data consistency during service failures?"**

WHAT THEY'RE REALLY ASKING

This question evaluates your real-world distributed systems expertise beyond resume claims. The interviewer wants to see if you understand the nuanced engineering challenges of high-scale event processing—specifically backpressure handling and failure recovery patterns that only come from hands-on experience building production systems.

WHAT GREAT LOOKS LIKE

- Specific backpressure strategies: Mentions concrete approaches like circuit breakers, exponential backoff, or queue-based buffering with overflow policies
- Failure isolation design: Describes bulkhead patterns, dead letter queues, or replay mechanisms that prevent cascade failures
- Data consistency trade-offs: Articulates specific consistency models (eventual vs strong) and when to use each in event-driven architectures
- Monitoring and observability: Details how they tracked system health with metrics like queue depth, processing lag, and error rates

RED FLAGS

- Vague theoretical answers: Discusses backpressure conceptually without specific implementation details or technologies used
- Single point of failure design: Proposes solutions that don't account for partial system failures or network partitions
- Ignoring consistency requirements: Focuses only on throughput without addressing data integrity during failures
- No operational perspective: Lacks discussion of monitoring, alerting, or how they detected and resolved issues in production

YOUR PREP

Use your Billion-Scale Data Pipeline story (Story 2) as your primary example, focusing on the specific backpressure and failure scenarios you encountered. Frame your technical decisions around the high standards LP—explain why you chose certain consistency models over others and how you insisted on robust failure handling even when it added complexity.

PRACTICE WITH AI

Act as an Amazon interviewer asking about event-driven architecture. Probe deeply into my specific implementation choices for backpressure and failure handling, asking follow-up questions about edge cases and production incidents I faced.

2 Coding — Algorithms

ROLE Algorithm problem based on candidate's performance optimization work

> **"You mentioned reducing p99 latency in your microservices at Microsoft. Write a function that finds the 99th percentile latency from a stream of response times. The stream is too large to fit in memory, and you need to handle millions of requests per second with minimal memory overhead."**

WHAT THEY'RE REALLY ASKING

This coding question tests your ability to translate real performance optimization experience into algorithmic solutions. The interviewer wants to see if you can design memory-efficient streaming algorithms while understanding the practical constraints of high-throughput systems—not just textbook algorithms, but production-ready code.

WHAT GREAT LOOKS LIKE

- Streaming algorithm approach: Implements reservoir sampling or approximate quantile algorithms (like t-digest or P2) rather than trying to store all values
- Memory complexity awareness: Explicitly discusses space-time trade-offs and why exact solutions won't work at scale
- Production considerations: Addresses thread safety, numerical precision issues, and how to handle bursty traffic patterns
- Code clarity and edge cases: Writes clean, testable code that handles boundary conditions like empty streams or extreme outliers

RED FLAGS

- Memory-intensive solutions: Attempts to store all values in arrays or tries to sort the entire dataset
- Ignores streaming constraint: Designs solutions that require multiple passes over the data or assume finite datasets
- No scalability discussion: Focuses only on correctness without considering memory usage or performance at millions of requests/second
- Incomplete implementation: Provides pseudocode or skips important details like initialization or error handling

YOUR PREP

Draw from your High-Scale Microservices Performance story (Story 1) to provide context for why p99 latency matters in practice. Before coding, mention specific performance challenges you faced and how percentile calculations helped you optimize systems—this shows you understand the real-world application of the algorithm.

PRACTICE WITH AI

Act as an Amazon interviewer giving me a streaming percentile calculation problem. Push me to optimize for memory usage and throughput, asking follow-up questions about thread safety and handling traffic spikes in production systems.

3 Technical — System Design

JD Team-specific scalability challenge for Search & Discovery

> **"Our Search & Discovery team needs to redesign our product ranking system to handle Black Friday traffic - 10x normal query volume with sub-100ms response times. How would you architect this using AWS services, considering we need to maintain search relevance while scaling horizontally?"**

From JD: *Design, build, and maintain highly scalable distributed systems*

WHAT THEY'RE REALLY ASKING

This system design question evaluates your ability to solve real Amazon-scale problems by designing search infrastructure that handles massive traffic spikes while maintaining quality. The interviewer is assessing your understanding of AWS services, search relevance algorithms, and horizontal scaling patterns specific to e-commerce search systems.

WHAT GREAT LOOKS LIKE

- Multi-tier caching strategy: Designs intelligent caching layers using ElastiCache and CloudFront with cache warming strategies for predictable spikes
- Search-specific scaling: Proposes elasticsearch cluster auto-scaling, index partitioning strategies, and relevance score caching for popular queries
- AWS service integration: Leverages services like Application Load Balancer, Auto Scaling Groups, and potentially SQS for async processing of search analytics
- Relevance preservation: Addresses how to maintain search quality during scale-out, possibly through machine learning model serving or pre-computed rankings

RED FLAGS

- Generic scaling solutions: Proposes database sharding or basic load balancing without understanding search-specific challenges
- Ignoring relevance requirements: Focuses only on throughput without addressing how search quality is maintained during horizontal scaling
- Poor AWS service selection: Misunderstands AWS services or proposes inappropriate solutions (like using RDS for search indices)
- No monitoring strategy: Fails to discuss how to measure search quality degradation or system performance during traffic spikes

YOUR PREP

Research Amazon's Search & Discovery domain focusing on how e-commerce search differs from web search—understand concepts like conversion-driven ranking, catalog freshness, and personalization at scale. Study AWS search-related services like OpenSearch Service, and think about how search relevance algorithms need to adapt during traffic spikes when computational resources are constrained.

PRACTICE WITH AI

Act as an Amazon Search & Discovery team interviewer. Challenge my system design for search ranking scalability, probing specifically about maintaining search relevance quality during 10x traffic spikes and asking detailed questions about AWS service choices.

4 Culture Fit — Are Right, A Lot — Are Right, A Lot COMPANY Amazon Leadership Principle assessment

> **"Tell me about a time when you had to make a difficult technical decision that affected your entire team, but you didn't have complete information. How did you approach the decision-making process, and what was the outcome?"**

WHAT THEY'RE REALLY ASKING

This behavioral question assesses the 'Are Right, A Lot' leadership principle by evaluating your decision-making process under uncertainty. Amazon wants to see if you can make high-quality decisions with incomplete information, learn from outcomes, and demonstrate the judgment that scales with increased responsibility and ambiguous situations.

WHAT GREAT LOOKS LIKE

- Structured decision framework: Describes a clear process for gathering available data, identifying assumptions, and weighing trade-offs systematically
- Stakeholder consultation: Shows how they leveraged diverse perspectives and expertise to fill information gaps without getting paralyzed by analysis
- Reversible vs irreversible distinction: Demonstrates understanding of when to make fast decisions vs when to invest more time in analysis based on consequence severity
- Learning integration: Explains what they learned from the outcome and how it improved their future decision-making process

RED FLAGS

- Decision avoidance: Describes trying to gather perfect information or delaying decisions until all uncertainty was removed
- Gut-feeling only: Makes decisions based purely on intuition without any systematic analysis or data gathering
- No ownership of outcome: Blames external factors for negative results or doesn't demonstrate learning from the experience
- Single-perspective bias: Failed to seek diverse input or consider alternative viewpoints before making the decision

YOUR PREP

This aligns perfectly with Amazon's 'Are Right, A Lot' principle—focus on demonstrating judgment and decision-making velocity. Choose from your Team Leadership story (Story 3) or Operational Excellence story (Story 4) where you had to make architectural or process decisions without complete data. Emphasize your systematic approach to uncertainty and how you balanced speed with accuracy.

PRACTICE WITH AI

Act as an Amazon interviewer assessing the 'Are Right, A Lot' leadership principle. Ask follow-up questions about my decision-making process, what information I wished I had, and how I validated my assumptions after making the decision.

5 Coding — Algorithms

ROLE Two-sum variant with practical e-commerce context

> **"You're given an array of integers representing customer purchase amounts. Write a function to find all pairs of purchases that sum to a target gift card value. Optimize for the case where customers frequently query different target values on the same purchase history."**

WHAT THEY'RE REALLY ASKING

This tests your ability to recognize classic algorithmic patterns while optimizing for real-world constraints. The interviewer wants to see if you can implement a clean two-sum solution, then think beyond the basic problem to handle the optimization requirement for repeated queries on the same dataset.

WHAT GREAT LOOKS LIKE

- Clean Implementation: Starts with hash map approach for O(n) single query, clearly explains the algorithm
- Optimization Recognition: Identifies that frequent queries on same data means preprocessing is worthwhile
- Practical Solution: Suggests sorting the array once, then using two pointers for O(log n) per query after O(n log n) preprocessing
- Edge Case Handling: Discusses duplicates, empty arrays, and negative numbers without being prompted

RED FLAGS

- Brute Force Tunnel Vision: Jumps to $O(n^2)$ nested loops without considering hash map approach
- Missing Optimization: Solves basic two-sum but ignores the 'frequent queries' constraint entirely
- Overengineering: Suggests complex caching or database solutions when simple preprocessing suffices
- Poor Communication: Codes silently without explaining approach or walking through examples

YOUR PREP

Practice the classic two-sum problem until you can code it cleanly in 5 minutes, then focus on the optimization twist. The key insight is recognizing when preprocessing makes sense for repeated operations - this pattern appears frequently in Amazon's high-volume systems.

PRACTICE WITH AI

Act as an Amazon interviewer asking the two-sum variant question. Probe my optimization thinking by asking follow-up questions about performance trade-offs and memory usage for different query patterns.

6 Technical — Operational Excellence

JD Day-to-day operational challenges for the Search team

> **"We're seeing intermittent 5xx errors in our search API that happen once every few hours, affecting about 0.1% of requests. Walk me through how you would investigate this issue. What monitoring and alerting would you set up to catch this earlier, and how would you build a runbook for the on-call engineer?"**

From JD: *Define and drive operational excellence through monitoring, alerting, and on-call practices*

WHAT THEY'RE REALLY ASKING

This evaluates your operational maturity and systematic debugging approach for production systems. The interviewer wants to see if you can methodically investigate intermittent issues, design comprehensive monitoring, and create actionable runbooks - core skills for Amazon's operational excellence culture.

WHAT GREAT LOOKS LIKE

- Systematic Investigation: Starts with logs, metrics, and traces; looks for patterns in timing, geography, or request characteristics
- Comprehensive Monitoring: Suggests error rate dashboards, latency percentiles, and correlation with deployment/traffic patterns
- Proactive Alerting: Designs alerts that catch 0.1% error rates without false positives; considers anomaly detection
- Actionable Runbooks: Creates step-by-step debugging guide with escalation paths and rollback procedures

RED FLAGS

- Random Walk Debugging: Suggests checking things without a systematic approach or hypothesis
- Alert Fatigue: Proposes overly sensitive alerts that would trigger constantly for normal fluctuations
- Vague Monitoring: Mentions dashboards without specifying what metrics or thresholds to track
- Incomplete Runbooks: Creates investigation steps but forgets escalation procedures or mitigation actions

YOUR PREP

Research Amazon's Search & Discovery team challenges - they handle massive query volumes with strict latency requirements. Study how intermittent errors manifest in distributed search systems, particularly around index updates, caching layers, and query routing. Focus on the observability tools and practices used for large-scale search infrastructure.

PRACTICE WITH AI

Act as an Amazon Search team interviewer discussing this 5xx error scenario. Challenge my investigation approach by asking about specific tools, metrics, and how I'd differentiate between various root cause hypotheses.

7 Culture Fit — Learn and Be Curious — Learn and Be Curious

COMPANY Amazon Leadership Principle assessment

> **"Describe a situation where you had to learn a completely new technology or domain quickly to deliver a project. What was your approach to ramping up, and how did you ensure you maintained high quality while learning?"**

WHAT THEY'RE REALLY ASKING

This assesses your alignment with Amazon's 'Learn and Be Curious' leadership principle and your ability to deliver results while acquiring new skills. The interviewer wants to see your learning methodology, how you balance speed with quality, and whether you can maintain Amazon's high standards during knowledge acquisition.

WHAT GREAT LOOKS LIKE

- Structured Learning Approach: Describes systematic method like documentation review, finding mentors, building small prototypes
- Quality Safeguards: Explains how they maintained standards through code reviews, testing, or pair programming while learning
- Delivery Focus: Shows they balanced learning time with project deadlines and found ways to de-risk the delivery
- Knowledge Sharing: Demonstrates they documented learnings or taught others, amplifying the investment in learning

RED FLAGS

- Passive Learning: Describes only reading documentation or watching videos without hands-on practice
- Quality Compromise: Admits to cutting corners on testing or best practices due to learning curve
- Isolation Tendency: Learned entirely alone without seeking help from experts or community resources
- No Learning Strategy: Took a scattered approach without clear milestones or validation checkpoints

YOUR PREP

Use Story 8 (Search Technology Integration) where you learned new technology to improve an existing system. Frame this around Amazon's expectation that senior engineers continuously expand their technical breadth while maintaining delivery excellence. Emphasize how you balanced curiosity-driven learning with the bias for action that Amazon values.

PRACTICE WITH AI

Act as an Amazon interviewer evaluating the Learn and Be Curious leadership principle. Probe how I balanced learning speed with delivery quality, and ask follow-up questions about knowledge retention and sharing.

8 Technical — System Design

COMPANY Amazon-scale personalization challenge

> **"Design a system to personalize product recommendations for Amazon's 300M+ active customers. The system needs to process real-time user interactions, handle cold start problems for new users, and serve recommendations with sub-50ms latency globally. Consider both the ML pipeline and the serving infrastructure."**

WHAT THEY'RE REALLY ASKING

This tests your ability to design systems at Amazon's massive scale while balancing multiple complex requirements. The interviewer wants to see if you can break down a multi-faceted problem, make reasonable trade-offs between latency/accuracy/cost, and design both the ML and infrastructure components with Amazon's operational principles in mind.

WHAT GREAT LOOKS LIKE

- Scale Recognition: Acknowledges 300M users requires distributed architecture with regional deployment and caching strategies
- Multi-Stage Design: Separates batch training pipeline, real-time feature processing, and low-latency serving infrastructure
- Cold Start Solutions: Proposes fallback strategies like popularity-based recommendations and demographic clustering for new users
- Latency Optimization: Designs precomputed recommendations with real-time adjustment layer to meet sub-50ms requirement

RED FLAGS

- Monolithic Thinking: Proposes single system trying to handle both ML training and real-time serving
- Latency Ignorance: Suggests complex real-time ML computations without considering 50ms constraint
- Scale Blindness: Designs for thousands of users instead of hundreds of millions; misses regional distribution needs
- Cold Start Amnesia: Focuses only on existing users and doesn't address new user recommendation problem

YOUR PREP

Study Amazon's actual recommendation systems and their emphasis on customer obsession through personalization. Focus on how Amazon's scale requires separation of concerns between ML training and serving infrastructure. Understand their approach to global content delivery and how they balance recommendation relevance with system performance.

PRACTICE WITH AI

Act as an Amazon interviewer evaluating this system design. Challenge my scalability assumptions and ask detailed questions about handling peak traffic during events like Prime Day.

9 Coding — Algorithms

ROLE Classic search algorithm problem

> **"You have a sorted array that has been rotated at some unknown pivot point. Write a function to find a target value in this array. The algorithm should run in O(log n) time complexity. How would you handle duplicates?"**

WHAT THEY'RE REALLY ASKING

This tests your ability to recognize search patterns, optimize algorithms for efficiency, and handle edge cases systematically. The interviewer wants to see if you can identify the modified binary search approach, explain the logic clearly, and demonstrate thorough problem-solving by addressing complications like duplicates.

WHAT GREAT LOOKS LIKE

- Pattern Recognition: Immediately identifies this as a modified binary search problem and explains why O(log n) is achievable
- Clear Algorithm: Describes the process of comparing mid element with left/right bounds to determine which half is sorted
- Edge Case Handling: Addresses duplicates by explaining when you need to fall back to linear search (when arr[left] == arr[mid] == arr[right])
- Code Quality: Writes clean, bug-free code with proper boundary handling and explains the time complexity degradation with duplicates

RED FLAGS

- Brute Force Approach: Suggests O(n) linear search without recognizing the binary search opportunity
- Incomplete Logic: Gets the basic binary search concept but fails to properly handle the rotation logic or boundary conditions
- Ignores Duplicates: Doesn't address the duplicate case or gives an incorrect solution for handling them
- Poor Communication: Writes code without explaining the thought process or rushes through without validating the solution

YOUR PREP

Focus on structured problem-solving: start by clarifying the problem, identify why standard binary search needs modification, then walk through the logic of determining which half is sorted. Practice explaining your reasoning out loud as you code, and make sure you can handle the duplicate case where worst-case becomes O(n).

PRACTICE WITH AI

Act as an Amazon interviewer asking about rotated sorted array search. Start with the basic problem, then probe deeper on edge cases, ask me to trace through examples, and challenge me on the duplicates extension. Push for clean code and time complexity analysis.

10 Culture Fit — Ownership — Ownership

COMPANY Amazon Leadership Principle assessment

> **"Tell me about a time when you disagreed with your manager or team lead about a technical approach. How did you handle the situation, and what was the result?"**

WHAT THEY'RE REALLY ASKING

This evaluates your ability to demonstrate Ownership by advocating for what you believe is right while maintaining respect for leadership and team dynamics. Amazon wants to see that you'll speak up when you have conviction, can influence through data and reasoning, and take responsibility for outcomes regardless of the initial disagreement.

WHAT GREAT LOOKS LIKE

- Respectful Advocacy: Shows how you presented your technical concerns professionally with data, examples, or concrete reasoning
- Influence Without Authority: Demonstrates ability to persuade through technical merit rather than hierarchy or politics
- Constructive Outcome: Explains either how you convinced leadership, found a compromise, or committed fully once a decision was made
- Ownership of Results: Takes responsibility for the outcome and shows what you learned about technical leadership

RED FLAGS

- Insubordination: Comes across as argumentative, disrespectful, or unable to work within team structures
- Weak Conviction: Describes a trivial disagreement or fails to show you truly believed in your position
- Poor Communication: Couldn't articulate technical concerns effectively or resorted to emotion over data
- Lack of Follow-through: Disagreed but then disengaged, or couldn't commit to the team's final decision

YOUR PREP

Amazon's Ownership principle means you're expected to respectfully challenge decisions when you believe there's a better way. Think of a time you had strong technical conviction - focus on how you prepared your case with data, presented alternatives respectfully, and remained committed to the team regardless of the outcome. Emphasize that ownership means speaking up, not staying silent.

PRACTICE WITH AI

Act as an Amazon interviewer probing the Ownership leadership principle. Ask me about disagreeing with leadership, then dig deeper on how I influenced the situation, what data I used, and how I handled the aftermath. Push me on whether I truly took ownership or just complained.

11 Coding — Algorithms

ROLE Sliding window algorithm with search context

> **"Given a string containing customer search queries, write a function to find the longest substring without repeating characters. Then extend it to handle the case where we want to find the longest substring with at most k distinct characters."**

WHAT THEY'RE REALLY ASKING

This tests your mastery of sliding window algorithms and ability to build upon solutions incrementally. The interviewer wants to see efficient pattern recognition, clean implementation of the two-pointer technique, and your ability to extend basic algorithms to handle more complex variations systematically.

WHAT GREAT LOOKS LIKE

- Sliding Window Mastery: Immediately recognizes both problems as sliding window variations and explains the two-pointer approach
- Efficient Implementation: Uses HashSet/HashMap appropriately for O(n) solutions with proper window expansion and contraction logic
- Clear Extension: Shows how the k-distinct variation builds on the first solution using character frequency tracking
- Edge Case Awareness: Handles empty strings, k=0, and explains time/space complexity for both solutions

RED FLAGS

- Inefficient Approach: Uses nested loops or other $O(n^2)$ approaches instead of recognizing the sliding window pattern
- Incorrect Logic: Gets the basic sliding window concept but implements expansion/contraction incorrectly
- Struggles with Extension: Can't adapt the first solution to handle the k-distinct case or uses a completely different approach
- Poor Data Structure Choice: Uses inappropriate data structures that lead to inefficient solutions or overcomplicated code

YOUR PREP

Master the sliding window framework: expand the window until a constraint is violated, then contract from the left until valid again. For the basic problem, track characters in a HashSet; for k-distinct, use a HashMap to count frequencies. Practice explaining how the two-pointer technique maintains the window invariant throughout.

PRACTICE WITH AI

Act as an Amazon interviewer testing sliding window algorithms. Give me the longest substring problems, ask me to code both variations, trace through examples, and probe my understanding of when and why sliding window is the optimal approach.

12 Culture Fit — Customer Obsession — Customer Obsession

COMPANY Amazon Leadership Principle assessment

> **"Give me an example of when you had to make a trade-off between perfect engineering and meeting a critical deadline. How did you ensure the customer wasn't negatively impacted by your decision?"**

WHAT THEY'RE REALLY ASKING

This assesses your ability to balance Amazon's Customer Obsession principle with engineering constraints and business pressures. The interviewer wants to see that you make customer-centric decisions under pressure, can articulate trade-offs clearly, and implement safeguards to minimize customer impact when taking shortcuts.

WHAT GREAT LOOKS LIKE

- Customer-First Reasoning: Clearly explains how customer impact drove the decision-making process and trade-off evaluation
- Smart Risk Management: Describes specific measures taken to minimize customer exposure (feature flags, monitoring, rollback plans)
- Quantified Impact: Provides concrete metrics or examples of how the approach protected or benefited customers despite the compromise
- Learning Integration: Shows how the experience influenced future technical debt management or process improvements

RED FLAGS

- Business Over Customer: Focuses primarily on meeting deadlines or business metrics without emphasizing customer protection
- Reckless Shortcuts: Describes cutting corners without implementing proper safeguards, monitoring, or mitigation strategies
- Vague Examples: Can't provide specific details about customer impact measurement or protection mechanisms
- No Long-term View: Fails to address technical debt paydown or doesn't show learning from the trade-off decision

YOUR PREP

Amazon's Customer Obsession means even when you compromise on engineering perfection, customer experience remains the top priority. Use your Customer-Driven Service Redesign or Operational Excellence Ownership stories, emphasizing how you implemented monitoring, gradual rollouts, or other customer protection measures. Show that 'good enough' engineering can still deliver excellent customer outcomes when done thoughtfully.

PRACTICE WITH AI

Act as an Amazon interviewer focusing on Customer Obsession. Ask me about engineering trade-offs under pressure, then dig deep into how I protected customers, what safeguards I implemented, and how I measured customer impact throughout the process.

7

Scripts for Awkward Questions

How to handle gaps, weaknesses, and curveballs with confidence.

Your Gap Analysis

JD REQUIREMENT	YOUR RESUME EVIDENCE	STATUS
Experience designing and building distributed systems at scale with high-performance requirements	Built distributed microservices handling 500K+ requests/second at Microsoft Azure, architected event-driven data pipeline processing 2B+ events per day with 99.99% uptime, and migrated monolith to microservices architecture at Expedia	✓ Covered
Proficiency in Java or Python (both preferred) for production systems	Extensive professional experience with both Java and Python across all roles - Java (Spring Boot) at Expedia, Python (Django) at Zillow, and both Java and Python for microservices at Microsoft Azure	✓ Covered
Strong understanding of data structures, algorithms, and system design	Resume demonstrates system design experience through distributed systems work, but lacks explicit mention of algorithms and data structures depth or computer science fundamentals	⚠ Gap
Experience leading technical projects or mentoring engineers with cross-team leadership	Led technical design reviews for team of 8 engineers and mentored 3 junior engineers to promotion, but evidence shows only within-team leadership rather than cross-team or cross-functional leadership	⚠ Gap
Alignment with Amazon Leadership Principles (Customer Obsession, Ownership, Invent and Simplify, Bias for Action, Deliver Results)	Shows some customer obsession (rewriting search indexing service to reduce customer errors) and ownership (full on-call rotation), but lacks clear demonstration of Amazon-specific leadership principles and cultural alignment	⚠ Gap

Bridge Scripts for Your Gaps

1 Algorithms Depth

Re: Strong understanding of data structures, algorithms, and system design

WHY INTERVIEWERS WILL PROBE THIS

The interviewer may worry that while you can build systems, you lack the computer science fundamentals needed to optimize complex algorithms or solve challenging technical problems that require deep algorithmic thinking.

YOUR BRIDGE SCRIPT

You're right that my resume focuses more on system design than algorithms. While I use data structures and algorithms daily - like optimizing [specific example, e.g., 'hash maps for caching layers' or 'priority queues for task scheduling'] *- I'll admit I haven't had recent practice with advanced algorithmic challenges. However, my CS degree gave me a solid foundation, and I'm confident I can quickly refresh those skills. In fact, I've been working through* [specific preparation, e.g., 'LeetCode problems' or 'Algorithm Design Manual'] *to sharpen this area. What I bring is the ability to translate algorithmic solutions into production systems that actually scale.*

BEFORE YOU USE THIS SCRIPT, VERIFY:

- [ ] Can you give a specific example of using data structures/algorithms in your current work?
- [ ] What algorithmic concepts from your CS degree do you remember most clearly?
- [ ] Are you actually doing any algorithms practice, or is this something you plan to start?

PRACTICE WITH AI

Act as an Amazon interviewer asking about algorithms and data structures. Challenge my answer about my algorithmic background, and push back if my response sounds like I'm avoiding the technical depth or being overly rehearsed.

2 Cross-team Leadership

Re: Experience leading technical projects or mentoring engineers with cross-team leadership

WHY INTERVIEWERS WILL PROBE THIS

The interviewer may worry that your leadership experience is limited to managing within your own team, but this role requires influencing across teams, departments, and stakeholders where you don't have direct authority.

YOUR BRIDGE SCRIPT

Most of my formal leadership has been within my team, but I have experience influencing across teams. For example, when I [specific cross-team project, e.g., 'led the microservices migration at Expedia'] *, I had to coordinate with* [specific teams, e.g., 'DevOps, QA, and Product teams'] *to align on timelines and requirements. I also drove* [another example of influence without authority] *by building consensus through technical demos and clear documentation rather than through hierarchy. While I'm excited to take on more formal cross-team leadership responsibilities, I've learned that successful influence often comes from earning credibility through technical excellence and clear communication.*

BEFORE YOU USE THIS SCRIPT, VERIFY:

- [ ] Can you think of a specific project where you had to influence people outside your direct team?
- [ ] How did you handle disagreements or resistance when you didn't have authority?
- [ ] What techniques have you used to build consensus across different stakeholders?

PRACTICE WITH AI

Act as an Amazon interviewer focusing on cross-team leadership. Challenge my answer about influencing without authority, and push back if my examples sound too small-scale or if I'm avoiding the leadership gap.

Bridge Scripts for Your Gaps

3

Amazon Culture Fit

Re: Alignment with Amazon Leadership Principles (Customer Obsession, Ownership, Invent and Simplify, Bias for Action, Deliver Results)

WHY INTERVIEWERS WILL PROBE THIS

The interviewer needs to see that you understand and embody Amazon's unique leadership principles beyond just technical skills, and that you'll thrive in Amazon's high-ownership, customer-obsessed culture.

YOUR BRIDGE SCRIPT

I've been studying Amazon's Leadership Principles and can see how they align with my work style. The customer obsession principle resonates with me - like when I [specific customer-focused example from resume, e.g., 'rewrote the search indexing service to reduce customer errors by 62%'] *because I couldn't stand knowing our users were having a poor experience. For ownership, I've always believed in* [specific ownership example, e.g., 'taking full on-call responsibility' or 'driving projects end-to-end']*. What excites me about Amazon is the chance to operate at this level across everything I do, not just on special projects. I'm ready to raise the bar on how deeply I think about long-term customer impact.*

BEFORE YOU USE THIS SCRIPT, VERIFY:

- [ ] Can you give specific examples from your experience that demonstrate 2-3 Amazon Leadership Principles?
- [ ] What do you know about Amazon's culture beyond the Leadership Principles?
- [ ] Why do you want to work at Amazon specifically, beyond the technical challenges?

PRACTICE WITH AI

Act as an Amazon interviewer testing cultural fit and leadership principles. Challenge my understanding of Amazon's culture and push back if my examples sound generic or if I seem to only have surface-level knowledge of the leadership principles.

When You Don't Know the Answer

UNIVERSAL FRAMEWORK

The 4-Step Recovery

1. **Pause**
   2-3 seconds of silence is fine. Don't panic.
2. **Acknowledge**
   "That's a great question. I haven't encountered that exact scenario."
3. **Reason**
   "Here's how I'd think about it..."
4. **Anchor**
   "In a similar situation, I [relevant experience]..."

WHAT TO AVOID

- ✗ Bluffing or making up answers
- ✗ Getting visibly flustered
- ✗ Saying "I have no idea" and stopping
- ✗ Overexplaining why you don't know

PHRASES THAT WORK

"I haven't worked with that specific technology, but here's how I'd approach learning it..."

"That's outside my direct experience, but my instinct would be to..."

"I'd want to understand more about [X] before giving a definitive answer, but my initial thinking is..."

Curveball Questions

These questions are designed to test how you think under pressure. There's rarely a "right" answer — they're evaluating your reasoning process.

"Tell me about your biggest professional failure."

Why they ask: Testing ownership and learning agility — Amazon expects leaders who raise the bar on themselves, not just their teams.

FRAMEWORK

- → Choose a real failure with meaningful scope — not a minor setback
- → Own your role fully — no external blame
- → Show the specific lesson and how it changed your approach
- → Demonstrate you have applied that lesson since

⚠ Pick a failure that shows growth at scale. Bar Raisers will probe if it sounds rehearsed or too tidy.

"Tell me about a time you made a decision with incomplete data."

Why they ask: Testing Bias for Action and Are Right A Lot — Amazon expects leaders to act under ambiguity, not wait for perfect information.

FRAMEWORK

- → Describe the decision scope and why waiting was not an option
- → Show the framework you used to assess what you knew vs. what you did not
- → Explain the outcome and what you would do differently with hindsight
- → Demonstrate you can calibrate confidence to evidence

⚠ Have a real example where the stakes were meaningful and the data was genuinely ambiguous — not just missing one metric.

"Tell me about a time you disagreed with your manager."

Why they ask: Testing Have Backbone; Disagree and Commit — Amazon wants leaders who push back with data, then commit fully once a decision is made.

FRAMEWORK

- → Show you raised the disagreement directly and with data — not passively

- → Demonstrate you understood their reasoning, not just your own position
- → Show you committed fully once the decision was made
- → Reflect honestly on whether the outcome validated your concern or theirs

⚠ The Bar Raiser will probe whether you truly committed or quietly resisted. Have an honest answer ready.

Quick Reference: The Graceful Bridge

For any gap or weakness, remember: **Acknowledge → Pivot → Evidence**. Never deny a gap exists. Instead, show adjacent experience and genuine enthusiasm to grow. Interviewers expect gaps — they're evaluating how you handle them, not whether you're perfect.

8

Questions That Make Them Want You

Strategic questions to ask each interviewer type.

The questions you ask tell interviewers as much about you as your answers. Great questions demonstrate that you've **done your research**, you're **thinking strategically** about the role, and you're **evaluating them**, not just hoping to be chosen.

Use **2-3 questions per interview** — more than that feels like an interrogation. Choose based on who you're talking to.

For the Recruiter

Focus: Process, timeline, culture fit

"What does the interview process look like from here, and what's a realistic timeline?"

Why it works: Shows you're organized and serious about moving forward.

"What traits have you seen in candidates who really thrive here?"

Why it works: Gets insider perspective on culture fit — recruiters have pattern-matched hundreds of hires.

★ COMPANY-SPECIFIC

"In your experience screening SDE candidates, what differentiates the ones who successfully navigate Amazon's behavioral interviews from those who struggle with the Leadership Principles assessment?"

Why it works: Recruiter has observed many candidates go through the LP-heavy behavioral process and can share patterns of what works vs. what doesn't, which is valuable intelligence for preparation.

For the Hiring Manager

Focus: Role expectations, success metrics, team dynamics

"If I were crushing it in this role after 6 months, what would that look like?"

Why it works: Shows you're thinking about impact, not just tasks. Reveals their real priorities.

★ COMPANY-SPECIFIC

"Amazon talks about 'Insist on the Highest Standards' as a core principle. When your team is under pressure to ship quickly, how do you actually balance that with maintaining those relentlessly high standards in practice?"

Why it works: This gets at the real tension between speed and quality that hiring managers navigate daily, and how the LP manifests in actual decision-making rather than theory.

◆ ROLE-SPECIFIC

"The role mentions leading technical design reviews and mentoring junior engineers. Given my background leading design reviews for a team of 8 at Microsoft, what would be different about how Amazon approaches technical leadership at the SDE2 level?"

Why it works: Directly connects candidate's proven experience to a specific JD requirement while asking about Amazon's unique approach to technical leadership.

For Peer Interviewers

Focus: Day-to-day reality, collaboration, culture

"What do you wish you'd known before joining?"

Why it works: Invites honest perspective and shows you value candor.

★ COMPANY-SPECIFIC

"Amazon's written culture is pretty unique - starting meetings by reading six-page narratives instead of PowerPoint. As someone who's lived through that transition, what's it actually like day-to-day? Does it really change how decisions get made?"

Why it works: Only someone working there daily can speak to the reality of this cultural practice and whether it's as transformative as it sounds or just a different process.

◆ ROLE-SPECIFIC

"I see the role involves collaborating cross-functionally with product managers and partner teams. Coming from a similar setup at Microsoft Azure, I'm curious - how does that collaboration actually flow here when you're trying to balance Search & Discovery priorities with what other teams need?"

Why it works: Peer can give candid insight into the reality of cross-functional work and potential friction points from someone actually doing the collaboration.

For Executives / Skip-Level

Focus: Company direction, strategic importance, vision

"Where do you see this product/team in 2-3 years?"

Why it works: Shows you're thinking long-term and want to understand the strategic trajectory.

★ COMPANY-SPECIFIC

"Amazon's Leadership Principles include both 'Think Big' and 'Frugality' - seemingly opposing forces. From your perspective, how does Amazon resolve that tension strategically, especially as the company has grown from that garage to global scale?"

Why it works: Executive can speak to high-level strategic trade-offs and how the company navigates competing principles at scale, which shapes all teams' decision-making.

◆ ROLE-SPECIFIC

"Search & Discovery seems critical to Amazon's retail success - customers finding what they want drives everything. How does investment in this team's technical capabilities factor into Amazon's broader competitive strategy?"

Why it works: Executive understands the strategic importance of search technology to Amazon's business model and can explain why this role matters to the company's future.

For the Bar Raiser

Focus: Raising the bar for Amazon overall — not just this team. Has veto power over the hiring decision.

★ BAR RAISER SPECIFIC

"I've been thinking about how 'Customer Obsession' might create tension with engineering best practices - sometimes the fastest path to customer value isn't the most scalable technical solution. How should someone at my level navigate those trade-offs while still raising the bar for engineering excellence?"

Why it works: Demonstrates sophisticated thinking about LP tensions and bar-raising that the Bar Raiser evaluates - shows candidate thinks beyond just shipping features to long-term technical excellence.

◆ ROLE-SPECIFIC

"Given my experience building distributed systems at Microsoft and now potentially joining Amazon's Search team, I'm curious about your perspective on what technical leadership looks like when you're operating at Amazon's scale and standards. What does it mean to truly own end-to-end delivery in this environment?"

Why it works: Shows candidate is thinking at the Bar Raiser's level about what high performance looks like across Amazon, not just this team - demonstrates understanding of the bar they'd need to meet and raise.

Questions That Hurt Your Candidacy

Avoid asking: "What does your company do?" (shows no research) • "How soon can I get promoted?" (sounds entitled) • "What's the work-life balance like?" (ask instead: "How does the team approach deadlines?") • "Did I get the job?" (awkward) • Anything easily Googleable (shows no prep) • "I don't have any questions" (signals disinterest)

MAKE THESE YOUR OWN

The personalized questions above are based on your target company and role.

- → Don't read these verbatim — internalize the intent and ask in your own words
- → Reference recent news or product launches you've seen
- → Mention something from the interviewer's LinkedIn (if visible)
- → Connect your specific experience to their challenges
- → If you know someone who works there, ask what they'd want to know

9

A Handout That Closes the Deal

Your 30/60/90 day approach — tailored to Amazon and your background.

WHY THIS WORKS

Show How You Think, Not What You'll Deliver

A 30/60/90 day plan signals **strategic thinking** and **self-awareness**. But the best plans don't over-promise — they show how you'll approach the role based on your specific background, while leaving room for the reality that you don't yet know what it's like to work there.

We've tailored this plan to your experience and Amazon's expectations. Print the next page and bring it to your interview — or offer to send it afterward.

Your Printable Handout

The next page is a clean, one-page summary designed to leave with your interviewer.
It has your name on it — no Interview101 branding — so it looks like you created it.

→ Next Page

My 30/60/90 Day Approach

Jordan M. Chen
SDE

DAYS 1-30 — Learn

Build the foundation to contribute effectively

WHAT I'LL SEEK TO UNDERSTAND

- Search & Discovery architecture serving hundreds of millions customers globally
- Technical design review process and cross-functional collaboration with PM/design teams
- Customer obsession mindset: start with customer needs, work backwards

WHERE MY BACKGROUND HELPS

- High-scale microservices experience (500K+ RPS) accelerates understanding distributed search systems
- Proven on-call ownership and CloudWatch alerting expertise translates directly to operational excellence

MY MILESTONE

By Day 30, I can contribute meaningfully to technical design discussions and customer impact decisions

DAYS 31-60 — Build

Add value while still learning

WHERE I'LL LOOK TO ADD VALUE

- Apply event-driven architecture patterns to search indexing pipeline improvements
- Leverage operational excellence background to enhance monitoring and alerting
- Support junior engineers with mentoring based on promotion track record

WHAT I'M EXCITED TO LEARN

- Deepen system design patterns for search relevance and recommendation algorithms at Amazon scale

MY MILESTONE

By Day 60, I've shipped first feature with comprehensive operational ownership and monitoring

DAYS 61-90 — Own

Begin driving, not just supporting

WHERE I MIGHT ADD UNIQUE VALUE

- Begin leading microservices optimization initiatives leveraging 500K+ RPS experience
- Start owning technical design reviews for search infrastructure components
- Take initiative on cross-team API design based on REST/gRPC background

WHAT I'LL DIAGNOSE FIRST

- What distributed systems design patterns need strengthening in current architecture?
- Where are the highest-leverage algorithm optimization opportunities in search ranking?
- What does leadership see as top technical debt priorities?

MY MILESTONE

By Day 90, I've established technical leadership in one module and proposed first mechanism

---

This is my approach, not a rigid commitment — I'll adapt based on what I learn.

interview101 Interview Day Cheat Sheet

Jordan M. Chen
SDE at Amazon

YOUR 30-SECOND PITCH

Senior SDE with 7 years architecting high-scale distributed systems handling 500K+ RPS and 2B+ daily events on AWS. Proven track record building microservices with 99.99% uptime while mentoring teams and driving customer obsession. Ready to own features end-to-end at Amazon.

YOUR STORY BANK — READY TO USE

1. **High-Scale Microservices Performance**
   → Tell me about a time you optimized system performance.
2. **Billion-Scale Data Pipeline**
   → Tell me about a time you thought big.
3. **Team Leadership and Development**
   → Tell me about a time you developed someone.
4. **Operational Excellence Ownership**
   → Tell me about a time you took ownership.
5. **Customer-Driven Service Redesign**
   → Tell me about a time you were customer obsessed.
6. **Monolith to Microservices Migration**
   → Tell me about a time you simplified a complex system.
7. **Quality Engineering Standards**
   → Tell me about a time you insisted on high standards.
8. **Search Technology Integration**
   → Tell me about a time you learned a new technology.

KNOW ABOUT AMAZON

- → **Values:** Customer obsession, ownership, bias for action, high standards, think big.
- → **Recent:** Amazon evaluates SDE candidates heavily on behavioral fit with 16 Leadership Principles through STAR method.
- → **Interview:** Expect 4-6 hour virtual loop with Bar Raiser veto power. Each interviewer owns 2-3 LPs.
- → **Culture:** SDE 2s own features end-to-end, write narrative design docs, lead technical reviews, drive operational excellence.

YOUR TOP SELLING POINTS

- ✓ Proven high-scale distributed systems architecture
- ✓ Technical leadership with mentoring impact
- ✓ Customer-obsessed operational excellence champion
- ✓ Shipped microservices and event pipelines handling billions of operations daily with near-perfect reliability

IF THEY ASK ABOUT ALGORITHMS AND DATA STRUCTURES REFRESHER

Use data structures daily—hash maps for caching, priority queues for scheduling—with solid CS foundation. Currently practicing LeetCode to sharpen advanced algorithmic problem-solving. Strength is translating algorithmic solutions into production systems that scale.

CURVEBALL READY

"Tell me about your biggest failure."

Name failure clearly. Own it fully—no blame. Explain what you learned. Show what you changed. Connect to Ownership or Learn and Be Curious LP.

QUESTIONS TO ASK

HIRING MANAGER
"When your team is under pressure to ship, how do you actually balance that with Amazon's insistence on highest standards?"

PEER
"How does Amazon's written narrative culture actually change day-to-day decision-making compared to PowerPoint-driven environments?"

EXECUTIVE
"How does Amazon resolve the tension between Think Big and Frugality as the company has scaled?"

RECRUITER
"What differentiates SDE candidates who navigate Amazon's behavioral interviews from those who struggle with Leadership Principles?"

⚡ REMEMBER

- → LP stories first — connect every answer to a Leadership Principle
- → Data and metrics in every story — 'improved performance' is not enough
- → Bar Raiser will probe harder — do not soften answers when pressure increases
- → Ask clarifying questions before designing any system