indeed®

Site Reliability Engineer Hiring Process

What is Site Reliability at Indeed?

SRE's Mission at Indeed

We increase product reliability without compromising development velocity. Our priorities are guided by Service Level Objectives (SLOs). Indeed has had aggressive growth over the past few years, and SRE is seen as critical to helping Indeed continue to scale effectively.

Indeed's Site Reliability Engineering Team

SRE is one of the newest groups in Engineering, bringing together Software Engineers and Systems Engineers to work on one of today's most complex problems: scale. Your career with SRE will help pave the way for Indeed's success for the next 5 to 10 years.

We follow the Google model of Site Reliability: teams are composed of a mix of Software Engineers and Systems Engineers. Some are embedded directly with product teams while others work on Indeed's centralized infrastructure. Site Reliability Engineers at Indeed focus on activities like these, and more:

★ Working with business critical Indeed product teams to meet strict SLOs, reduce MTTR, and learn from from operating high traffic services in production in a constantly changing environment.
  ○ Evaluate and discuss reliability risk and practices to mitigate during architecture design review and new feature development
  ○ Plan and expand capacity

    - Participate in the on-call rotations
    - Reflect, evaluate risk, and propose improvements to process and code during blameless postmortems
    - Permanently address reliability problems by making code or process changes
- Creating new capabilities to improve production, especially around monitoring, alerting, fault tolerance, incident response, etc.
- Enabling developer self-service for work that used to require manual intervention from SREs
- Automating routine manual tasks

What are Site Reliability Engineers doing in each office?

Austin

- **Job Search team** - Indeed’s flagship job search product with 200M+ unique visitors per month. Strict latency and availability SLOs.
- **Employer Platform team** - Indeed’s product for employers to post jobs and manage candidates that apply to those jobs. Uptime and performance are business critical.
- **Datacenter failover project** - Increasing regional fault-tolerance at the datacenter level for key systems
- **Canary artifact project** - Progressive loading of large in-memory data sets using a distributed consensus protocol to establish that the new data set won’t take down many instances at once (Java, Consul)
- **Database failover project** - Automated failover of database connections to a replica or warm standby (Java, MySQL)

Dublin

- **Docker on Mesos project** - Testing and automating the installation of a registry, caching, and necessary components so that our Mesos tasks can run Docker containers
- **SRE access security project** - Modifying scripting libraries to use HashiCorp’s Vault for secure authentication to infrastructure components

San Francisco

- **Notifications Platform team** - a distributed system for generating and sending various types of notifications, including high-volume users like job alerts (both emails and push notifications), and lower-volume users with strict latency requirements (like account registration confirmation emails).

Seattle

- **Email Relay team** - a system that facilitates and mediates email-based communication between employers and candidates.
- **Database-as-a-Service team** - a system that enables developer self-service provisioning and management of databases (MySQL and MongoDB).

Tokyo

- **Apply Platform team** - Indeed’s product that makes it very easy for job seekers to apply to jobs
- **Logging modernization project** - Improve the performance of a centralized logging system (filebeat, kafka, logstash, elasticsearch) by introducing kafka, replacing logstash-forwarder with filebeat and using SSDs
- **memcached data availability project** - Improve memcached data availability by utilizing mcrouter (memcached, mcrouter)

- **MySQL load balancer project** - Load balanced and lag-aware read-only database connections (mysql, haproxy)

What we’re looking for during the onsite interviews

- Collaboration
- Communication
- Critical Thinking & Analysis
- Expressing Solutions in Code
- Troubleshooting
- Initiative
- Logical Reasoning
- Problem Solving
- Tenacity
- Thoroughness

Overview of the hiring process

Screen

In the pre-onsite screen, you’ll be answering SRE knowledge questions and writing code in a collaborative editing environment.

Onsite Interviews

Whiteboard Problem Solving

You’ll be solving one or more technical problems by writing code on a whiteboard. All of these problems will have you writing code on the whiteboard (or pen and paper if you prefer.)

Programming Exercise

This activity gives you a chance to demonstrate your problem solving ability while working at a computer. We’ll give you a HackerRank problem and loan you a laptop (if you haven’t brought your own.) You'll be expected to deliver working code to solve the problem in the language of your choice (supported languages). There will be an assistant from Indeed in the room to make sure that the question is clear and that the development environment is working for you.

Systems Architecture Interview

In this interview, we ask experienced candidates to spend time with one or two of our engineers to create a high-level design for a new system. You won’t be writing any code, but rather you’ll be focusing on the big picture for this system from design through production readiness.

Troubleshooting Discussion

This interview is likely to be quite different than other interviews encountered previously. It will simulate the experience of being on-call outside of business hours and evaluates your ability to think through a new problem in an unfamiliar architecture. You will be evaluated on your approach to troubleshooting, logical reasoning, risk mitigation, collaboration and problem solving.

Troubleshooting Exercise

This interview combines elements of the Troubleshooting Discussion and Programming Exercise (see above). You can bring your own laptop, or we can loan you one -- your choice. For the interview, you'll be at a keyboard troubleshooting real problems that we've caused in a distributed system running on Linux VMs that serves a web site. You'll be expected to investigate the problems and document your findings. There will be an assistant from Indeed in the room to make sure that the question is clear and that the environment is working for you.

Resumé Review

For experienced candidates this interview focuses on your background, the places you've worked, the teams you've led, and the impact you've had. We're interested in learning the story of how you arrived at where you are in your career and understanding the key experiences and opportunities that have shaped you in the process.

Closing

This interview is your time to ask any final questions you might have about Indeed and to provide feedback about your experiences during the day.

What you should expect during interviews

A brief introduction to Indeed and a tour of the office.

Each interview will last approximately 60 minutes (aside from the Programming Exercise where we typically schedule 90 minutes) and many will give you a chance to ask questions about Indeed or the role. These interviews will cover a number of technical, leadership and cultural assessments as described above.

Lunch with a member of the Indeed Engineering team. This is not a part of the interview process and is not included in the evaluation. It provides you an opportunity to learn more about Indeed by asking questions about our products, engineering culture, what it's like to work at Indeed, etc.

During the whiteboard interviews

We're interested in how you approach the problem. Think out loud and ask questions.

Our questions will typically start simple enough, but often leave room to explore interesting challenges. Real world challenges are often nuanced or subtle, and solutions to them often have second-order effects. Parts of our interview process are designed to explore these types of scenarios with you. You might be asked to solve a question multiple times with a variety of different constraints.

Most of the problems will have a variety of different answers ranging from a naive solution to some much too complex to fit on a whiteboard. An optimal answer would be great, but we're also interested in how you approach the problem – your thought process.

How to prepare for your onsite interview

Review Computer Science fundamentals including basic algorithms and data structures. Understand how a variety of data structures work and why you'd use one over another. Brush up on Big O notation and be able to perform time and space complexity analysis.

Practice writing code, especially if you're out of practice. Focus on your strongest language. Practice getting comfortable with writing real code on a whiteboard or a piece of paper. Avoid pseudo-code; we want to see syntactically correct code in whichever language you choose to code. Solve a few problems on HackerRank, so that you can become familiar with the environment we use for some of our interviews.

Take some time to review your skills around configuring, managing, and troubleshooting complex distributed systems. We use Linux at Indeed, but being able to talk about your current environment and your troubleshooting experience within that environment will come in handy during the interview panel. We have resources below for helping you prepare. An understanding of architecture principles as well as our troubleshooting mentality here at Indeed will make you a more successful candidate.

Review your resume and recall specific stories you may want to share. Determine the projects you have been a part of that best exemplify your skills, accomplishments, abilities, and approach towards business.

Resources to help you prepare for the interviews

Websites

Project Euler Problems
The Stony Brook Algorithm Repository
Indeed Engineering Blog
LeetCode.com
System Design Primer
Topcoder
HackerRank
Interview Cake
Big-O Cheat Sheet

Books

The Algorithm Design Manual
Cracking the Coding Interview
Site Reliability Engineering: How Google Runs Production Systems
Algorithms and Programming: Problems and Solutions

Linux Fundamentals

Linux memory management and virtual memory
Linux page cache
Linux system calls

What is the difference between a process and a thread?
Process states: zombies and uninterruptible sleep
What happens when you type google.com into your browser's address box and press enter?
Debugging techniques for sysadmins
Understanding the Linux OOMKiller
Unix signals

Linux debugging and utilities:

How to interpret the numbers in top
How to use strace
How to use tcpdump
Linux debugging tools
Solving problems with /proc filesystem