

SRE Skills Self-Assessment Checklist

A Comprehensive Self-Assessment Tool for Building and Enhancing
Your SRE Skillset

Introduction

Site Reliability Engineering (SRE) is an interdisciplinary role that combines software engineering, systems operations, and a strong focus on maintaining the reliability, availability, and performance of large-scale systems.

As SREs take on more complex and mission-critical responsibilities, it's essential to have a well-rounded skill set. This SRE Skills Self-Assessment Checklist is designed to help you evaluate your current proficiency across key areas of the SRE discipline, identify gaps in your knowledge, and target areas for improvement.

Whether you're an aspiring SRE, a seasoned professional looking to sharpen your expertise, or someone preparing for an SRE interview, this guide offers a structured approach to assessing your core competencies.

To use this guide effectively, simply go through each section and rate your skills honestly.

For each area, ask yourself the questions provided to assess your comfort level and familiarity. If you feel confident, that's great! If not, consider that an opportunity for growth.

This guide is structured to help you identify where your strengths lie and which skills require further development.

Focus on building a tailored learning plan based on the results of this self-assessment, and revisit this checklist periodically to track your progress over time.

1. Systems Engineering Fundamentals

Skills to Assess:

- Understanding of distributed systems, fault tolerance, and high availability
- Proficiency in designing, building, and scaling resilient systems
- Familiarity with hardware and software architectures

Questions for Self-Assessment:

- Can I describe how to design a system to handle failures gracefully?
- Do I understand the principles of high availability and disaster recovery?
- Am I comfortable working with different system architectures and technologies (e.g., cloud, on-premises, hybrid)?

Improvement Tips:

- If you're unsure about fault tolerance or system resiliency, consider studying distributed system designs such as CAP theorem and techniques for building fault-tolerant systems.
- Participate in practical exercises like building high-availability systems or contributing to projects that require scaling and redundancy strategies.

2. Monitoring and Observability

Skills to Assess:

- Ability to implement and manage monitoring solutions for systems and applications
- Knowledge of observability principles, including metrics, logs, and traces
- Experience with popular monitoring tools such as Prometheus, Grafana, or Datadog

Questions for Self-Assessment:

- Can I configure and use monitoring tools to track system health and performance?
- Am I proficient in setting up alerts for service disruptions or performance anomalies?
- Do I understand how to use logs, metrics, and traces to troubleshoot issues in production environments?

Improvement Tips:

- Start by implementing a simple monitoring solution for your system, ensuring it captures essential metrics like CPU usage, memory consumption, request latency, and error rates.
- Familiarize yourself with observability frameworks (e.g., OpenTelemetry) and dive deeper into tools like Prometheus and Grafana for building dashboards and setting up automated alerts.

3. Incident Management and Response

Skills to Assess:

- Understanding incident response lifecycle (detection, mitigation, resolution, and post-mortem)
- Familiarity with tools and techniques for managing production incidents
- Ability to collaborate with cross-functional teams during incidents

Questions for Self-Assessment:

- Do I know how to respond quickly to production incidents and reduce downtime?
- Am I able to identify the root cause of incidents and take corrective actions?
- Can I lead or contribute to a post-mortem analysis to prevent similar incidents in the future?

Improvement Tips:

- Practice simulating incidents (e.g., chaos engineering) to understand how to react under pressure.
- If you haven't yet, take part in post-mortem meetings to learn from incidents and incorporate improvements into your systems and workflows.

4. Automation and Scripting

Skills to Assess:

- Proficiency in automating repetitive tasks such as deployments, monitoring, and scaling
- Experience with scripting languages like Python, Bash, or Go
- Familiarity with configuration management tools (e.g., Ansible, Puppet, Chef) and Infrastructure as Code (IaC) tools like Terraform or CloudFormation

Questions for Self-Assessment:

- Am I able to automate routine operational tasks to reduce manual intervention?
- Do I have experience with writing scripts to improve the efficiency of deployment pipelines or monitoring?
- Can I use IaC tools to automate infrastructure provisioning and management?

Improvement Tips:

- Start by automating simple tasks like server provisioning or application deployments.
- Work on writing scripts to automate logging, monitoring, or performance testing.
- Experiment with IaC tools (e.g., Terraform or AWS CloudFormation) to automate infrastructure provisioning.

5. Cloud and Containerization

Skills to Assess:

- Understanding of cloud platforms such as AWS, Google Cloud, or Azure
- Familiarity with containerization and container orchestration tools like Docker and Kubernetes
- Knowledge of managing services in a cloud environment, including load balancing and autoscaling

Questions for Self-Assessment:

- Do I understand the basic concepts and services of major cloud providers (e.g., EC2, S3, Kubernetes)?
- Am I comfortable with containerization concepts and running applications in containers using Docker or Kubernetes?
- Can I design a highly available and scalable service architecture using cloud-based tools?

Improvement Tips:

- If you're new to cloud platforms, start by exploring the basic services (compute, storage, networking) on platforms like AWS or Google Cloud.
- Learn Docker to containerize applications and dive into Kubernetes for orchestrating containerized applications in production.
- Try building and managing a small-scale Kubernetes cluster to understand container deployment and management better.

6. Service Level Objectives (SLOs), Service Level Indicators (SLIs), and Service Level Agreements (SLAs)

Skills to Assess:

- Ability to define, measure, and manage SLIs, SLOs, and SLAs
- Understanding of the role these metrics play in measuring and ensuring reliability
- Knowledge of error budgets and how they influence decision-making

Questions for Self-Assessment:

- Can I define SLIs and SLOs for critical systems in my organization?
- Do I understand how to calculate and interpret error budgets for reliable service delivery?
- Am I familiar with the differences between SLOs, SLIs, and SLAs, and how they are applied in day-to-day operations?

Improvement Tips:

- Review how SLOs and SLIs are defined in various applications and ensure you're measuring the right metrics that reflect system reliability.
- Use tools like Prometheus and Datadog to track SLIs and monitor how close the system is to breaching SLOs, and learn how to take corrective actions if you reach your error budget.

7. Security Practices

Skills to Assess:

- Knowledge of securing production systems and protecting sensitive data
- Familiarity with encryption, access controls, and authentication protocols
- Understanding of common security threats and how to mitigate them (e.g., DDoS, data breaches)

Questions for Self-Assessment:

- Am I aware of the latest security best practices for securing cloud infrastructure and applications?
- Do I know how to implement role-based access control (RBAC) and encryption for sensitive data?
- Can I identify and mitigate common security threats in my infrastructure?

Improvement Tips:

- Review security best practices in cloud environments and for web applications.
- Practice securing systems by setting up encryption, multi-factor authentication (MFA), and security monitoring systems.
- Consider taking a security-focused certification, like AWS Certified Security Specialty or Google Cloud Security Engineer.

8. Collaboration and Communication

Skills to Assess:

- Ability to collaborate with development, operations, and product teams
- Proficiency in writing clear documentation, including runbooks and incident reports
- Comfort with conducting technical discussions and decision-making across teams

Questions for Self-Assessment:

- Do I communicate clearly and effectively with other teams, especially during incidents or system upgrades?
- Can I write runbooks and incident reports that are helpful for resolving issues efficiently?
- Am I able to explain complex technical concepts to non-technical stakeholders?

Improvement Tips:

- Focus on improving your communication skills, especially during high-pressure situations like incidents.
- Practice writing incident reports and technical documentation that is clear, concise, and actionable.
- Work on collaborating more closely with developers and other teams to improve cross-team understanding of reliability goals.

Moving Forward in Your SRE Journey

By working through this SRE Skills Self-Assessment Checklist, you've taken a crucial step toward becoming a more effective and well-rounded Site Reliability Engineer.

The SRE field is fast-paced and constantly evolving, so continuous self-assessment and improvement are vital to keeping up with new tools, practices, and technologies.

This checklist is not a one-time activity but a tool to guide your ongoing development and ensure you're prepared for the challenges that come with building and maintaining reliable systems.

Use this guide as a reference to set specific, actionable goals for your professional growth.

Once you identify areas that need improvement, explore additional resources such as online courses, hands-on projects, or mentorship to deepen your expertise.

As you enhance your skills, you'll be better positioned to excel in your current role, contribute to your team's success, and take on new and more complex challenges within the realm of Site Reliability Engineering.

Remember, the road to SRE success is a continuous learning journey, and with regular self-assessments, you can ensure you're always moving in the right direction.

SITE RELIABILITY ENGINEERING (SRE) FOUNDATION CERTIFICATION (CSREF)

SRE Certification is based on SRE Principles and Scalable IT Operations.



ABOUT GSDC CERTIFICATION



LIFETIME VALIDITY

GSDC Certification is an globally accredited certification with lifetime validity.



EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.



CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.



LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Apply SRE concepts to improve reliability in modern IT environments.
- Learn to define, monitor, and manage service level objectives (SLOs).
- Develop skills in incident response, automation, and alerting.
- Promote a culture of continuous learning and operational improvement.

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now



www.gsdccouncil.org