



# SRE

## Why SRE?

As systems grow:

- Downtime becomes expensive
- Manual ops do not scale
- Fast DevOps releases increase production risk

SRE exists to **control risk while moving fast**.

It addresses:

- Production outages
- Unpredictable system behavior
- Alert fatigue
- Slow incident recovery
- Reliability regressions due to rapid releases

---

## What is SRE?

SRE is:

- A **role, practice, and engineering approach**
- Operations written as **code**
- Reliability treated as a **measurable feature**

Not:

- A monitoring-only team
- A replacement for DevOps

SRE **implements DevOps goals** with strict reliability controls.

---

## How SRE Works

SRE operates as a **feedback-controlled loop**:

1. Design for reliability
  2. Define reliability targets (SLOs)
  3. Automate operations
  4. Monitor production continuously
  5. Respond to incidents
  6. Learn via postmortems
  7. Improve systems, not people
- 

## Core SRE Concepts

- **SLI (Service Level Indicator)**  
Actual measured metrics (latency, error rate, uptime)
- **SLO (Service Level Objective)**  
Target reliability (e.g. 99.9% availability)
- **SLA (Service Level Agreement)**  
Business contract based on SLOs

- **Error Budget**

Allowed failure window

If exhausted → feature releases stop, reliability work starts

- **Toil**

Manual, repetitive operational work

Must be automated or eliminated

---

## Benefits of SRE

### Reliability Benefits

- Predictable uptime
- Controlled risk during releases
- Fewer production incidents

### Engineering Benefits

- Automation-first operations
- Scalable systems
- Reduced manual intervention

### Operational Benefits

- Faster incident detection
  - Faster recovery (lower MTTR)
  - Clear ownership during outages
- 

## Business-Level Benefits

- Reduced downtime cost
- Higher customer trust
- Reliability-backed SLAs

- Safer rapid innovation
- 

## Technical-Level Benefits

- Advanced monitoring & alerting
  - Automated incident response
  - Capacity planning based on data
  - Chaos engineering and failure testing
- 

## SRE vs DevOps (Clear Separation)

DevOps	SRE
Cultural & workflow model	Engineering implementation
Focus on speed	Focus on reliability
CI/CD driven	SLO & error-budget driven
Shared responsibility	Dedicated reliability ownership

DevOps enables speed.

SRE **controls the blast radius.**

---

## SRE Principles

- **Reliability as a feature**
- **Error budgets over blind velocity**
- **Automate everything**
- **Eliminate toil**
- **Blameless postmortems**
- **Observability by default**
- **Production is the priority system**