SRE Fundamentals

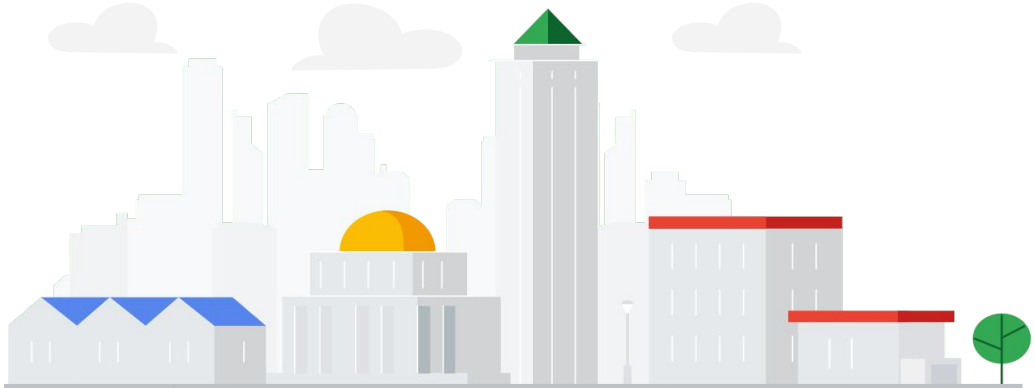
TAM Webinar

SEP 2022

Google Cloud

Let's go

1. Purpose & Target
2. Agenda
3. Learning & Certification
4. Q&A



Purpose
& Target

Google Cloud

SRE Fundamentals

O objetivo deste TAM webinar é compartilhar com a Google Cloud Community sobre o tema de Site Reliability Engineering (SRE).
Você conhecerá os princípios e as práticas que possibilitam deixar os sistemas mais escaláveis, confiáveis e eficientes – lições que podem ser diretamente aplicáveis à sua empresa.

Agenda

Agenda do SRE Fundamentals

Sep

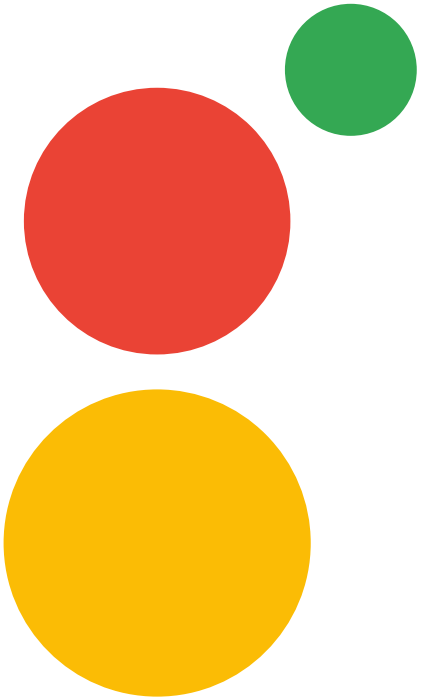
XX

14:00 (BRT)

14:00 ~ 14:05	{ Abertura }
14:05 ~ 14:50	{ Introdução ao SRE }
14:50 ~ 15:00	{ Q&A}

Pamella Canova

Technical Account Manager



Introdução ao SRE

Pamella Canova
Technical Account Manager

Google Cloud

Site Reliability Engineering

Tópicos

O que é SRE?

Princípios de SRE

Práticas de SRE

Como obter ajuda

Google Cloud

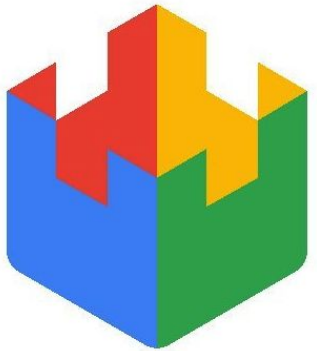
O que é SRE?

1

Definição & História

"SRE é "o que acontece quando um engenheiro de software é encarregado do que costumava ser chamado de operações." Benjamin Treynor, VP de SRE.

Site Reliability Engineering (SRE), ou "Engenharia de Confiabilidade de Sites", é uma disciplina que incorpora aspectos da engenharia de software e os aplica à resolução de problemas de operações de TI. Ou seja, são profissionais de engenharia de software que se responsabilizam, de forma multidisciplinar, na gestão e automação do ambiente de tecnologia.



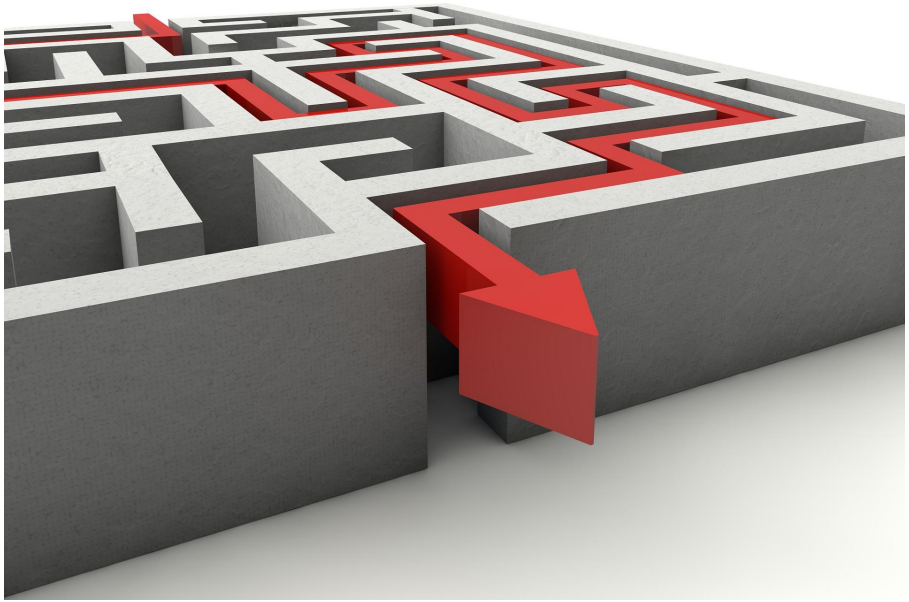
O conceito de SRE foi criado no Google em 2003, quando Ben Treynor foi contratado para liderar uma equipe de engenheiros de software responsáveis por um ambiente de produção. A equipe foi encarregada de fazer com que os sites do Google funcionassem de maneira suave, eficiente e confiável.

No início, os sistemas de larga escala do Google exigiam que a empresa apresentasse novos paradigmas sobre como gerenciar sistemas tão grandes e, ao mesmo tempo, introduzir novos recursos continuamente, mas com uma experiência de usuário final de alta qualidade.

Custo do software a longo prazo

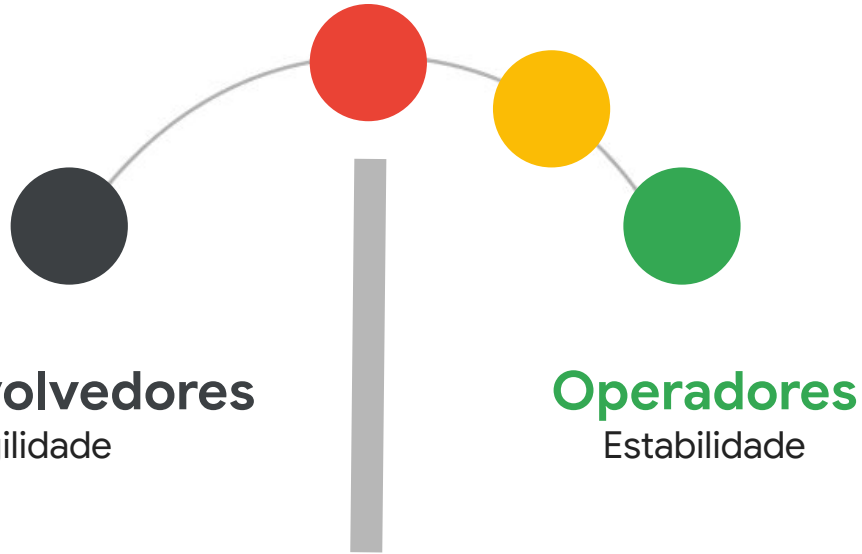
A engenharia de software como disciplina tem como foco projetar e construir, em vez de operar e manter, apesar das estimativas de que 40%[1] a 90%[2] do custo total é incorrido depois do lançamento.

1 Glass, R. (2002). Facts and Fallacies of Software Engineering, Addison-Wesley Professional; p. 115.
2 Dehaghani, S. M. H., & Hajrahimi, N. (2013). Which Factors Affect Software Projects Maintenance Cost More? Acta Informatica Medica, 21(1), 63–66. http://doi.org/10.5455/AIM.2012.21.63-66



Google Cloud

Os incentivos não estão alinhados.



Redução do atrito do ciclo de vida do produto

Google Cloud

Interface DevOps

- **DevOps**

  Conjunto de práticas, diretrizes e cultura projetado para quebrar silos em desenvolvimento, operações, arquitetura, rede e segurança de TI.

- **5 áreas principais**

  1. Reduzir silos organizacionais
  2. Aceitar falhas como normais
  3. Implementar mudanças graduais
  4. Usar ferramentas e automatização
  5. Medir tudo

Abordagem SRE para operações

Use dados para guiar tomada de decisão.

Trate operações como um problema de engenharia de software:

- Contrate pessoas motivadas e capazes de escrever código.
- Use software para realizar tarefas normalmente feitas pelos sysadmins.
- Projete arquiteturas de serviços mais confiáveis.

Google Cloud

O que a equipe de SRE faz?

- Os SREs desenvolvem soluções para desenhar, criar e executar sistemas de larga escala de forma **escalável, confiável e eficiente.**
- Os SREs **guiam a arquitetura do sistema** combinando desenvolvimento de software e engenharia de sistemas.

- SRE é um trabalho, uma mentalidade e um conjunto de **abordagens de engenharia** para executar melhores sistemas de produção.
- Realizamos nosso trabalho com um espírito pessimista construtivo: **esperamos o melhor, mas nos preparamos para o pior**.

SRE implementa DevOps

- **DevOps**

  Conjunto de práticas, diretrizes e cultura projetado para quebrar silos em desenvolvimento, operações, arquitetura, rede e segurança de TI.

- **Site Reliability Engineering**

  Conjunto de práticas que criamos, algumas filosofias que sustentam essas práticas e um cargo.

- **5 áreas principais**

  1. Reduzir silos organizacionais
  2. Aceitar falhas como normais
  3. Implementar mudanças graduais
  4. Usar ferramentas e automação
  5. Medir tudo

Orçamento de Erros

O princípio chave de SRE

2

Como medir a confiabilidade

- Abordagem ingênua:

$$\text{Disponibilidade} = \frac{\text{Tempo bom}}{\text{Tempo total}}$$

= fração de tempo na qual o serviço está disponível e funcionando

- Intuitivo para o ser humano

- Relativamente fácil de medir se for uma métrica binária contínua, por exemplo, tempo de atividade de um servidor.

- Bem mais difícil para serviços distribuídos
  - Um servidor que atualmente não recebe solicitações, está online ou offline?
  - Se um de três servidores estiver inoperante, o serviço estará online ou offline?

Como medir a confiabilidade

- Abordagem mais sofisticada:

$$\text{Disponibilidade} = \frac{\text{Interações boas}}{\text{Total de interações}}$$

= fração de usuários reais para os quais o serviço está disponível e funcionando

- Lida bem com serviços de requisições e respostas distribuídas
- Permite estes casos:
  - Um servidor que atualmente não recebe solicitações, está online ou offline?
  - Se um de três servidores estiver inoperante, o serviço estará online ou offline?

Nível de confiabilidade	Intervalo de não confiabilidade permitido		
	por ano	por trimestre	por 30 dias
90%	36,5 dias	9 dias	3 dias
95%	18,25 dias	4,5 dias	1,5 dias
99%	3,65 dias	21,6 horas	7,2 horas
99,5%	1,83 dias	10,8 horas	3,6 horas
99,9%	8,76 horas	2,16 horas	43,2 minutes
99,95%	4,38 horas	1,08 horas	21,6 minutos
99,99%	52,6 minutos	12,96 minutos	4,32 minutos
99,999%	5,26 minutos	1,30 minutos	25,9 segundos

Nível de confiabilidade	Intervalo de não confiabilidade permitido		
	por ano	por trimestre	por 30 dias
90%	36,5 dias	9 dias	3 dias
95%	18,25 dias	4,5 dias	1,5 dias
99%	3,65 dias	21,6 horas	7,2 horas
99,5%	1,83 dias	10,8 horas	3,6 horas
99,9%	8,76 horas	2,16 horas	43,2 minutes
99,95%	4,38 horas	1,08 horas	21,6 minutos
99,99%	52,6 minutos	12,96 minutos	4,32 minutos
99,999%	5,26 minutos	1,30 minutos	25,9 segundos

Error Rate	Allowed duration
100%	21.6 minutes
10%	3.6 hours
1%	36 hours
0.1%	15 days
<0.05%	all month

Source: https://landing.google.com/sre/sre-book/chapters/availability-table/

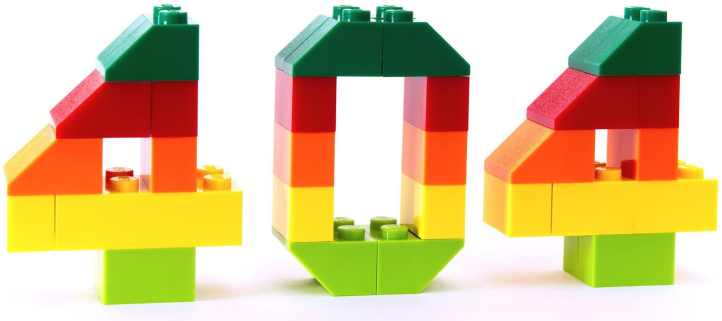
“100% é a meta de confiabilidade errada para praticamente tudo.”

Benjamin Treynor Sloss, Vice-Presidente de Engenharia Contínua, Google

Google Cloud

Orçamentos de erro

- Os Gerentes de Produtos e SREs definem a **meta de disponibilidade**.
- 100% - meta de disponibilidade = “orçamento de não confiabilidade” (ou **orçamento de erro**).
- O monitoramento mede o tempo de ***uptime* real**.
- Ciclos de controle para usar o orçamento!



Public Domain Image

Google Cloud

Vantagens do orçamento de erro

- **Incentivo comum para desenvolvedores e SREs**

  Encontra um bom equilíbrio entre inovação e confiabilidade
- **A equipe de desenvolvimento pode gerenciar o risco**

  Decidem como gastar o orçamento de erro
- **Metas de confiabilidade irreais são pouco atraentes**

  As metas prejudicam a rapidez da inovação

- **A equipe de desenvolvimento se policia**

  O orçamento de erro é um recurso importante para ela
- **Responsabilidade pelo *uptime* do sistema compartilhada**

  Falhas na infraestrutura afetam o orçamento de erro dos desenvolvedores

Glossário dos termos

SLI

Service Level Indicator

indicador de nível de serviço: uma medida que define claramente o que é “satisfatório”

- *usado para especificar SLO/SLA*
- *Func(métrica) < limite*

SLO

Service Level Objective

objetivo de nível de serviço: uma boa meta para a porcentagem de interações bem-sucedidas

- *especifica metas (SLI + meta)*

SLA

Service Level Agreement

acordo de nível de serviço: consequências

- *SLA = (SLO + limite) + consequências = SLI + meta + consequências*

Definição e avaliação do SLO

- Objetivo de nível de serviço (SLO): uma meta para SLIs **agregados ao longo do tempo**
  - Medido usando um **SLI (indicador de nível de serviço)**
  - Em geral: **sum(SLI cumprido) / intervalo >= porcentagem-alvo**
- Tente exceder a meta do SLO, *mas não muito*

- Não é fácil escolher o SLO apropriado.
  Tente manter as coisas simples, evite absolutos, a perfeição pode esperar.

  Por quê?
  - Define prioridades e restrições para SRE e o trabalho dos desenvolvedores
  - Define as expectativas do usuário sobre o nível de serviço

Ciclo de vida do produto

SRE implementa DevOps

- **DevOps**

  Conjunto de práticas, diretrizes e cultura projetado para quebrar silos em desenvolvimento, operações, arquitetura, rede e segurança de TI.

- **Site Reliability Engineering**

  Conjunto de práticas que criamos, algumas crenças que sustentam essas práticas e um cargo.

- **5 áreas principais**

  1. Reduzir silos organizacionais: **compartilhar a responsabilidade**
  2. Aceitar falhas como normais: **orçamento de erro**
  3. Implementar mudanças graduais
  4. Usar ferramentas e automação
  5. Medir tudo: **medir a confiabilidade**

As práticas de SRE

3

Áreas de prática

Métrica e monitoramento

Planejamento de capacidade

Gerenciamento de mudanças

Resposta à emergências

Cultura

Monitoramento e alertas

- **Monitoramento: automatize a captura de métricas**

  Principais meios para determinar e manter a confiabilidade

- **Alerta: envie uma notificação quando algumas condições são detectadas**

  Pager: resposta humana imediata é necessária

  Ticket: intervenção humana é necessária, mas não imediata

- **Só solicite a intervenção humana se o SLO estiver ameaçado**

  As pessoas não devem ficar visualizando painéis, lendo logs, etc. apenas para determinar se o sistema está funcionando

Previsão de demanda e planejamento de capacidade

Planejar o crescimento orgânico
Maior adoção e uso de produtos pelos clientes.

Determinar o crescimento inorgânico
Aumentos repentinos na demanda por causa de lançamentos de recursos, campanhas de marketing etc.

Equiparar os recursos brutos com a capacidade de serviço
A capacidade disponível deve ser suficiente para atender às metas de confiabilidade.

Eficiência e desempenho

A capacidade pode ser cara ⟶ otimize a utilização

- O uso de recursos é uma função da demanda (carga), capacidade e da eficiência do software
- O SRE exige previsão e provisionamento e pode modificar o software

O SRE monitora a utilização e o desempenho

- Pode detectar retrocessos e tomar uma medida
- Equipe imatura: ajusta recursos ou melhora a eficiência de software
- Equipe madura: *rollback*

Source: Pixabay (no attribution required)

Google Cloud

Gestão de mudanças

- Cerca de 70%[1] das interrupções acontecem por causa das mudanças no sistema em uso

[1] Análise de dados internos do Google, 2011 a 2018

Medidas para minimizar:

- Realizar implementações progressivas
- Detectar problemas com rapidez e precisão
- Reverter alterações com segurança quando surgirem problemas

- Remover a intervenção humana do processo com automação para:
  - Reduzir erros
  - Reduzir a fadiga
  - Acelerar o ritmo

Em busca da velocidade máxima na mudança

100% é a meta de confiabilidade errada para praticamente tudo

- Determine a confiabilidade desejada para o seu produto
- Não procure fornecer uma qualidade melhor que a desejada

Use o orçamento de erro para acelerar o processo de desenvolvimento

- O objetivo não é ter nenhuma falha, mas sim a velocidade máxima dentro do orçamento de erro
- Use o orçamento de erro para lançamentos, testes etc..

Google Cloud

Provisionamento

Uma combinação de gerenciamento de mudanças e planejamento de capacidade

- Aumentar o tamanho de uma instância de serviço existente
- Executar mais instâncias/locais

Deve ser rápido

- A capacidade não utilizada pode sair caro

Deve ser adequado

- A capacidade adicionada precisa ser testada
- Muitas vezes, uma mudança significativa na configuração⟶arriscado

Google Cloud

Resposta de emergência

“As coisas quebram, faz parte da vida”

Poucas pessoas reagem naturalmente às emergências, por isso você precisa de um plano:

- **Em primeiro lugar, não entre em pânico!** Você não está sozinho e o mundo não está acabando.
- Mantenha a calma, detecte e corrija o problema.
- Se você se sentir sobrecarregado, peça ajuda a outras pessoas.

Google Cloud

Início do incidente e da Retrospectiva

- Tempo de inatividade visível para o usuário ou degradação além de um certo limite
- Perda de dados de qualquer tipo
- Tempo de resolução acima do limite

É importante definir critérios de início e retrospectiva de incidente antes que ele ocorra.

Used with permission of the image owner Jennifer Petoff, Sidewalk Safari Blog

Google Cloud

Filosofia da Retrospectiva

O objetivo de registrar uma Retrospectiva é garantir que:

- O incidente seja documentado
- Todas as causas envolvidas sejam compreendidas
- Sejam adotadas medidas preventivas adequadas para reduzir as chances e/ou o impacto de uma nova ocorrência

- **A retrospectiva deve acontecer após qualquer evento indesejável significativo**
- Fazer a retrospectiva não é castigo

Ninguém tem culpa

- A retrospectiva deve identificar as causas envolvidas sem mencionar qualquer pessoa ou equipe
- A retrospectiva neutra parte do princípio de que todos os envolvidos em um incidente tinham boas intenções
- O erro "humano" é **problema do sistema**. Não dá para "consertar" pessoas, mas sim sistemas e processos que ajudem a fazer as escolhas certas.
- Se o costume de jogar a culpa prevalecer, as pessoas não vão apresentar os problemas com medo da punição

Gerenciamento de trabalho operacional

Por quê?

- **Porque:**
  - A exposição a falhas reais ajuda a projetar os sistemas
  - Não dá para automatizar tudo
  - Se você realizar parte do trabalho operacional, saberá o que automatizar

O quê?

- **Trabalho diretamente relacionado à execução de um serviço que seja:**
  - Manual (executar um script manualmente)
  - Repetitivo (feito todos os dias ou para cada novo cliente)
  - Automatizável (não requer julgamento humano)
  - Tático (baseado na interrupção ou reativo)
  - Sem valor temporal (sem melhorias no sistema a longo prazo)
  - O(n) para crescimento de serviço (cresce com o número de usuários ou abrangência do serviço)

Habilidades da equipe

Contrate bons **engenheiros de software (SWE)** *e* bons **engenheiros de sistema (SE)**.

Não necessariamente tudo na mesma pessoa.

Tente incluir 50% de cada (SWE e SE) na equipe

Todos devem saber programar.

SE != "trabalho operacional"

Para obter mais detalhes, consulte “Hiring Site Reliability Engineers”, de Chris Jones, Todd Underwood e Shylaja Nukala,; login :, junho de 2015

Empoderamento dos SREs

- Os SREs devem ter o poder aplicar o orçamento de erro e o orçamento de trabalho exaustivo.
- Os SREs são recursos valiosos. Use seu tempo com sabedoria.
- Procure não impor muita carga operacional sobre os SREs; divida a carga para ter uma equipe saudável.

Source: Pixabay (no attribution required)

SRE implementa DevOps

- **DevOps**

  Conjunto de práticas, diretrizes e cultura projetado para quebrar silos em desenvolvimento, operações, arquitetura, rede e segurança de TI.

- **Site Reliability Engineering**

  Conjunto de práticas que criamos, algumas crenças que sustentam essas práticas e um cargo.

- **5 áreas principais**

  1. Reduzir silos organizacionais
     compartilhar a responsabilidade
  2. Aceitar falhas como normais
     orçamento de erro ***e postmortem*** **neutro**
  3. Implementar mudanças graduais
     reduzir custo de falhas
  4. Usar ferramentas e automação
     automatizar casos comuns
  5. Medir tudo
     medir a confiabilidade **e trabalho exaustivo**

Como começar

4

Google

Faça essas quatro coisas

1. Comece com Service Level Objectives.
   Time de SRE trabalham para um SLO e/ou orçamento de erro. Eles defendem o SLO.
2. Contrate pessoas de operações para escrever código
   Eles irão ficar entediados do trabalho manual e o substituirão por código e automação.
3. Garanta a paridade de respeito com o restante da organização de desenvolvimento/engenharia
4. Providencie um loop de feedback para auto-regulação
   Os times de SRE escolhem seu trabalho. Os SREs devem poder recusar novos trabalhos ou diminuir SLOs quando estão sobrecarregados.

Google

Você é capaz.

- Escolha **um** serviço para rodar no modelo SRE.
- Empodere o time com apoio e patrocínio executivo.
- A segurança cultural e psicológica é crítica.
- Meça os Service Level Objectives a saúde de seu time.
- Progresso incremental libera tempo para mais progresso.

Google

Espalhe o amor.

- Compartilhe as técnicas e conhecimento com sua companhia assim que tiver um caso de estudo sólido.
- Se você tiver SLOs bem definidos, o Google pode trabalhar com você na redução de fricção com monitoramento compartilhado e outras colaborações.

Google

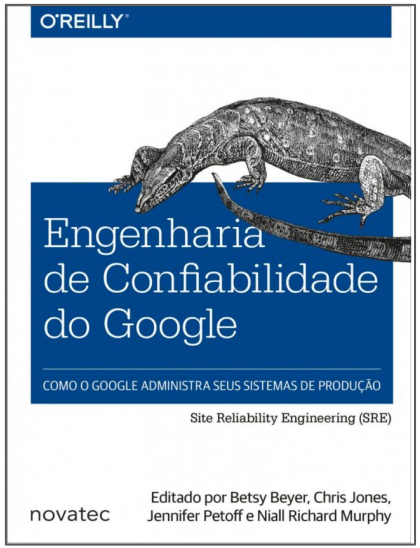
SRE resolve a confiabilidade da nuvem

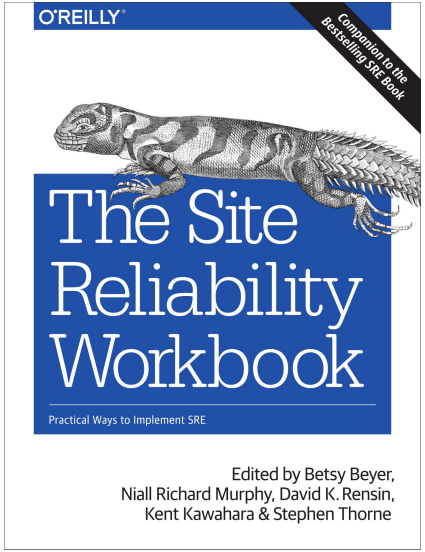
- Escalabilidade sem esforço não significa uma escalada de demandas operacionais.
- Automação e engenharia de operações permitem escalar sistemas sem escalar as organizações.
- A tensão entre o desenvolvimento e operações não precisa existir.
- Orçamentos de erro providenciam o mensuramento e flexibilidade para entregar confiabilidade e velocidade de produtos.

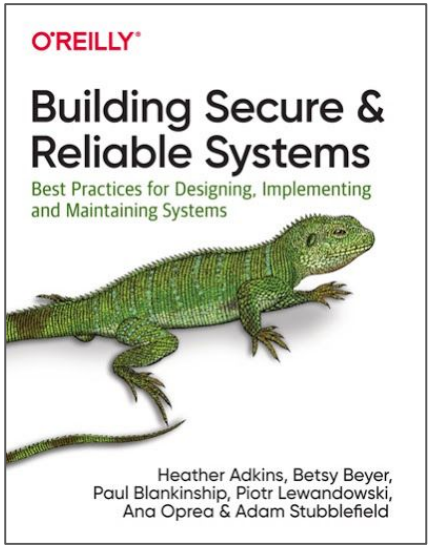
Google

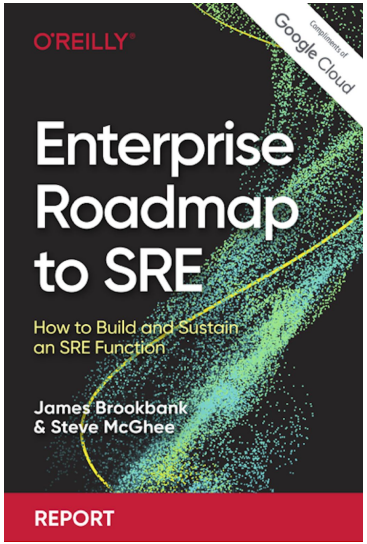
Learning & Certification

Encontre as publicações de SRE do Google SRE — incluindo os livros de SRE, artigos, treinamentos e mais - gratuitos em sre.google/resources.









Site Reliability Engineering: Measuring and Managing Reliability

https://www.coursera.org/learn/site-reliability-engineering-slos

Google Cloud

Certificações do Google Cloud

Professional Cloud DevOps Engineer

Professional Collaboration Engineer

Professional Cloud Architect

Professional Data Engineer

Professional Machine Learning Engineer

Cloud Digital Leader

Associate Cloud Engineer

Professional Cloud Developer

Professional Cloud Network Engineer

Professional Cloud Security Engineer

Foundational
Cloud knowledge and working in the cloud

Associate
Recommended 6+ months hands-on experience with GCP

Professional
Recommended 3+ years industry experience & 1 year hands-on experience with GCP

A educação dos Developers é importante para o Google Cloud

"Há poucos consensos tão difundidos quanto o do poder da educação. A única maneira de uma indústria crescer é por meio de uma estrutura sólida de programas de aprendizagem e treinamento que garantam talentos que atendam à demanda por novos empregos à medida que os negócios crescem."

Eduardo López

Presidente, Google Cloud Latin America

Perguntas?

Google

Obrigada!