



SRE是新运维"奇点"

一浅谈谷歌SRE与开源技术的新发展

刘峰中国SRE社区布道师/金牌讲师





- I. SRE定义、原则和方法论
- II. SRE自动化工具和开源技术
- III.SRE的新发展
- IV.SRE是新运维的"奇点"

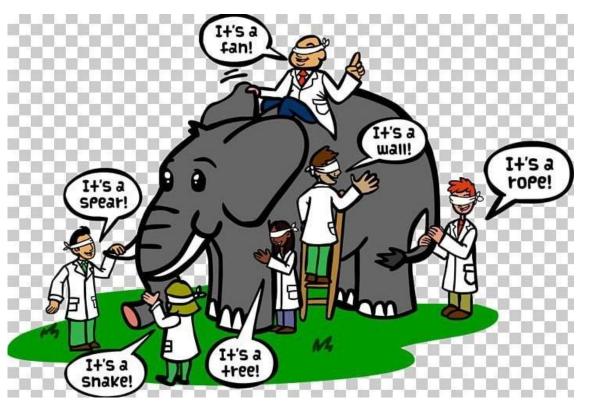


I.SRE定义、原则和方法论

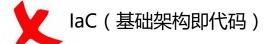
- · SRE的起源与定义
- · SRE的六大原则
- · SRE的方法论

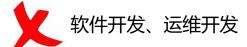
什么是SRE?



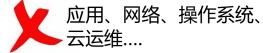












Google:什么是SRE?



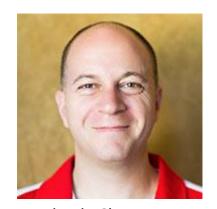
SRE

Site Reliability Engineering

- · SRE是一个学科
- · SRE是一种最佳实践
- · SRE是一类创新岗位

SRE:起源





Benjamin Sloss Treynor Google 的副总裁,主要工作就是确保Google 的网站永不掉线

- SRE:站点可靠性工程(Google负责7*24小时运维的VP命名)
 - 1. SRE是学科:使用计算机和软件工程手段设计和研发大型、分布式计算机软件系统;
 - 2. SRE关注:焦点是**可靠性**,包括架构设计、运维流程优化—> "足够可靠";
 - 3. SRE主要工作:运维分布式集群系统上的具体业务服务(Service)
- **SRE是一种职业**:专注于**软件系统生命周期**管理的IT工程师

SRE:起源



- 世界上第一个<u>SRE</u> 专注于参与**阿波罗登月计划**的MIT教授 Margaret Hamilton
- Apollos计划中的SRE

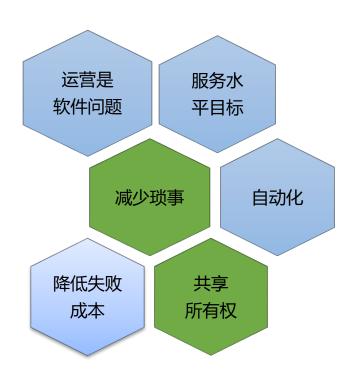
Apollo 7 飞船研发事故的启发:

- 1. 类似的情况不断发生
- 2. 对于细节的不懈关注
- 3. 做好充足的灾难预案和准备工作
- 4. 时刻警惕
- 5. 不放弃一切机会去避免灾难发生以上是SRE最重要的理念!



Google SRE 原则 Principle





- 核心原则
- 关键成功因素(KSF)

Google SRE 实践总结



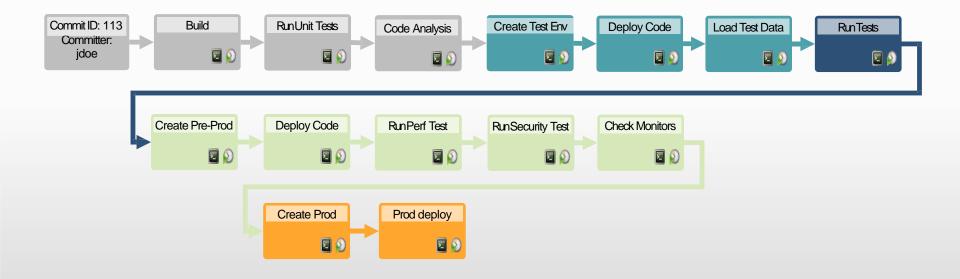
- SRE方法论:
 - 1. 确保长期关注研发工作
 - 2. 在保障SLO的前提下最大化迭代速度
 - 3. 监控系统
 - 4. 应急事件处理
 - 5. 变更管理
 - 6. 需求预测和容量规划
 - 7. 资源部署
 - 8. 效率与性能



II.SRE自动化工具与开源技术

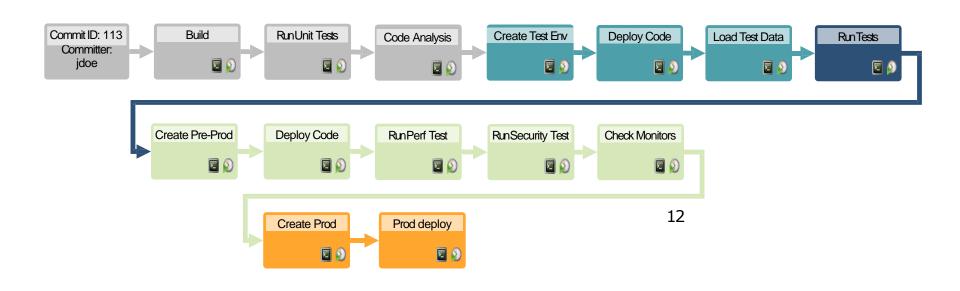
- · DevOps自动化的挑战
- · SRE领导的自动化
- · SRE支持全栈开源技术

自动化的焦点: DEVOPS交付流水线

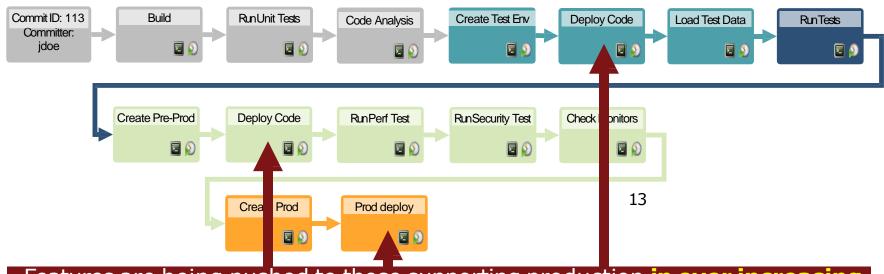




A lot of automation effort is "Dev" led (left-to-right), DevOps with a "big Dev and small Ops".

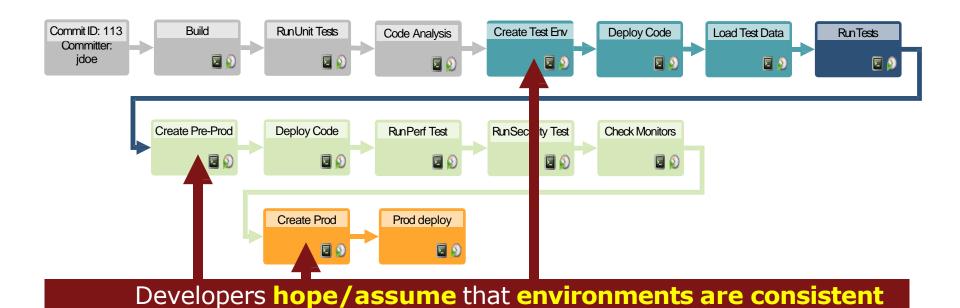




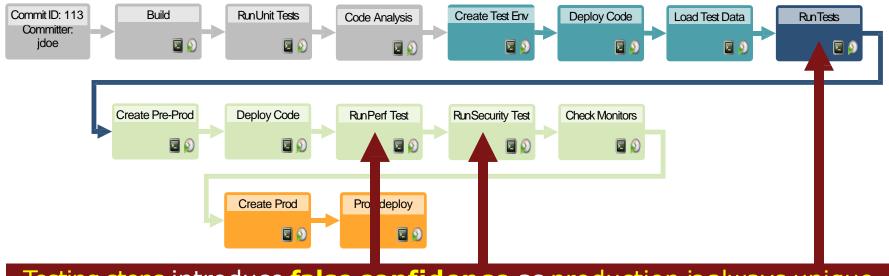


Features are being pushed to those supporting production in ever increasing numbers



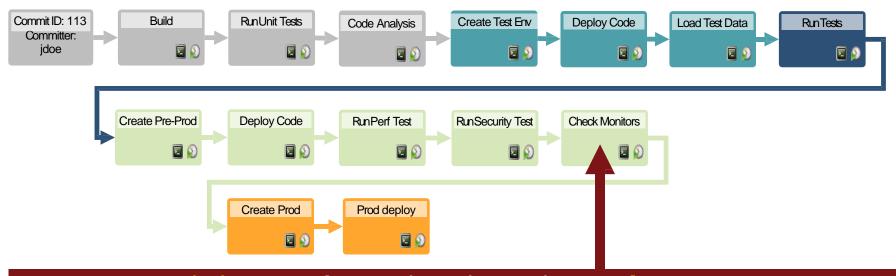






Testing steps introduce **false confidence** as production is always unique





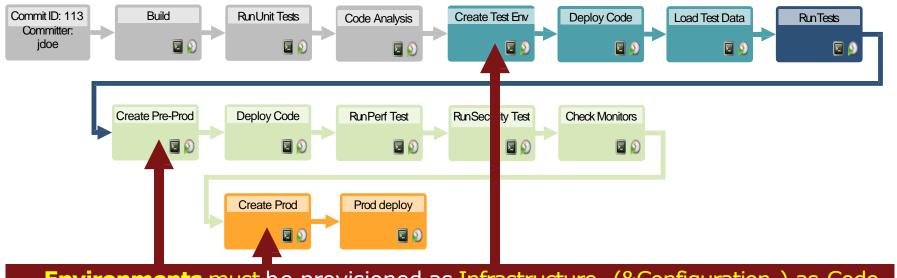
Monitoring and alerting is focused on things that are known to go wrong



SRE主导的自动化



Automation effort is "Ops" led ("shifting left"), to ensure reliability engineering priorities



Environments must be provisioned as Infrastructure- (&Configuration-) as-Code

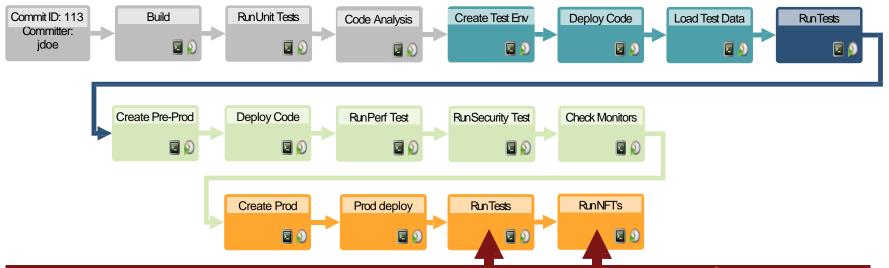


All code can be rebuilt from a **code repository** e.g. GitLab, Azure DevOps, Bitbucket.

- 1. Environments provisioned using Infrastructure/Config as Code
- 2. Automated functional and nonfunctional tests in production
- 3. Versioned (& signed) artefacts to deploy system components
- 4. Instrumentation in place to make the service externally viewable
- 5. Future growth envelope outlined
- 6. Clear anti-fragility strategy

	Components	Example Tools
Infrastructur e as Code (IaC)	Servers, network , storage	Terraform, AWS CloudFormation, Azure Resource Manager
Configuratio n as Code (CaC)	Software, dependencies, containers	Puppet, Chef, Ansible, Saltstack, Docker, GCP Deployment Manager





Automated functional and non-functional tests in production





Environment progression includes prod – Dev, Test, Pre-Prod, Prod (inc. "hidden live")

- 1. Environments provisioned using Infrastructure/Config as Code
- 2. Automated functional and nonfunctional tests in production
- 3. Versioned (& signed) artefacts to deploy system components
- 4. Instrumentation in place to make the service externally viewable
- 5. Future growth envelope outlined
- 6. Clear anti-fragility strategy

	Tests	Example Tools
Extend build pipeline	Automated functional	Selenium, Cucumber, Jasmine, Mocha, Zephyr, Mockito
Extend test pipeline	Automated non-functional	JMeter, <u>Sonatype</u> Nexus Lifecycle, SoapUI, <u>WhiteSource</u> , Veracode, Nagios



















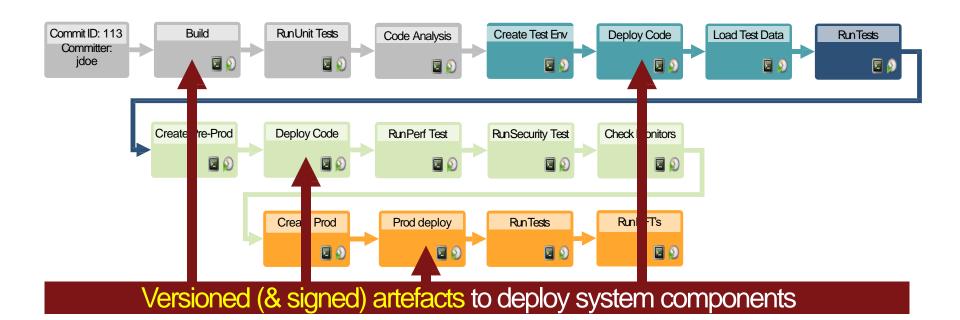














All service components, libraries and dependencies (or containers) stored in a artifact repository

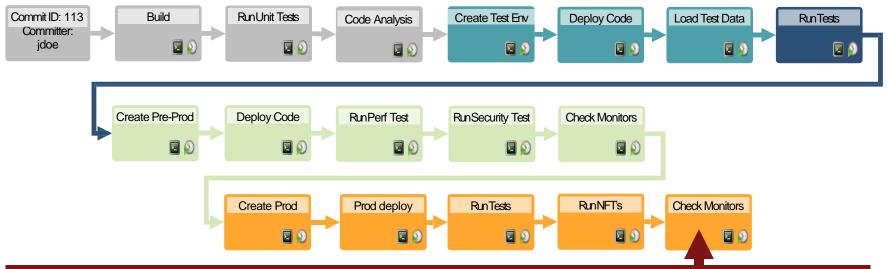
- 1. Environments provisioned using Infrastructure/Config as Code
- 2. Automated functional and nonfunctional tests in production
- 3. Versioned (&signed) artifacts to deploy system components
- 4. Instrumentation in place to make the service externally viewable
- 5. Future growth envelope outlined
- 6. Clear anti-fragility strategy

	Artifacts	How	Example Tools
	Digitally versioned	With semantic versioning x.y.z	Nexus, Artifactory
9	Digitally signed	For security and auditability	Nexus, Artifactory









Instrumentation in place to make the service **externally observable**



Alignment with SLAs, SLOs, SLIs and telemetry everywhere

- 1. Environments provisioned using Infrastructure/Config as Code
- Automated functional and nonfunctional tests in production
- 3. Versioned (& signed) artifacts to deploy system components
- 4. Instrumentation in place to make the service externally viewable
- Future growth envelope outlined
- 6. Clear anti-fragility strategy

	Consider	What	Example Tools
	Service Level Indicato rs	Are understood and published	OpsGenie
e	Instrumentation	Provides additional data and analytics	Nagios, Dynatrace, AppDynamics, Prometheus
t	Log files	Aggregated and ready for access	Splunk, LogStash







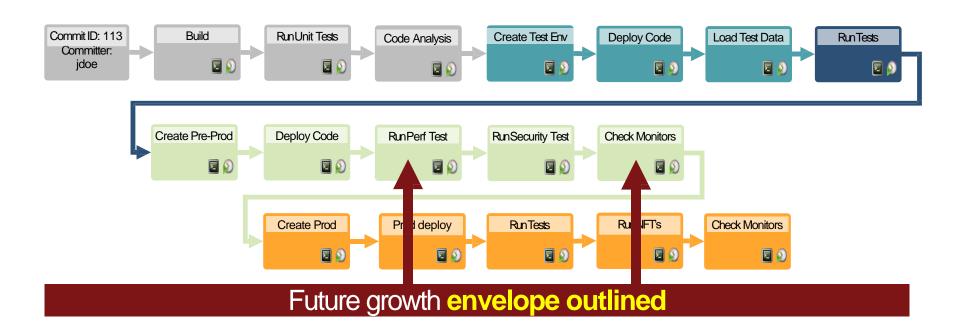














Current and future scale estimates

- 1. Environments provisioned using Infrastructure/Config as Code
- 2. Automated functional and nonfunctional tests in production
- 3. Versioned (& signed) artifacts to deploy system components
- 4. Instrumentation in place to make the service externally viewable
- 5. Future growth envelope outlined
- 6. Clear anti-fragility strategy

Consider	What	Example Tools
Autoscaling	In place	Amazon Cloud Auto Scaling, Kubernetes Pod Scaling
Administrati ve activities	Automated	Custom built tooling Cloud API's
Databases	Scalable	Amazon Cloud RDS, NoSQL-type databases like MongoDB, Couchbase

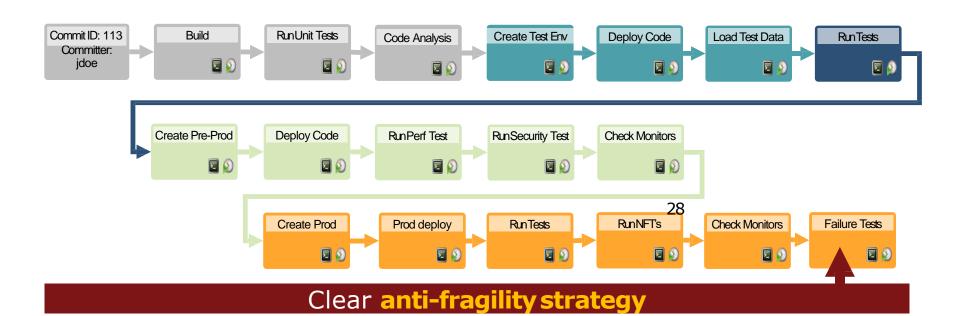














Current and future scale estimates

- 1. Environments provisioned using Infrastructure/Config as Code
- 2. Automated functional and nonfunctional tests in production
- 3. Versioned (& signed) artifacts to deploy system components
- 4. Instrumentation in place to make the service externally viewable
- 5. Future growth envelope outlined
- Clear anti-fragility strategy

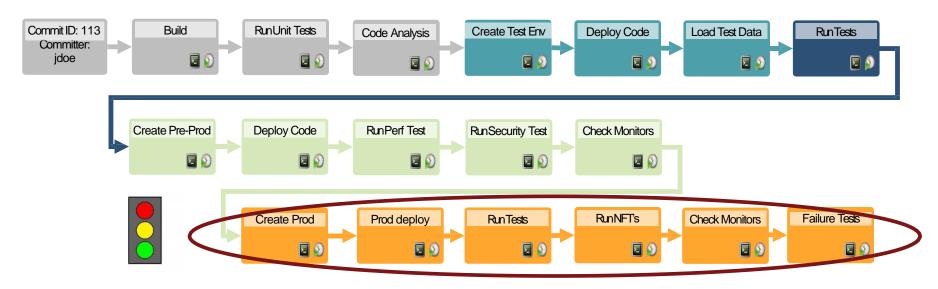
Consider	What	Example Tools
Disaster Recovery (DR)	Tests comple te	Fire drills
Chaos engineering	Practiced	Chaos Monkey
On call mechanis ms	In place	PagerDuty, VictorOps, Squadcast











Overall there is **more focus on prod** – DevOps gains **"wisdom of production"**, **SRE can say "No"**



III.SRE的新发展

- · 与DevOps统一文化、融合互补
- 可观测性异军突起、方兴未艾
- 引领平台工程、实现自助平台
- · FinOps降本增效、云服务公平公正

新发展之一:与DevOps统一文化、融合互补



Culture

Reliability @ Scale, Shift-Left "Wisdom of Production", and Continuous Improvement

Toil Reduction

Reduce Non-Value Add Work using Tooling and Automation

SLAs/SLOs/SLIs

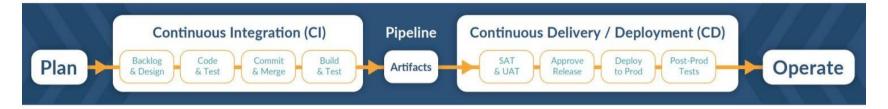
Metrics such as Availability, Latency, and Response Time with Error Budgets

Measurements

Observability, Monitoring, Telemetry, and Instrumentation

Anti-Fragility

Improve Resilience using Fire Drills, Chaos Monkey, Security and Automation



Work Sharing

Work Technical Debt in Small Increments

Manage Load % for Ops, Dev and On-Call Work

Deployments

Gradual Releases using Green/Blue, A/B, Canary Deployments, Automation Scripts, Testing and Monitoring

Performance Management

Monitoring, APM, Capacity Testing & Auto-Scaling

Incident Management

Emergency Response, 50% Ops/Dev Load, 25% On-Call Load, and Blameless Retrospectives

© DevOps Institute. All rights reserved.

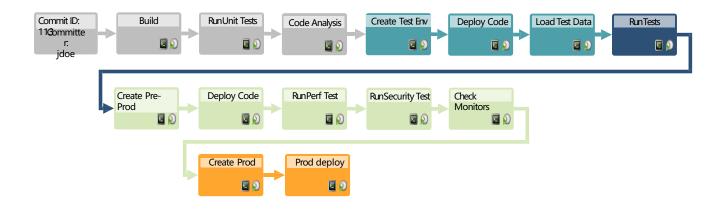
www.DevOpsInstitute.com

SRE负责整个产品生命周期,关注整体规划和交付共享平台,促进统一的SRE和DevOps文化

新发展之一:与DevOps统一文化、融合互补.. 🔃 SRE 专奏会



经典的DevOps自动化

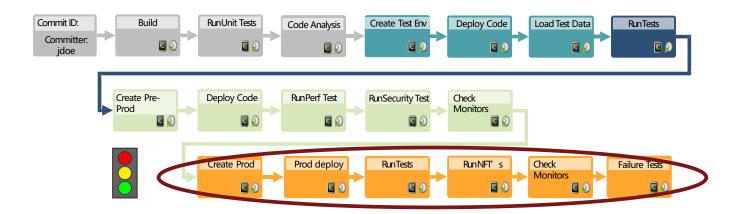


大量的自动化工作是"**Dev"领导的(从左到右),流水线大大提高了研发交付效率,但同时** "重Dev、轻Ops"的特点,落地中存在很多问题和挑战。

新发展之一:与DevOps统一文化、融合互补.. 🔃 SRE 专奏会



SRE 主导的服务自动化



总体来说,更关注生产环境(PROD)——使DevOps获得了"生产智慧",针对自动化流水线的风 险 , SRE <mark>可以说"不"。</mark>

新发展之一:与DevOps统一文化、融合互补







3. SRE 和 DevOps

一个成功的技术团队对其组织的贡献不只是交付代码——需要比交付高质量代码贡献更多 内容。要确保提供的服务保持高可用、高性能,并且随着时间的推移与用户的期望保持一致。 可靠性是衡量团队履行这些承诺的标准,今年我们继续探索软件交付和运维的可靠性这个 因素。

站点可靠性工程 (SRE) 是一种有影响力的运维方法,起源于谷歌,现在在许多组织中得到应用。SRE 优先考虑经验学习、跨职能协作、广泛依赖自动化以及服务级别目标 (SLO) 等度量技术的使用。其他前沿的运营实践采用类似的方法,只是命名不同。因此,为了尽可能客观地评估这些方法落地的程度,我们十分注意在我们提供给受访者的调查问卷中使用中性的描述性语言。我们还收集有关可靠性工程成果的数据: 团队能够实现其可靠性目标的程度。输入和输出——SRE 实践和可靠性成果——与其他 DevOps 能力均反映在我们的预测模型中。

1) 稳定性至关重要

在我们调查的团队中采用 SRE 很普遍:大多数受访者在使用我们调研问卷中的一种或多种 实践。在如此广泛的团队中,数据揭示了可靠性、软件交付和结果之间的微妙关系:当可 靠性较差时,软件交付效能并不能预言组织的成功。然而,随着可靠性的提高,我们开始 看到软件交付对业务成功的积极影响。

谷歌:没有稳定性,软件交付效能就无法确保组织成功

新发展之二:可观测性方兴未艾



"我认为,在未来3年内,这三 个类别—APM、监控/指标、日 志,可能还有其他类别—都可能 不复存在。将只有一个类别: 可 <u>从测性</u>。并且它将包含您需要了 解系统可以进入的任何状态所需 的所有洞察。"

> Charity Majors CEO Honeycomb



新发展之二:可观测性方兴未艾..



监控和可观测性

- 分布式、复杂的服务以不可预测的用户和可变吞吐量大规模运行,意味着有数百万种不同的方法出错
- 但是我们不能预测它们(监控神话)
- 刚需:将服务的所有输出外部化,使我们能够推断该服务的内部状态(可观测性)

新发展之二:可观测性方兴未艾..



举例说明: SLO, SLI& 可观测性

- · SLO 从用户的角度,帮助确定什么是重要的
 - 例如,90%的用户应在一分钟内完成完整的付款交易
- · SLI 详细介绍了我们目前的表现
 - 例如,98%的用户在一个月内在不到一分钟内完成付款交易
- 可观测性 使用服务的正常状态
 - 38秒是用户完成付款的"正常"时间,这时整体服务都健康

新发展之二:可观测性方兴未艾..



O'REILLY"

Observability Engineering

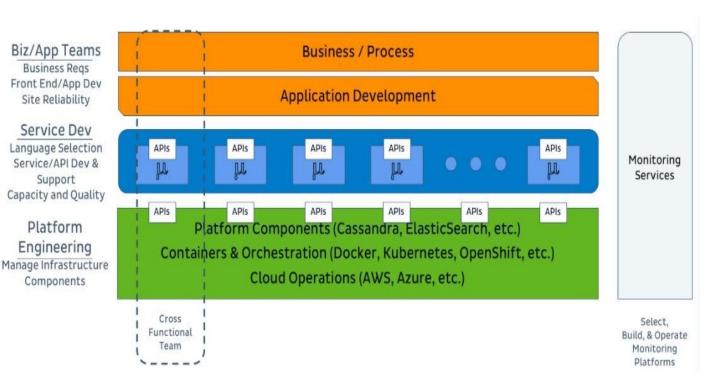
Achieving Production Excellence



2023年,中国SRE社区 将发起OREILLY 专著 **《Observability** Engineering》(可观 测性工程)在线读书分 享,期待参与...

新发展之三:引领平台工程,实现自助服务





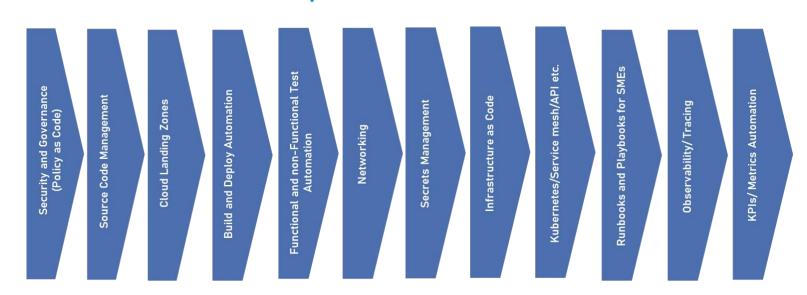
- SRE不再需要管理服 务器更新、操作系统 升级或任何过去浪费 时间的任务(琐事)
- 现在,SRE确保平台 组件、配置正确,并 作为服务提供给开发 和运维使用

SRE负责平台工程,采用"内部开源"模型进行进一步的开发

举例:SRE开启自助服务



- 平台提供了"自助"的基础设施、功能、配置和环境,可供开发团队以及第三方使用
- 嵌入式治理、控制和标准是内置的
- 端到端部署自动化,服务或应用程序的基础架构剧本
- 通过运行手册和剧本为多/混合云抽象特定于基础设施的实现
- 平台团队构建的产品可以通过SRE/DevOps或任何其他方式进行扩展或增强



新发展之四:FinOps推动云服务公平、降本增效





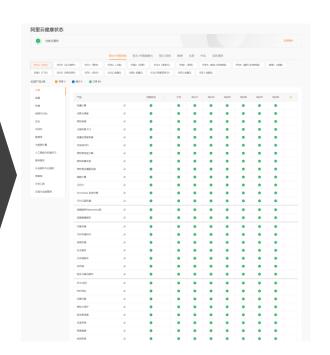


公有云账单居高不下,云服务的财务管理成为企业重要痛点

新发展之四:FinOps推动云服务公平、降本增效



国内公有管 到示 现 到企 来 对 强 不 说 来 求 求 平 或 需 降 平 公 正



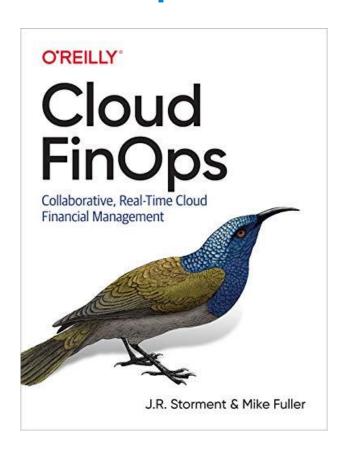


https://status.aliyun.com/

https://status.cloud.tencent.com/

新发展之四:FinOps推动云服务公平、降本增效





2024年上半年,中国SRE 社区拟发起OREILLY 专著 《Cloud FinOps》(云 成本优化)在线读书学习, 期待参与…



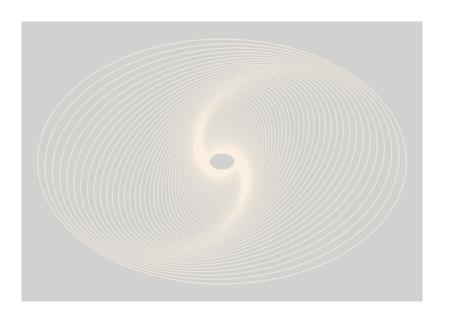
IV.SRE是新运维的"奇点"



奇点概念

奇点是一个物理学概念

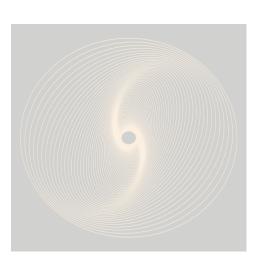
- 1.世界万物的起源是一致的
- 2.一切已知物理定律均在奇点失效





SRE:新运维奇点

在开发和运维之间----引入和推动"奇点"



SRE本质上鼓励一种文化,即在应用程序及其生态系统之间,在开发与运维之间,在软件工程和系统工程之间,引入和推动奇点。

SRE的愿景是确保通过满足客户期望的可靠服务来交付业务价值。



V.中国SRE社区介绍

中国SRE联盟-2021年

CHINA SRE ALLIANCE- 2021

- •于2021年4月,正式登记为中关村人才协会下的"SRE专委会"专业机构(北京民政局合法 注册);
- •于2021年12月,正式加入中国计算机大会(CNCC),并成功举办中国SRE第一次高峰 圆桌论坛;

30*

SRE宣讲

SRE理事单位

SRE成员单位

中国SRE社区 成立情况

中国SRE联盟发起人

INITIATOR OF CHINA SRE ALLIANCE



王钧





刘征



社区官网: https://www.srenow.cn

2023年度工作计划





2023年SRE联盟工作规划

- 正式运营技术专业组(可观测性、运维机器人、 DevSecOps等)
- SRE双周读书分享 (20次、含可观测性等)
- SRE年度会议 (2次, SRE年度大会+可观测性)
- ·SRE学术专著的引入和翻译 (1本)
- SRE白皮书研讨&发布 (1本)

2023年度专委会工作专委会工作汇总(暂定):

- 1.成立技术专业组
- "可观测性、运维机器人、开源软件" 三个 专业组
- 2. SRE双周读书分享
 - —20次读书分享(SRE+OE, 2本书)
- 3. SRE年度会议
 - —SRE年度大会2次; SRE MeetUp 4次)
- 4. SRE学术专著的引入和翻译(待定)
- 5. SRE白皮书研讨&发布
 - 一与信通院合办1期可靠性白皮书

年度SRE大会及全国MeetUp安排



I. SRE专委会大会:

- 上半年SRE专委会大会
 - 一 落实1次(杭州、上海)
- •2023 中国SRE大会
 - 一落实1次(待定)

II.全国MeetUp安排:

- 华南区(深圳、广州/珠海)
- 一 落实2次(已完成1次)
- 华东区(上海、杭州)
- 一 落实1次(待落实)
- 华北区(北京、大连/西安)
- 一 落实1次 (待落实)

信通院SRE标准合作事宜



CAICT 中国信通院 23年工作计划



 • 01
 • 02
 • 03
 • 04
 •

 标准制定
 研究报告
 技术活动
 行业服务

纵向挖掘业务 稳定性保障之法

- · 《业务稳定运行成熟度模型》
- 《电信业务系统稳定性评估方案》
- 《电信业务系统稳定性成熟度模型》
- 《"一云多芯"云平台稳定性评估模型》
- · 《容量管理平台能力要求》
- 《金融场景系统稳定性成熟度模型》
- 《证券基金期货稳定性保障模型》
- 《根因分析平台能力要求》
- 《金融场景全链路压测技术能力要求》
- · 云服务稳定性运行评估方案:
 - 《容器集群稳定性评估》
 - 《CDN稳定性评估》
 - 《网络设备稳定性评估》
 - 《云主机 (含宿主机) 服务稳定性评估》
 - 《云存储服务稳定性评估》

深化技术研究

- 《可观测性技术发展白皮书》
- 《中国混沌工程调查报告(2023)》
 - 《中国混沌工程技术实践指南》
- 《一云多芯 发展报告 (2023) 》
- 《一云多芯系统稳定性建设指南》

9场技术沙龙

系統稳定性沙龙: 2月(SRE主题)3月、4月、5月、7月、9月、

推进全国沙龙巡演

3月、4月、5月、7月、9月、 10月、11月、12月

2场行业大会

- 系统稳定性大会: 6月
- 可信云大会——系统稳定性分论坛

落实行业服务 及政府支撑

- 行业平台:系统稳定性保障官网, 提供混沌工程工具、模糊测试工具 及ChaosHub
- 政府支撑: 持续支撑工信部信管局 "稳保行动"

2

SRE知识体系和国际研究机构



成熟的SRE需要掌握多方面知识,包括DOI在内的国际研究 机构在此方面有持续的积累和官方认证,主要包括:

- Site Reliability Engineering
- Observability
- AIOps
- DevSecOps
- DevOps Engineering
- DevOps Foundation
- DevOps Leadership
- Value Stream Management
- Agile Service Management
- Continuous Testing



https://www.devopsinstitute.com/certifications

欢迎联系





liu feng

L海 浦东新区



扫一扫上面的二维码图案, 加我微信