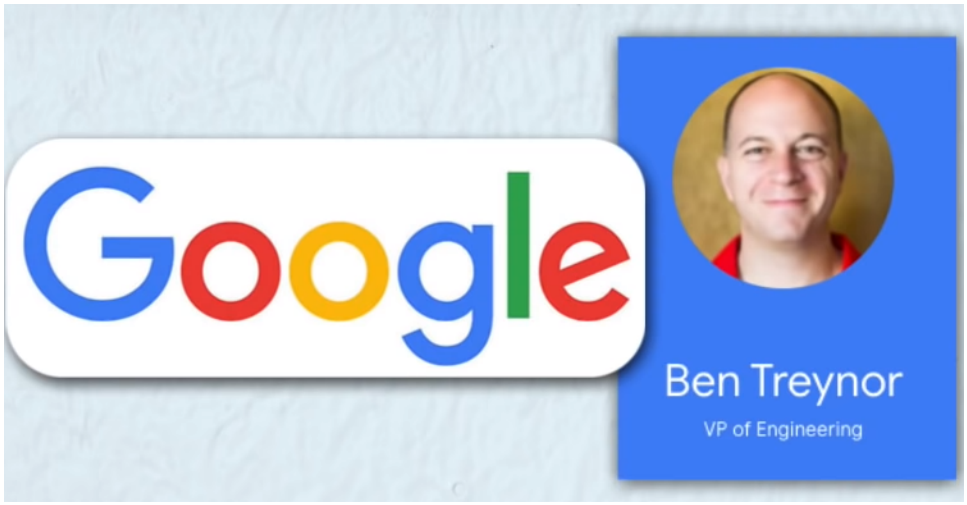
Storie 1 : Introduction to SRE

Introduction to SRE

SRE — Site Reliability Engineering

SRE, which stands for Site Reliability Engineer, is a specialized position at the intersection of software development and operations. The primary objective of an SRE is to ensure the reliability and stability of systems and services. Unlike traditional DevOps teams, SREs focus explicitly on maintaining and improving the reliability of software systems.

SRE Term originator is



SRE Team

SRE Team is made of software engineers
Improve the software their system/services.

Why we need SRE

As in Devops team we don't have any dedicated role for keeping system reliable.

What is Reliable

Reliability, in simple terms, means you can count on something to do what it's supposed to do, consistently and without unexpected problems.

Example:

1. If you have a reliable car, for example, you can trust that it will start and run smoothly every time you use it.
2. if a service of bank is down when we need then its not available to use its a unreliable service.

Unreliable services causes It its unreliable then Loss of revenue(we cant order any thing online)unhappy customers

Reliable service : AWS, azure ,Instagram, FaceBook ,WhatsApp etc.

Service Level Agreements (SLA)

SLA represents the commitment between a service provider and its customers. Achieving 100% SLA is unattainable, but various levels such as 99.9%, 99.99%, and 99.99999% indicate the acceptable outage duration, translating to minimum downtime.

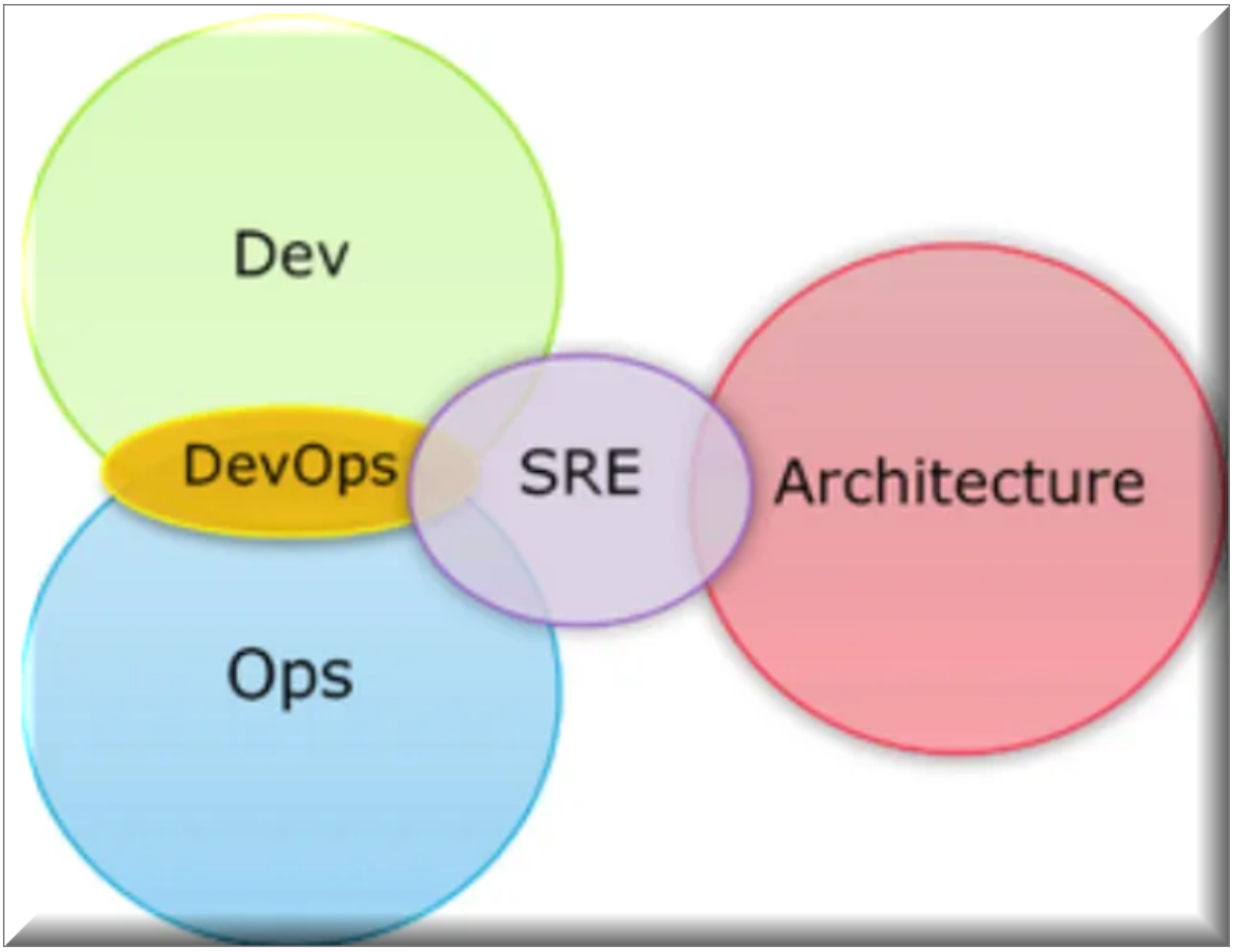
How it becomes unreliable ?

unreliability can occur through various means:

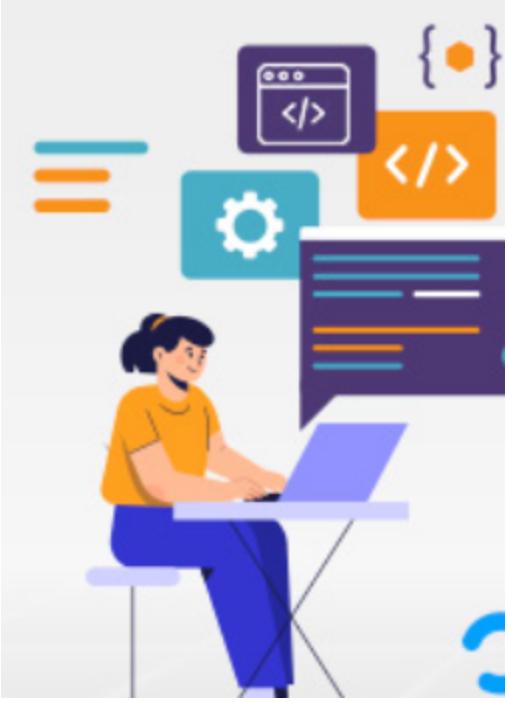
1. **Changes:**
   Introducing alterations to the system, whether in infrastructure, platform (e.g., Kubernetes), or the service/application itself.
2. **Infrastructure Modifications:**
   Modifying the foundational elements that support the system, which can disrupt its normal functioning.
3. **Platform Adjustments (e.g., k8s):**
   Implementing changes in the platform, such as Kubernetes adjustments, which may introduce instability.

4. **Service or Application Updates:**
   Making modifications to the core service or application, potentially leading to unforeseen issues and reduced reliability.

we can't avoid making changes to the system if we want to constantly improve its performance and make it better over time.



DEV Team - The development team consistently makes changes to the code.



OPERATION TEAM :

The operations team examines number of checklists so that they can maintain stability.
Every release risk responsibility.

CHECKLIST:

In DevOps, a checklist is a set of items or tasks that need to be verified or completed to ensure a smooth and successful development and deployment process.

Checklists in DevOps typically cover various aspects, including

- code quality
- infrastructure provisioning
- testing
- deployment steps and

- monitoring configurations.

SRE : Team aims to automates the process

Scope of SRE :
SRE is responsible for ensuring everything is running smoothly. The things that SRE's are responsible for include:

- Incident Response -- responding to incidents in production environments
- Post-mortems -- root cause analysis, preventing the issue in the future
- Monitoring & Alerting -- creating and using dashboards and alerting mechanisms
- Capacity Planning -- planning new infrastructure to accommodate application demands

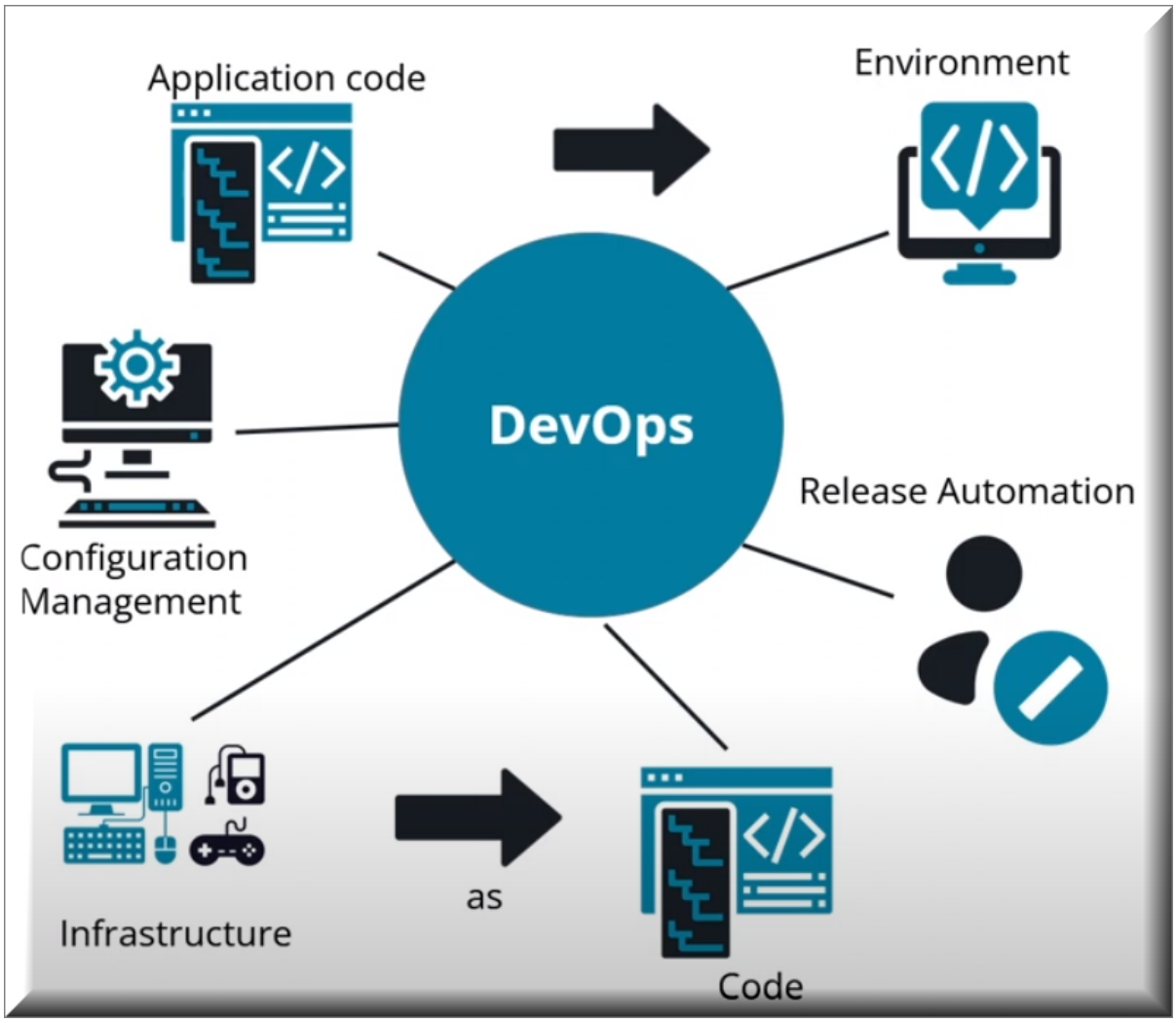
SRE Team coordinates with Different teams like

Devops Team
System Admin and

Developers Team

in order to resolve the issue

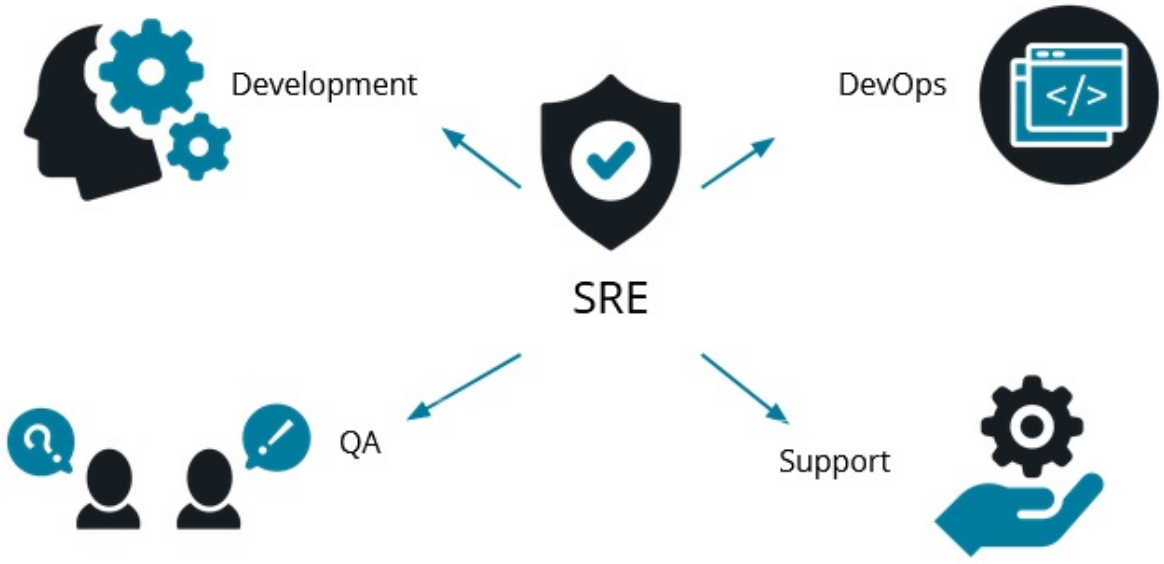
Devops Responsibilities



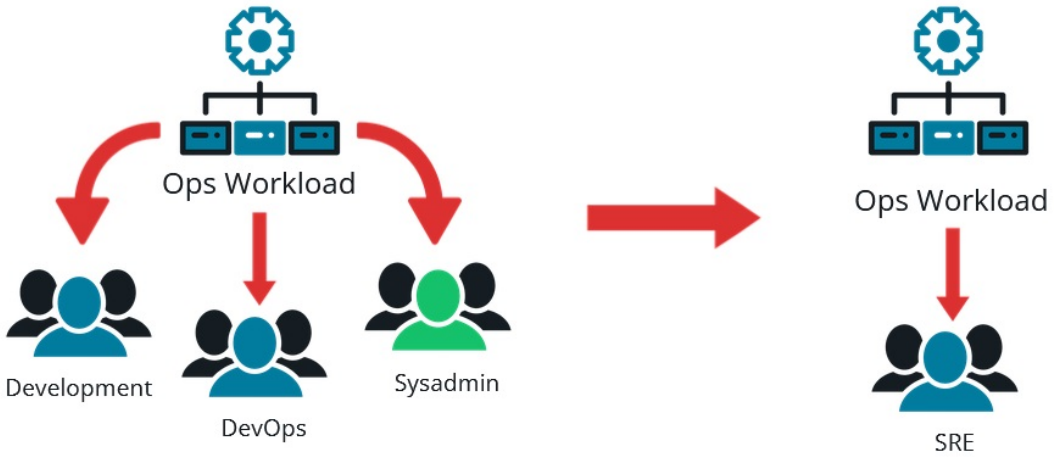
SRE Depend on

It really depends on the organization but the teams you work with could include:

- **Development** -- you will probably interface with this team the most, as there are a lot of overlapping tasks.
- **DevOps** -- since DevOps typically owns the delivery of software, they will be interested to know how well it is performing for end users.
- **QA** -- SRE and QA may have more of a consulting relationship since both teams use automation
- **Support** -- this team is on the front line with clients and may need a hand from SRE teams for more technical issues.



SRE Team



Use Case of SRE Team

When a company creates a website or web service that is available for users to connect to and consume, unforeseen events happen which cause the operations of the site or service to be interrupted. These include:

- Connection problems
- Unavailability of the service
- Errors within parts of the service

Teams need to respond in order to restore the service.

Without an SRE team, other teams share parts of the ops work, leaving Ops fragmented and disorganized.

For example, the Development team may have documentation on how to resolve an issue, and the Sysadmin may not know about it.

This also takes away from the different team's primary focus, which could cause them to miss delivery dates or the implementation of new technology.

With an SRE team, the ops work is centralized to one team, freeing up the other teams to focus on their responsibilities.
And the SRE team would own and manage the Ops work, making the resolution of issues more efficient and streamlined.

New Terms most know as SRE

latency - the time it takes to send a request and obtain a response

saturation - a measure of the amount of traffic

errors - failed requests

traffic - how many requests are happening

Example Think of it as a highway:

traffic measures all of the cars on the highway

latency measures how long it will take a car to get from point a to point b

saturation measures how busy the highway is

errors measure the number of cars that are broken down