Stories 2 : SRE Practices and Roles and responsibilities

SRE Teams

SRE Teams can help us improve enterprise workflows in the following manners:

- Reduce overhead and improve the efficiency of IT Departments
    - Automate tasks and reduce toil; with an SRE team's focus, tasks that are not normally automated can be, which helps the IT Department as a whole.
- Reduce the risk of an organization
    - Make sure you have a plan for when someone needs to be on call, and create guides for what to do. Instead of passing the on-call pager around between different teams, let one team handle it and follow the guides to fix problems faster.
- Improve uptime
    - Understand the root cause of outages, and prevent them in the future. By making a conscious effort to understand and prevent an outage, it will not be repeated.

> **New terms:** "T**oil**" in Site Reliability Engineering (SRE) refers to the **repetitive and manual work** that people have to do to keep a computer system running smoothly. It includes tasks like fixing routine problems, responding to alerts, and doing regular maintenance. SREs aim to minimize toil by using automation and smart engineering solutions so that they can focus on more important and

interesting aspects of improving and maintaining a system's reliability. The idea is to reduce time spent on boring, repetitive tasks and use that time for more valuable and creative work.

SRE practices

A few common SRE practices that we will cover in this lesson include:

- on-call and emergency response
- post-mortems
- Reliable product launches

On-call & incident response strategies:

- Primary/secondary contacts are on-call one week a month. With a secondary contact, if you happen to be away from the phone, the call will divert to that person.
- The on-call contact only handles essential events.
- Instituting a "follow the sun" model to eliminate night shifts.
- Limit the number of incident responses to two per shift.
- Have playbooks (see the **example** below) that are comprised of escalation paths and incident-management procedures.

Post-mortems:

Events that occur which lead to the creation of post-mortems:

- Data-loss
- On-call engineer getting involved, which comes from an alert or a phone call
- Failures with monitoring or visible downtime

Items to include in a post-mortem:

- A record of an incident and its impact
- Actions taken to mitigate the incident
- The incident root cause
- Any prevention measures
- Identification of the causes of system failure, not pointing fingers

Instituting a blameless post-mortem culture:

- The problem and system failure
- Involving everyone on the team
- Collecting thoughts and solutions from smart people
- Finding a permanent solution

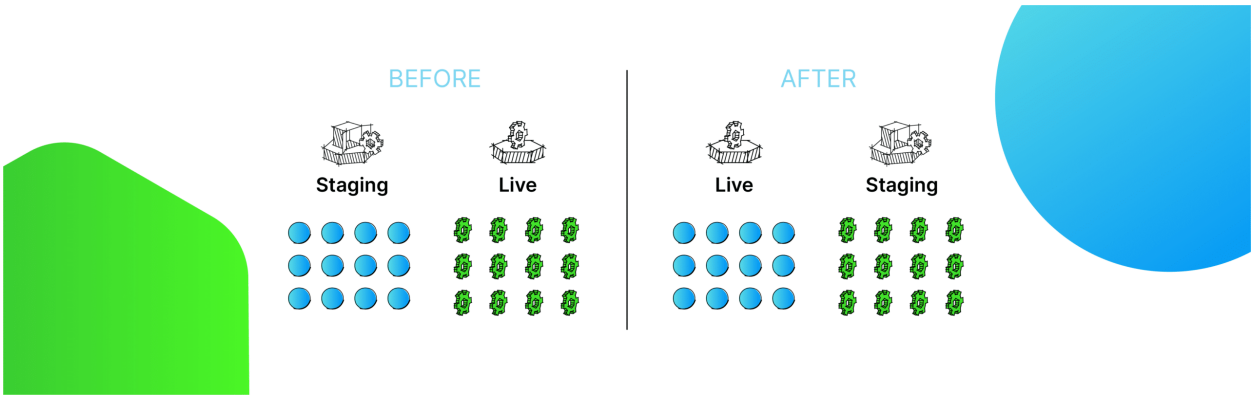
Reliable product releases include:

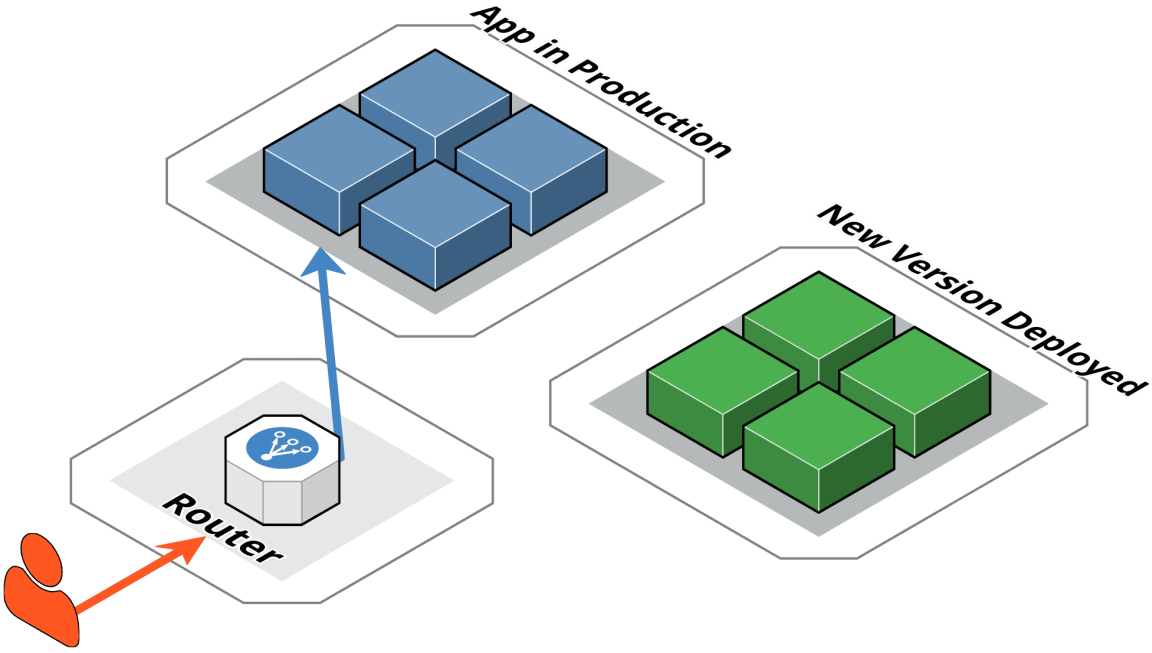
- Architecture and dependencies
    - Latency requirements : Refer to the specific time constraints or delays that are acceptable
    - Request flow from front-end to back-end
    - The volume of requests
- Integration consists of:
    - DNS

    - Load balancing
    - Monitoring
    - Cloud front CDN(Edge location) or redis cache
- Failure Modes** **are:
    - Single-points of failure
- Process and Automation highlights:
    - Any manual process
    - Documentation
- Rollout Planning involves:
    - How the change goes live
    - **Canary deployments** ：Instead of releasing a new version of a software application to everyone at once, a canary deployment involves gradually rolling out the new version to a small subset of users or servers.

Blue-green deployment is a software release management strategy that aims to minimize downtime and reduce the risk associated with deploying new versions of an application. The basic idea is to have **two identical environments,** referred to as "blue" and "green," and only one of them is live at any given time.

Here's an overview of how the blue-green deployment model works:



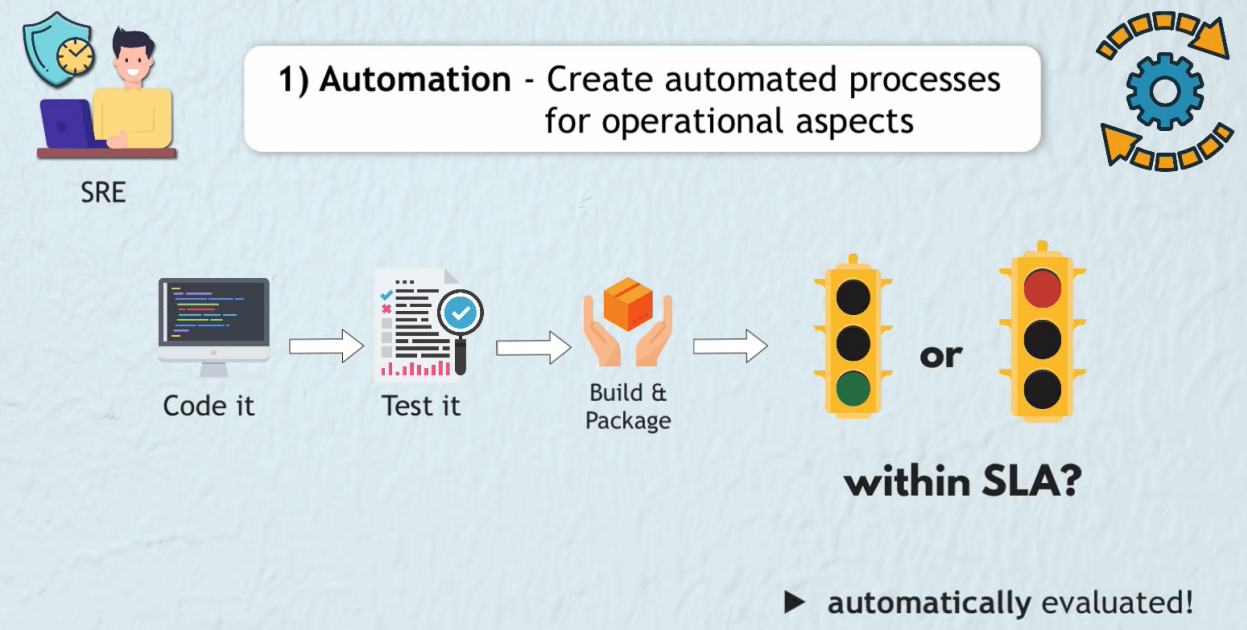


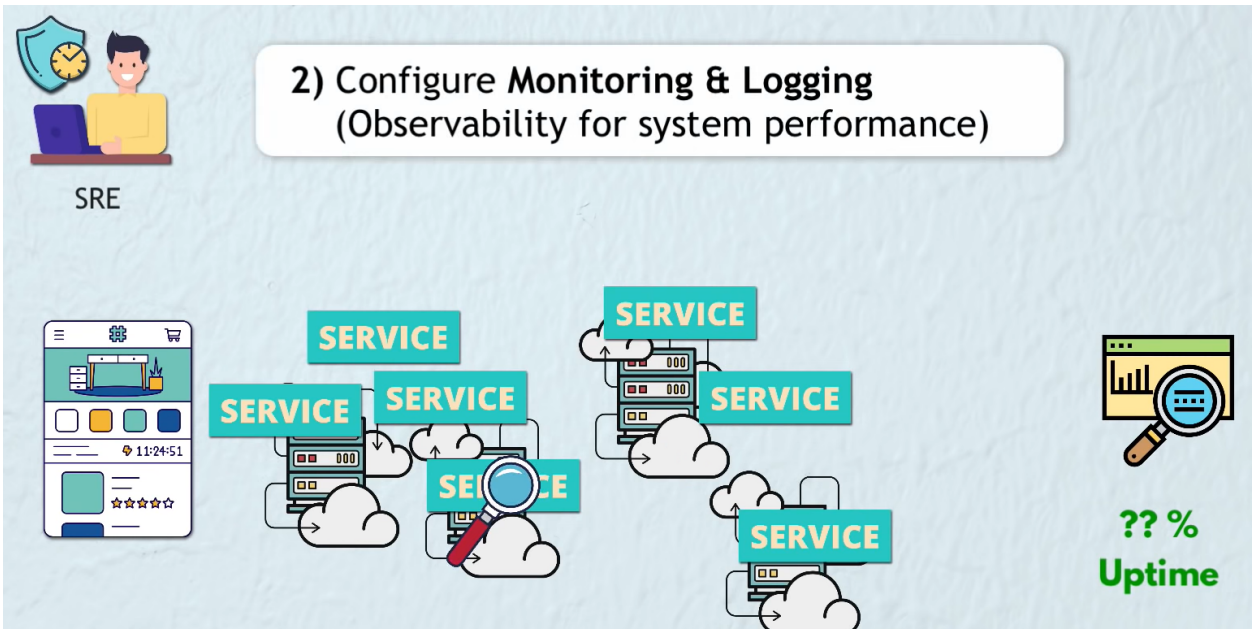
1. **Blue Environment (Production):**
    - This is the currently live and stable version of your application.
    - Users access and interact with this environment.
    - Any updates or changes are made in the "green" environment.
2. **Green Environment (Staging or Testing):**
    - This is an identical environment to the blue one.
    - It is a replica of the production environment where new changes, updates, or versions are deployed for testing.
    - The green environment is not exposed to users during the deployment process.

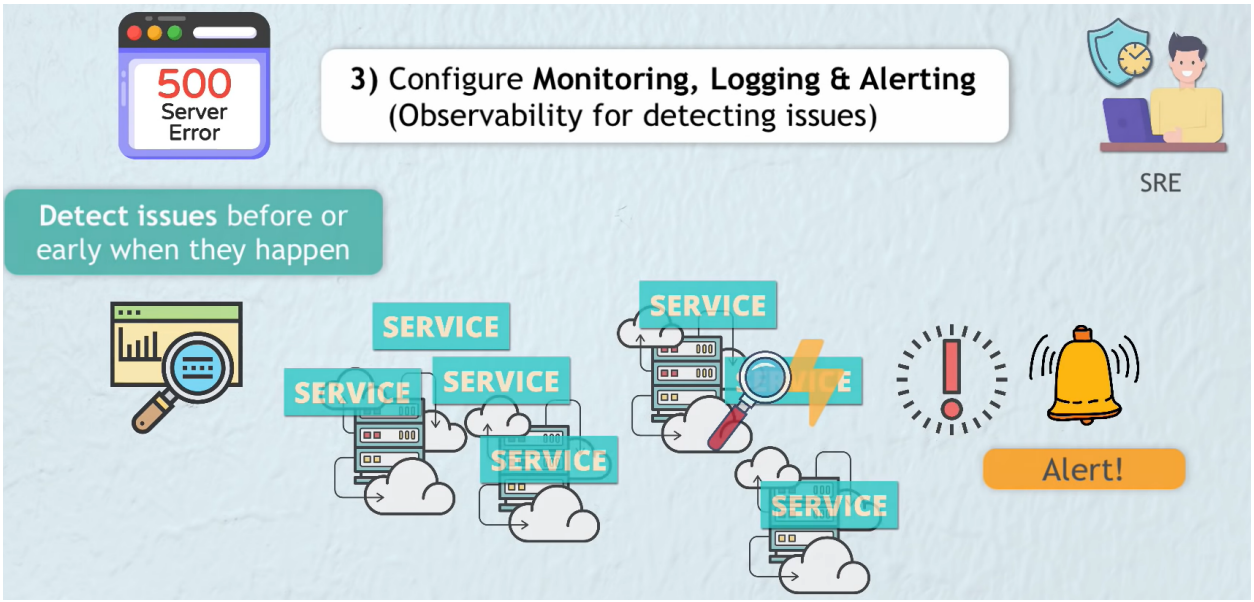
Roles and responsibilities

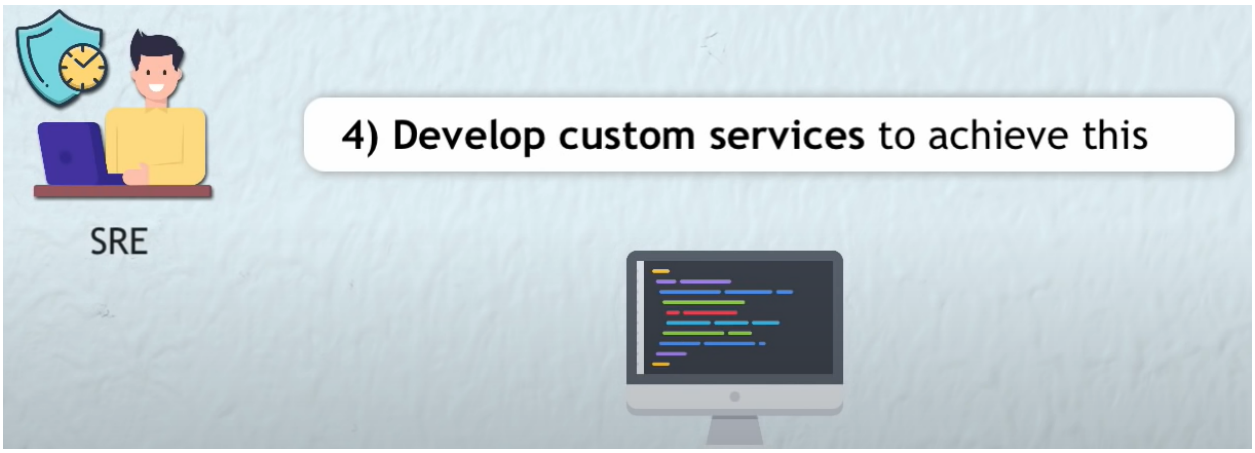
1. Automation : create automated processes for operational aspects

2. configure monitoring and logging (observability for system performance)
3. configure monitoring, logging and alerting(observability for detecting issues)
4. Develop custom services to achieve this.
5. on call support




2) Configure Monitoring & Logging
(Observability for system performance)
SRE
11:24:51
SERVICE
SERVICE
SERVICE
SERVICE
SERVICE
SERVICE
?? %
Uptime


500
Server
Error
3) Configure Monitoring, Logging & Alerting
(Observability for detecting issues)
SRE
Detect issues before or
early when they happen
SERVICE
SERVICE
SERVICE
SERVICE
SERVICE
SERVICE
Alert!



GOAL OF SRE :

short duration of outage

few people effected

few services effected