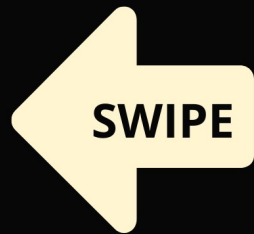




**#ASLI ENGINEERING**

# GitHub Outage Downtime they thought was avoided



**BY**

**ARPIT BHAYANI**

# Dissecting GitHub Outage

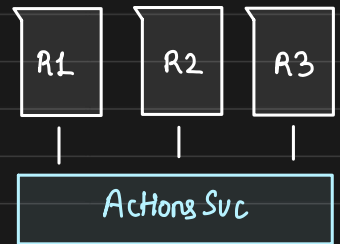
Downtime that they thought was avoided

On March 1, 2021

- High error rate in Repo Check Service, Jobs failed on queued
- Search to find workflow was returning incomplete results

## Context

GitHub Actions are triggered  
when some state/event  
happens on a repo



Before an action is run, a check suite is created

\* for every action

Repo changed → Action Triggered

Create check suite

Run the action



↑  
when you commit, we see checks  
on the repository

Upon every event that the check runs

it is given a unique ID ← integer

Given that checks run upon every important event, "commit", it would lead to a lot of entries for each check execution



Table could explode with this high freq activity

When any table "explodes"

When there are large number of rows that we insert in table, the table's AUTO-INCREMENT ID column might reach its limit.

A 32-bit integer column would exhaust at  $\sim 2B$

GitHub's Anticipation and preparedness

GitHub anticipated this! and they migrated all relevant columns to BIGINT "six months back"

But then, why are we talking about this?  
What went wrong?

What exactly happened?

Check suite IDs exceeded INT32 range

So, DB is not the

culprit here

↳

GitHub Team already migrated DB columns

GitHub uses some GraphQL library

that supports INT32 when unmarshalling JSON

↓  
JSON string to Object/Dict/Map

When some job needs to run on repo

- create check suite

INT64

↓

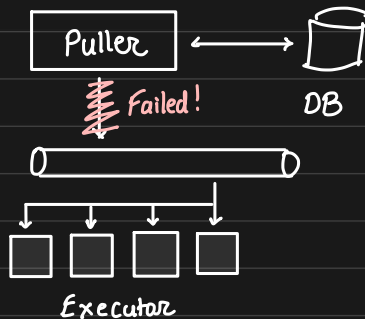
↳ creation of check suite worked fine because DB supported

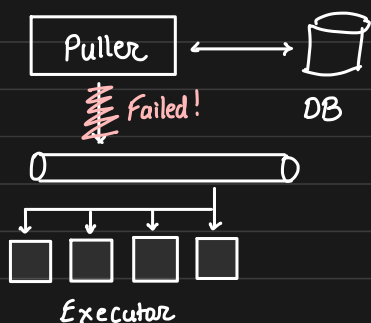
- processing the data failed

↳ some one would pull data from DB and process

The executor/scheduler  
when fetch data from DB

failed as it only supported  
INT32.





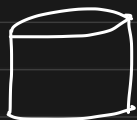
Jobs failed to be queued

↓  
check suited were left in pending state

Mitigation: Pushed a code fix so as to ensure that tasks are queued for execution

Another impact: Search

Searching for workflows via UI & API was incomplete



Say, they use ES as their search engine.

Elasticsearch

Every doc added should have an ID

Because the library was not able to pull data from DB,

the workflows never got indexed in ES

and hence search results were incomplete.