Day - 3

25 June 2025 10:06

SLO :-

SLA = 99.7%

SLO

99.85 → warning, → failure

99.9% → good.

SRE – Site Reliability Engineer → Google[team]

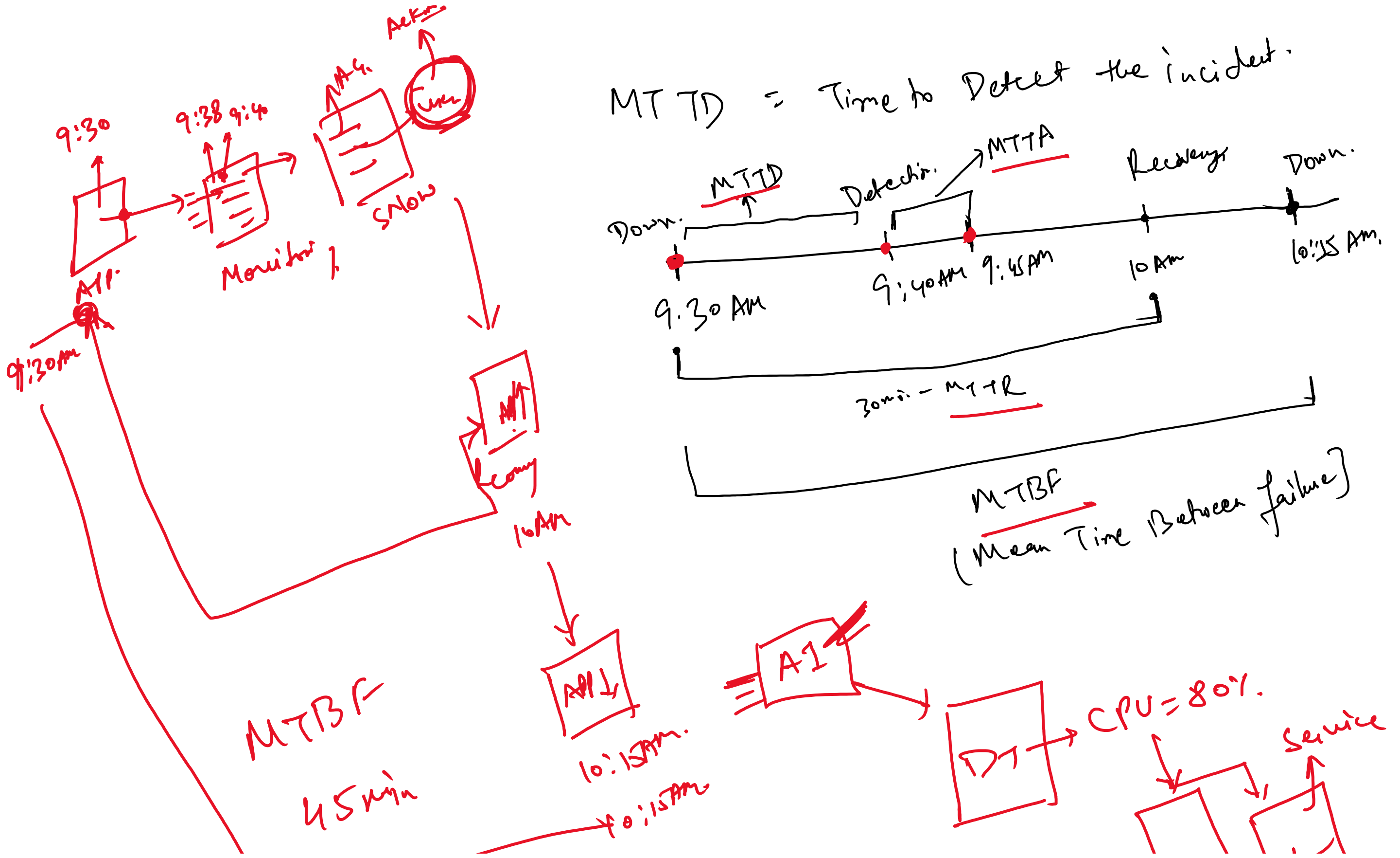
1. SLI → Service level Indicator → CPU = 80%.
2. SLO → Service level objective → SLO → 99.6% > SLA
3. SLA → Service level Agreement → 99.7%.
4. Error Budget → 0.8% → 5.76 hours
5. MTTR → Mean time to recovery.
   ↓
   how quickly a system recover from failure.

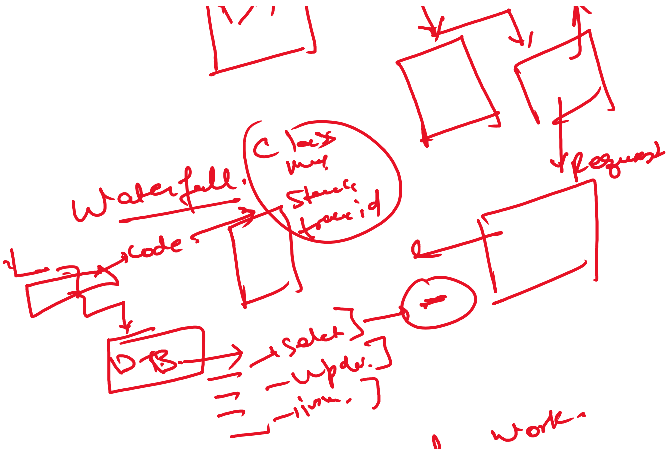
CPU = 80% → Script, Notification.
↓
CPU = ↓ ← Restart.

$$MTTR = \left(\frac{\text{Total Downtime.}}{\text{No. of incident}}\right)$$

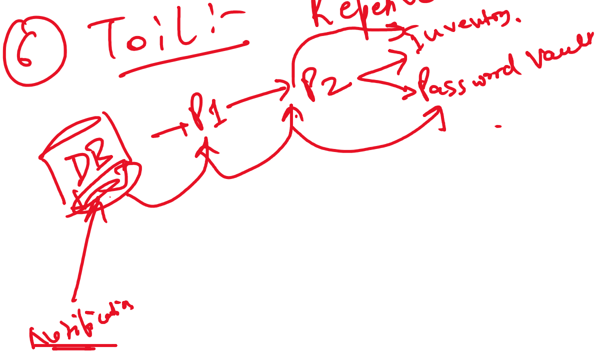
$$= \left(\frac{250\text{ min.}}{5}\right) = 50\text{ minutes.}$$

MTTD = Time to Detect the incident.





⑥ Toil:- Repetive manual work.



SLO:-

1. Service level Availability
2. Single Request.
3. Response time level.
4. Synthetic SLO
5. Synthetic step SLO.

1. Service level Availability:- Define the SLO on the Service level.

2. Single Request:-

(Successfull recommendatio / total recommendation) * 100

Burn Rate = (Actual error rate / rate) (error rate / error budget)

$$\text{Burn Rate} = \left(\frac{\text{Actual error rate}}{\text{Allowed error rate}}\right) \rightarrow \left(\text{Error budget}\right)$$

↓

how Quickly, you will exhaust the error budget:-

30min → 0

SLO = 99.9%

Error budget = 0.1%

Burn rate = 1 → error budget steady rate.

Burn rate < 1 → Burn slow than expected

Burn rate > 1 → Burn the error budget much faster.

SLO = 99.9%

error budget = 0.1% → (30 day) → 43.2 mins.

Used = 21.6 min in just 3 days

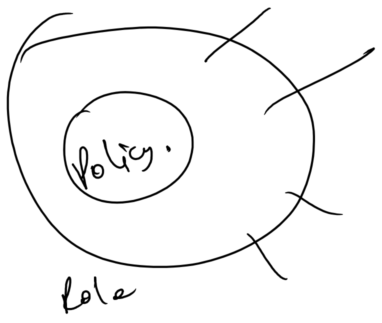
$$\text{Burn rate} = \frac{21.6}{3\times(43.2/30)} = \frac{7.2\times30}{43.2} = 5$$

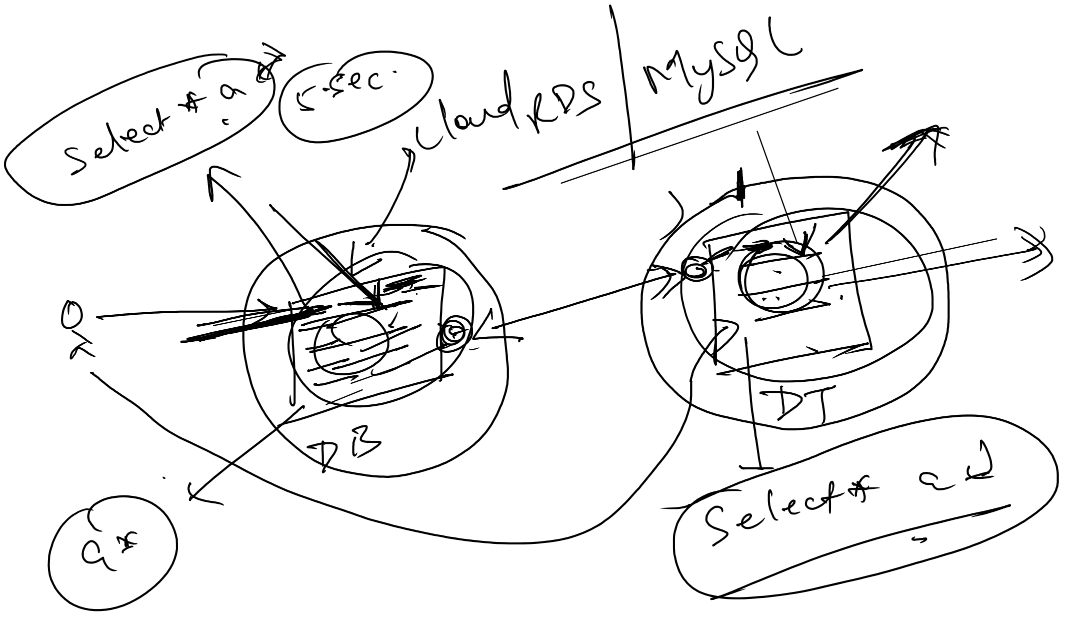
= 5× faster → error budget →

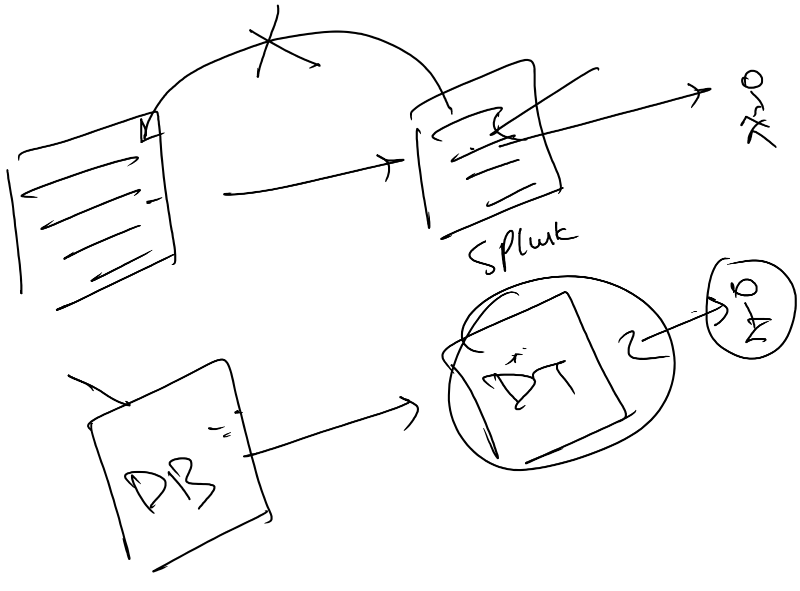
Why?

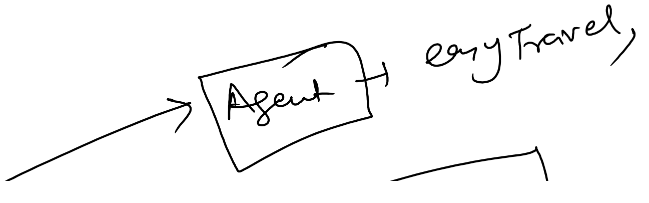
1. Detect incident early, bf breaching SLO
2. Priortize recovery over feature release.
3. Drive alerting logic. (eg: alert if BT > 2 over 1 hour)

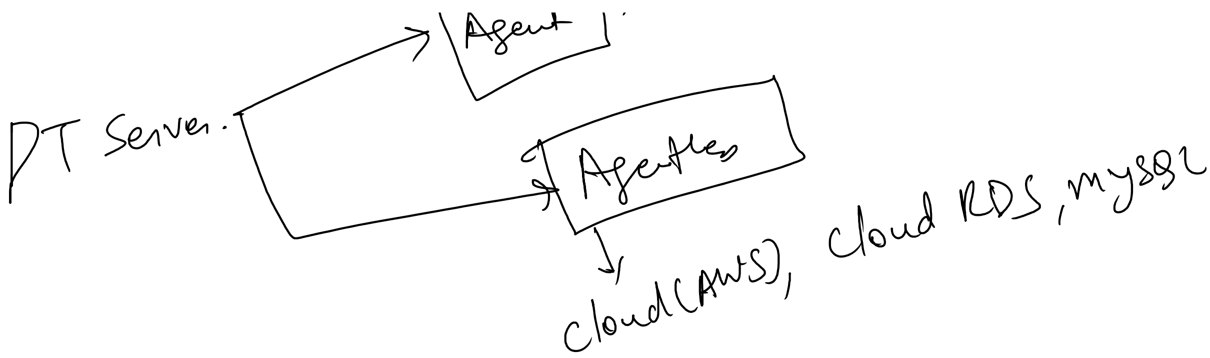
* cloud integration using AWS:-

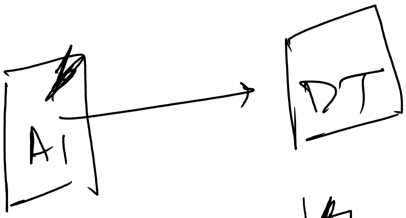

Policy.
Role

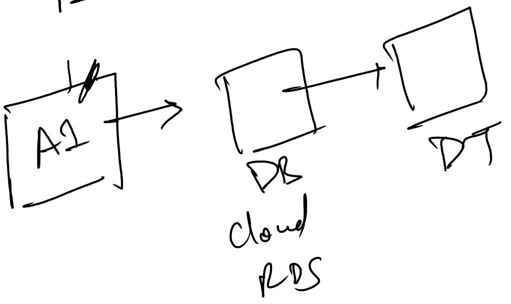

Select * a
s-sec.
Cloud RDS | MySQL
DB
DT
a*
Select * a d

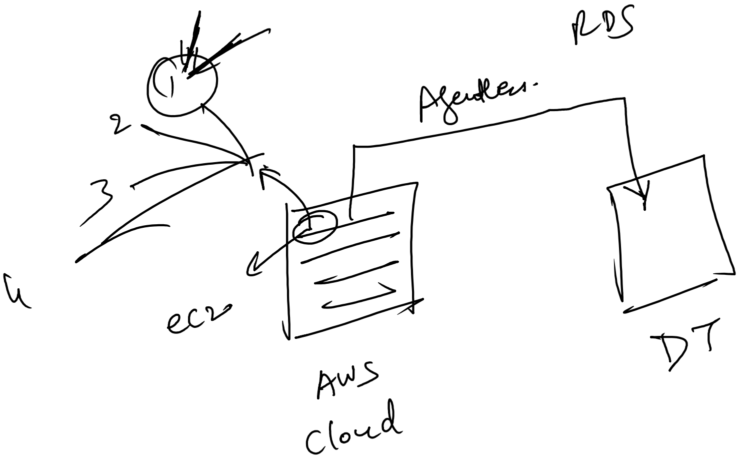

Splunk
DB
DT


Agent
easyTravel,







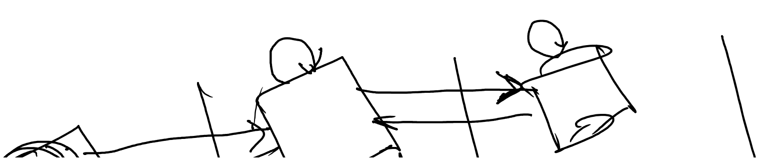


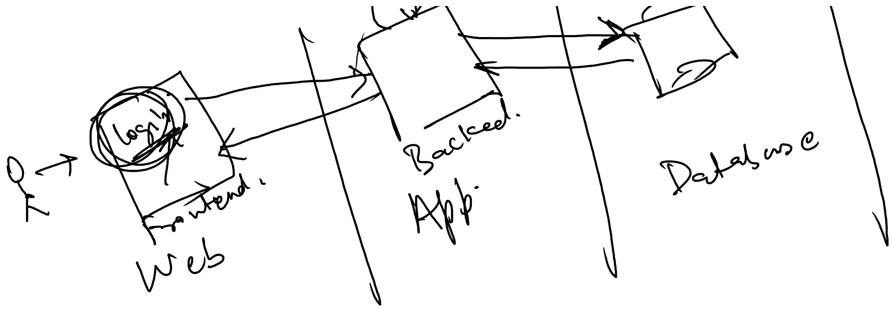
Container Monitoring.

Docker engine.

Image.

Containers.




login
Frontend.
Web
Backeed.
App.
Databuse