

FSI315 : JPMorganChase: Real-time fraud screening at massive scale ❤️

Wednesday, December 04, 04:30 PM PST, 60 min

Bernie Herdan



Location

Wynn | Upper Convention Promenade | Cristal 7

"Fraud Int" Platform
Scope:

Building a fraud decisioning platform

- Real-time ingestion of payments
- Analyze payment data to detect potential fraud
- Auto-scalable to process high volumes
- Low-latency, highly available, resilient application
- Capability to have offline analysis of data

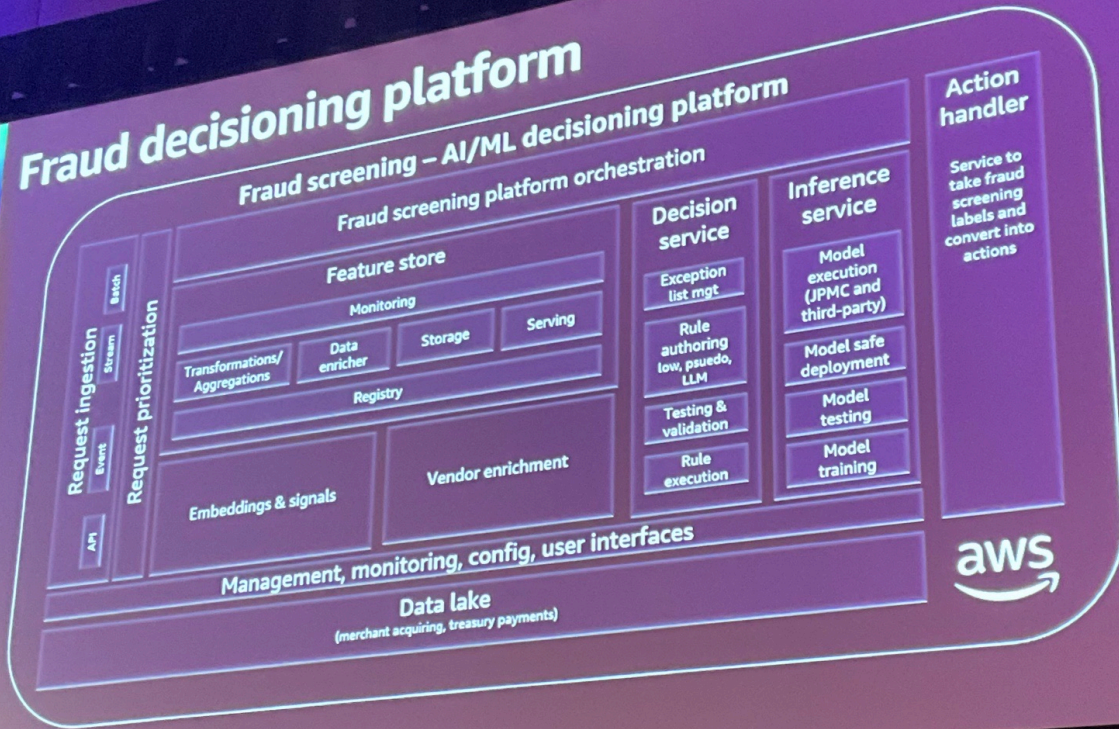
Quick new features



© 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

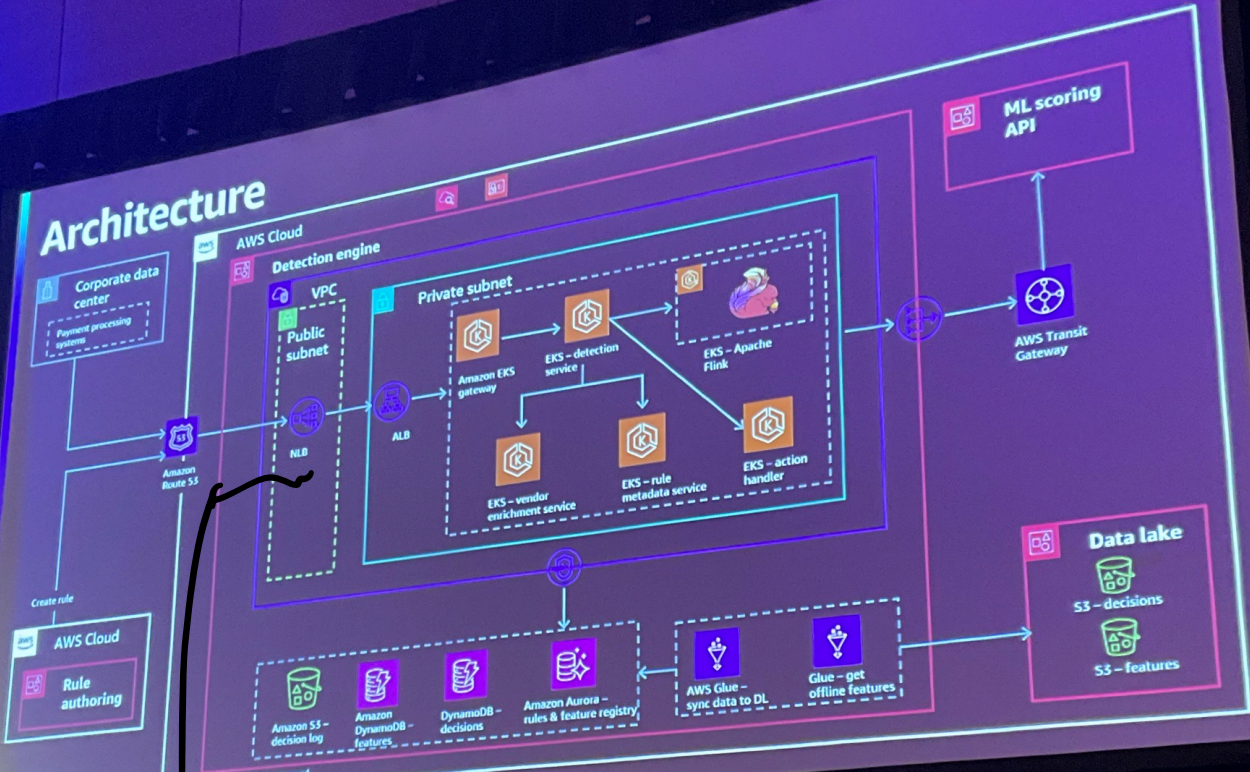
Fraud decisioning platform

Fraud screening – AI/ML decisioning platform



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture

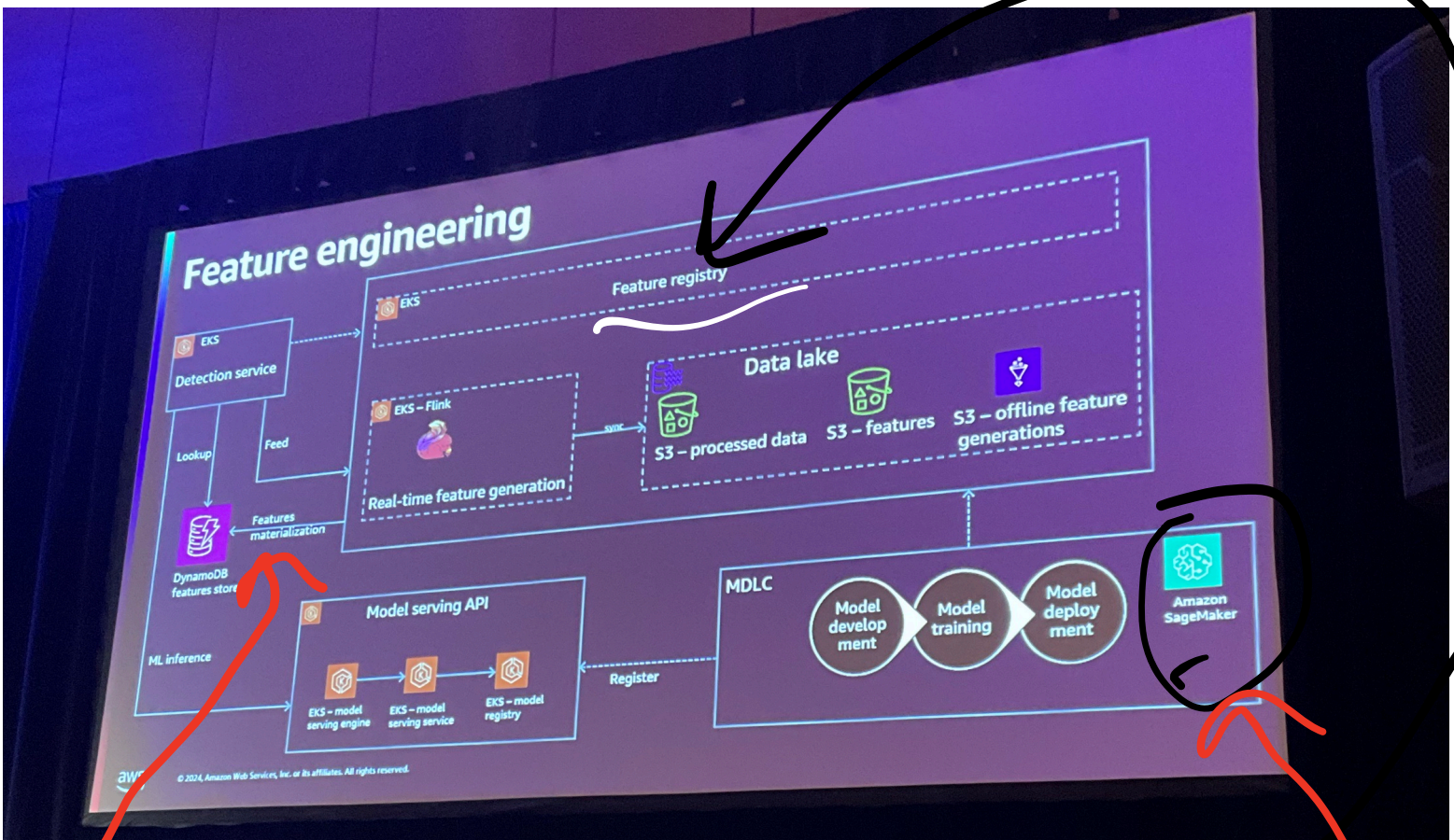


© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

why - 101? aavda for rule duty

S3 audit User Interface
for New Rules

Glue Jobs for ETL for data
lake



Containers all with one job

Flink for real time feature generation

SageMaker → AI vs Redrock

Dynamo for "features stack" with

Build Container with final

"feature" for processing

JP Morgan uses EK5

10s millions 400/Req/Sec
day

AWS Load Bal Controller

* Plugin - eliminates Rev Proxy

VPC and points for cross region

86% cross region Amazon RDS

Large Locks DB
save big tables more capacity

Based on CPU/Mem / SQS
Scaling

~~Scale Early~~

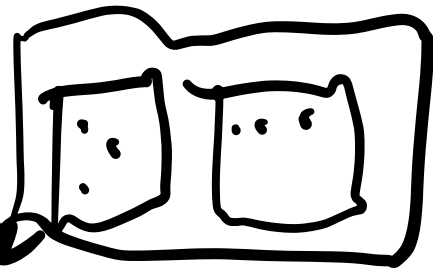
50% CPU + 50% Mem
Start Scale

Many Layers of Redundancy

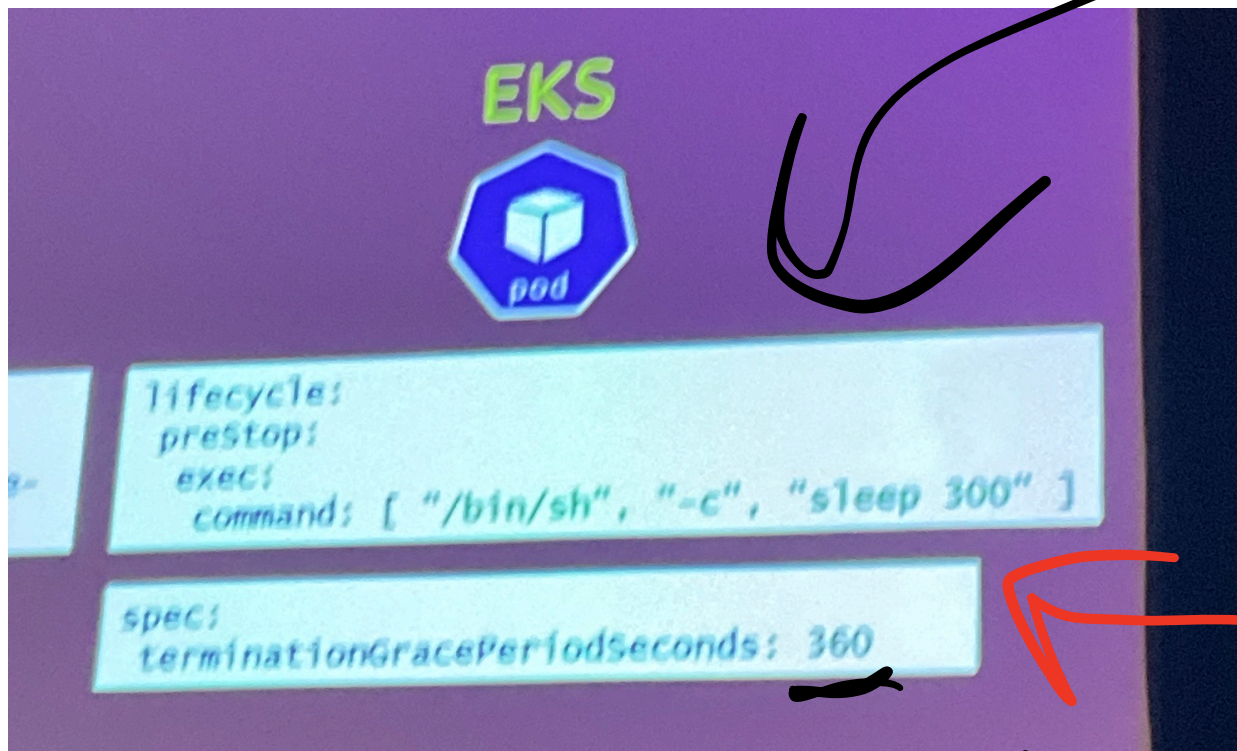
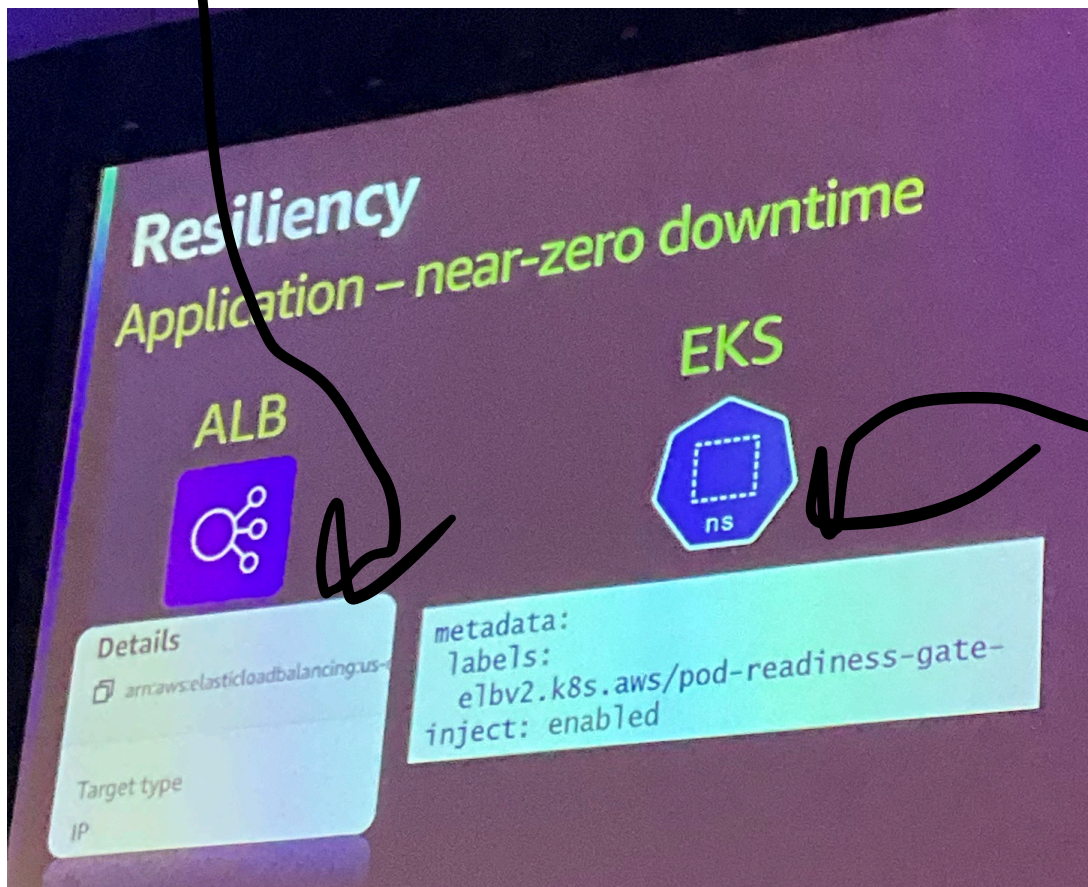
ALB
Type IP

Important 2x
Because
Many Pods
Could be on
one Instance

6 Pods



4/4



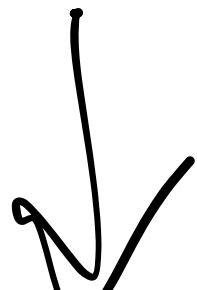
high vol 30 sec default
for shutdown was not enough time

"Pre Stop Sleep hook"
to give alb time to take
it out Not Blocking

in region resiliency

A2s sometimes get very
uneven in pod deployment
anti affinity

Topology spread
constraints



Resiliency

In-Region resiliency

"Anti-"
affinity

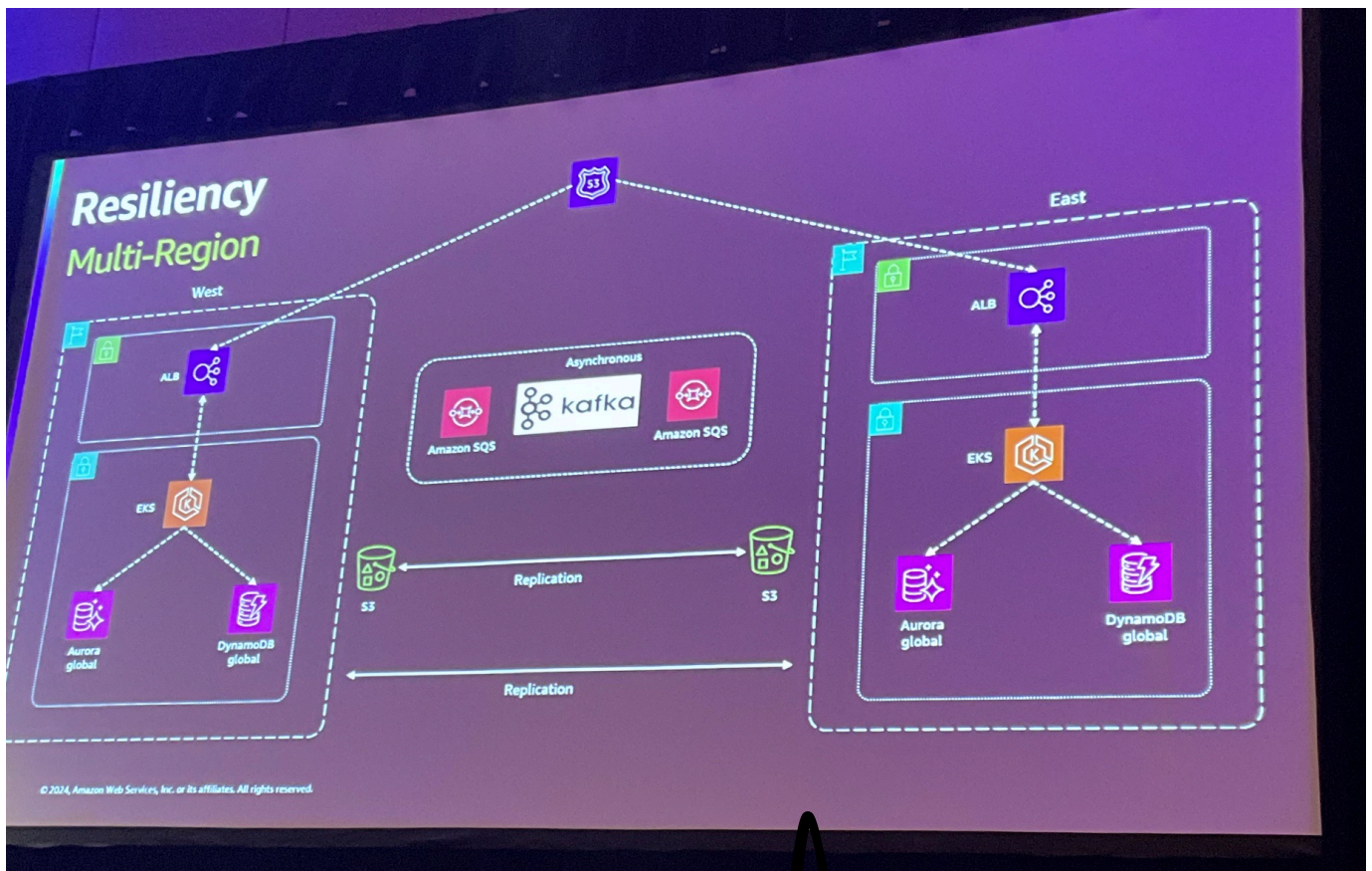
```
affinity:
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
      - podAffinityTerm:
          labelSelector:
            matchExpressions:
              - key: app
                operator: In
                values:
                  - app
          topologyKey: topology.kubernetes.io/zone
          weight: 100
```


They are ready for full region failure

Topology spread constraints

topologySpreadConstraints:

- maxSkew: 1
topologyKey: topology.kubernetes.io/zone
whenUnsatisfiable: ScheduleAnyway
labelSelector:
matchLabels:
app: app
- maxSkew: 1
topologyKey: kubernetes.io/hostname
whenUnsatisfiable: DoNotSchedule
labelSelector:
matchLabels:
app: app



KAFKA

Separation of work

Streams

CC fastest Account

ACH

Wire

etc

Separate
Accounts

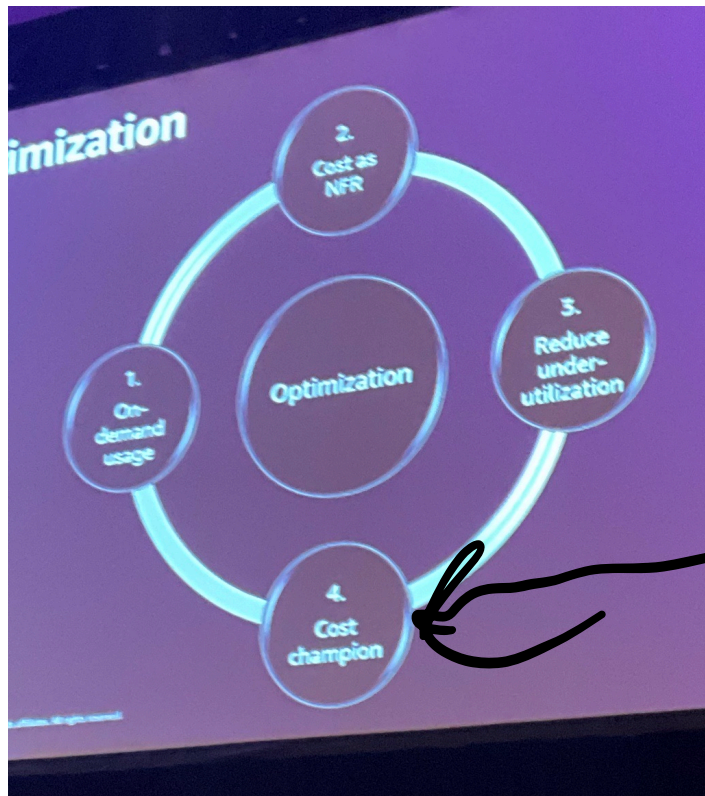
Logical separation of services

"Cost optimization has to be a day 2 activity"

* devs can start, but auto shut down

Cost as

Non Functional Requirement "NFR"



Whole, Total
Savings
\$