ORACLE

Kubernetes
The Good, The Bad, The Ugly

Karthik Gaekwad
Developer Advocate, Oracle Cloud
@iteration1

ORACLE

Registry

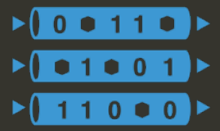
Container Engine for K8s

Functions

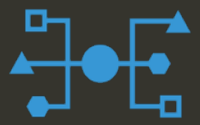
Notifications

Monitoring

Cloud Native @Oracle



Streaming



Events

Logging

API Gateway

Resource Manager

ORACLE

Registry

Container Engine for K8s

Functions

Cloud Native @Oracle

Behind the Scenes…

Top 5 learnings

- I used to be a developer on Oracle Container for Kubernetes (OKE).

“OKE: One of the most popular platforms on Oracle Cloud”

OKE Popularity

- Used by:
  - External customers
  - Teams leveraging platform for internal work
  - Teams leveraging platform to build platforms on Oracle Cloud

Architect your teams

- Started with single “full stack” team.
- All members of the team could work on any part of the infrastructure.
- Lots of learning to build V1.

Architect your teams

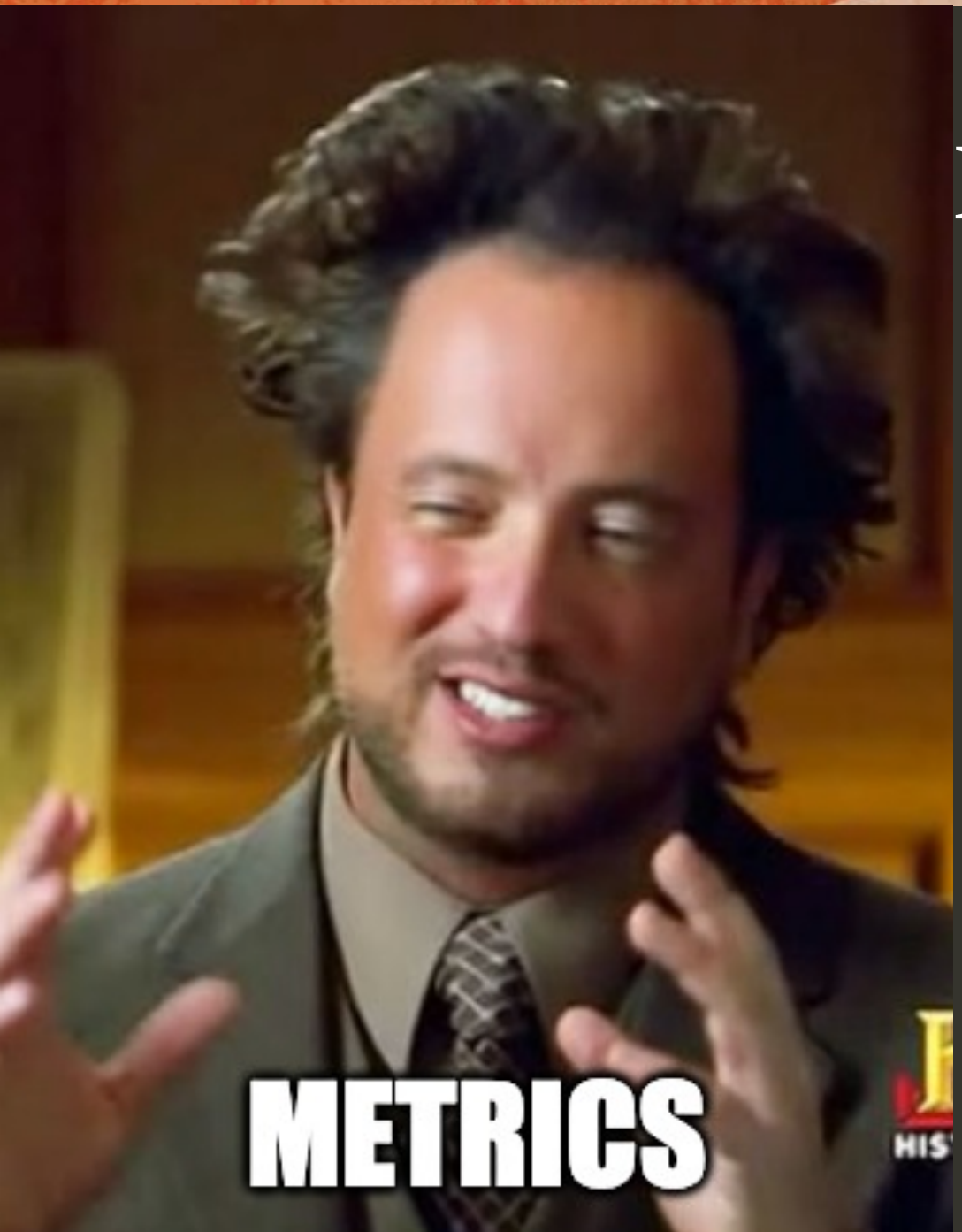
BUT….

- Kubernetes landscape is vast.
- Difficult to continuously keep up to date on all infrastructure.
- Also, individual developers gained deep expertise in different areas.

Team reworked

- Split single team into 3 teams:
  - Control Plane (OKE management)
  - Data Plane (Customer cluster creation and management)
  - Platform (UI/Oracle specific/ devops concerns)


METRICS
HISTORY.COM
imgflip.com



Metrics

- "Collect first and analyze later strategy"
- Faced issues with collection and retention at scale.
- Hard to understand what we were actually collecting, and whether it was correct.

Metrics Reworked

- **Step 1: Spend more time understanding what is important to the service.**
- Step 2: Verify that you are collecting metrics that are important to you from above step.
- Step 3: Collect other metrics, but lower the resolution.

- End result:
  - Utilized metrics to our advantage to debug and fix issues.
  - Dashboards were useful again!

Burnout

- Hard problem: Managing large popular complex platforms

- Teams struggle if they can’t balance time between fixing issues and firefighting.
- Constant firefighting without fixing underlying issues can lead to apathy, disillusionment and burnout.

Burnout

- Empower team to fix issues.
- Larger team to share operational burden.
  - (Grew from 10->40+)
- Prioritize bug fixes and features equally.
- Rotate feature team to be on call.

Pick the right platform

“Kubernetes for x” is a common question…

Pick the right platform

- Kubernetes is a great generic platform that you can use to build other platforms on.
- There might be other native services that already fit.
- IaC/Serverless/Native services are also viable choices.

Summarize

- Building a popular platform is exciting!

- Empower your Product and SRE teams.
- Watch for burnout.
- Have a plan for what metrics to collect.
- Pick the right platform- between IaC, Kubernetes (Managed or otherwise), Serverless.

Come Chat More > Booth 201

Session @11:35 today in M1

The cloud native systems analyst

Come visit the Cloud Native showcase @ Oracle Booth 201

Daily Raffles during the Afternoon break

Headphones and a BB8 Droid

Lightning talks during the breaks

Thank you

Karthik Gaekwad
@iteration1