Serverless....

Going Serverless definitly gave back control to

(the IT department of) the customer.

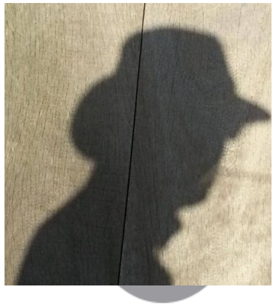
But “cloud” and “serverless” did not come without some challenges.

The software-technology in use is Java, Node.JS, some Python, and a good dose of PL/SQL.

The deployment in the new situation is either via Containers or Lambdas, and the databases are all DBaaS (actually AWS-RDS, and still mostly Oracle).

The overall experience has been very Positive and has given us a few lessons about our legacy-systems, software-tools and the data, in the process.

- Road Trip
- Rovinj



Piet de Visser
PDVBV

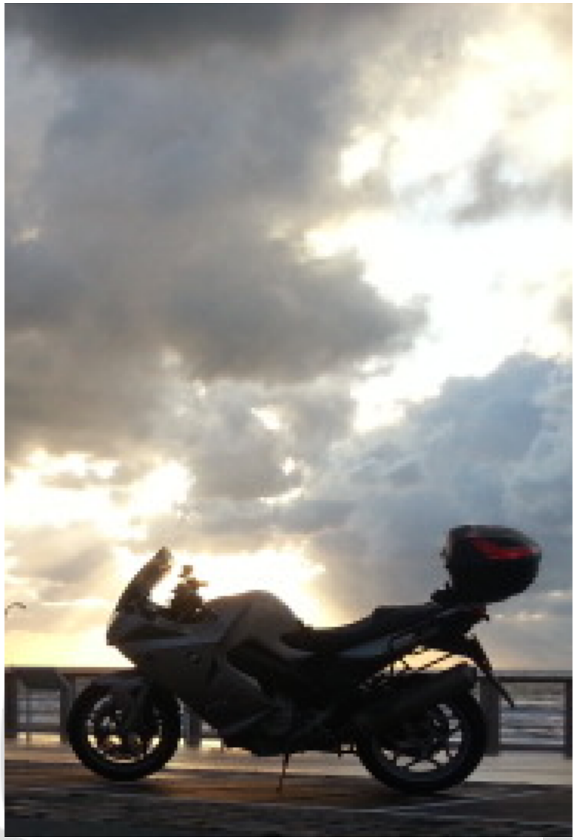
PGDAY'17 RUSSIA

КОНФЕРЕНЦИЯ ПО БАЗАМ ДАННЫХ

Serverless…

Really?

PDVBV – The Simple (oracle) DBA



Favorite Quotes: “The Limitation shows the master” (Goethe), “Simplicity is not a luxury, it is a necessity. Unfortunately, “Complex’ solutions sell better. (EW Dijkstra).

Logo Cloud

See you all in Wroclaw.

- Portbase
- (dutch gov)
- Shell
- Philips
- ING bank
- Nokia
- Insinger, BNP
- Etihad
- NHS
- BT
- Claritas, Niels
- Unilever
- Exxon
- GE

Don’t waste time on Self-Inflation… but Hey, this was such a cool Idea (from a marketing guy)…
Logos of my major customers over time. If you want your logo here: Hire me.

What does it look like..

Pinne

ann

Has #Brexit?

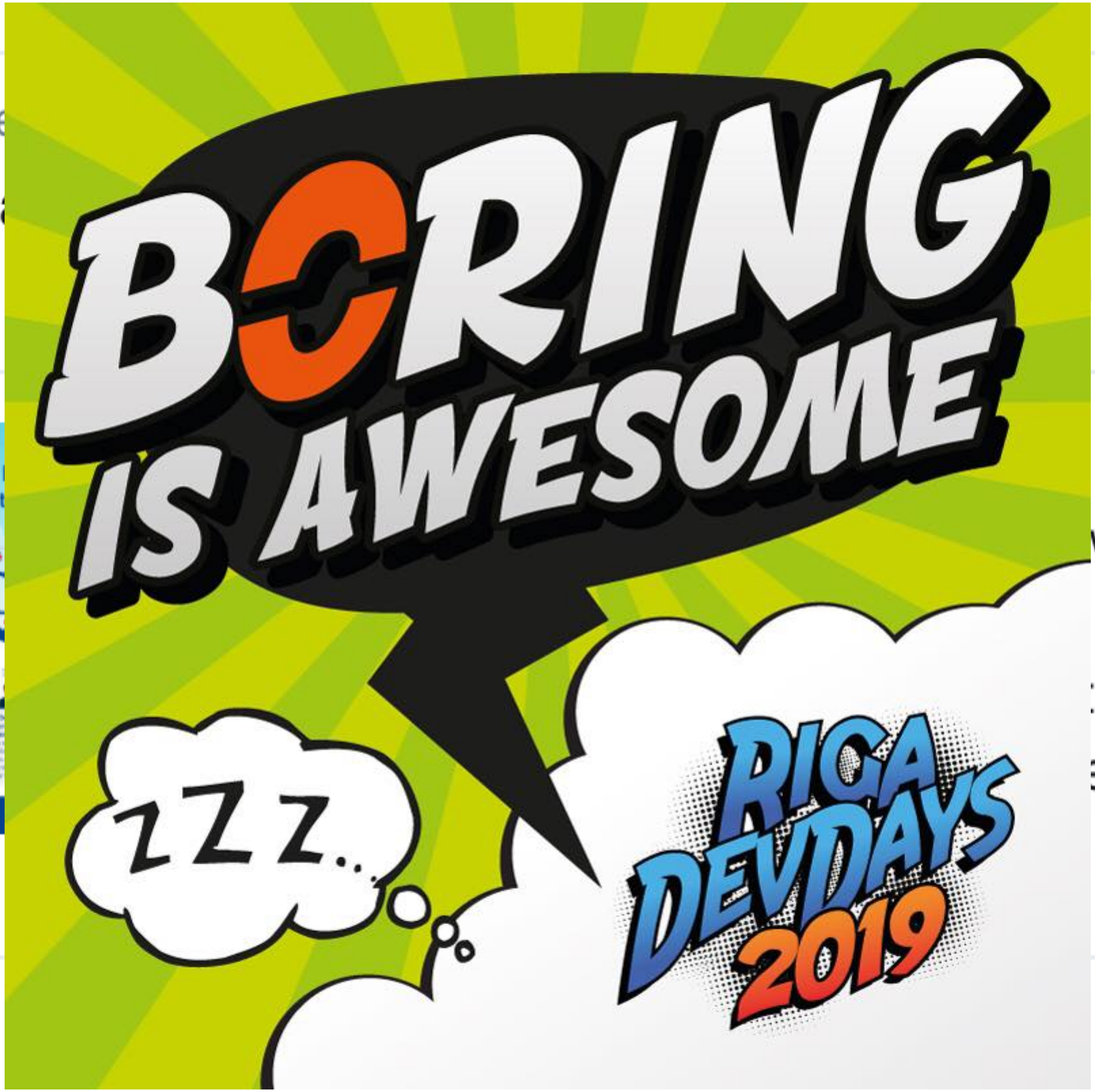
y for

port

ve easy steps for

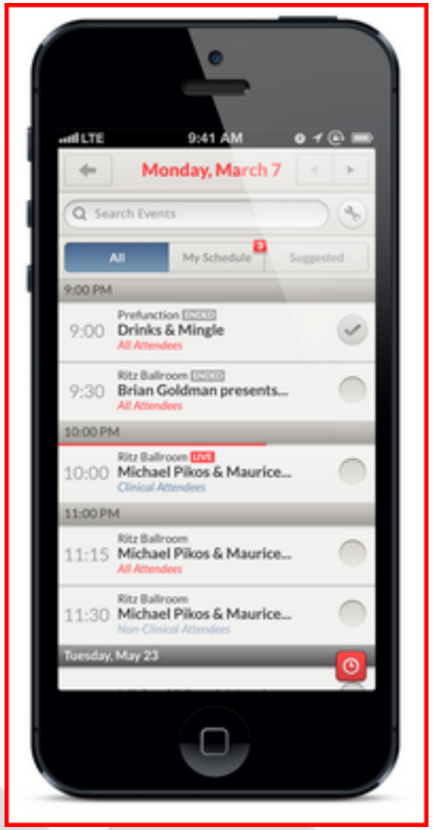
h English

e UK? #Brexit



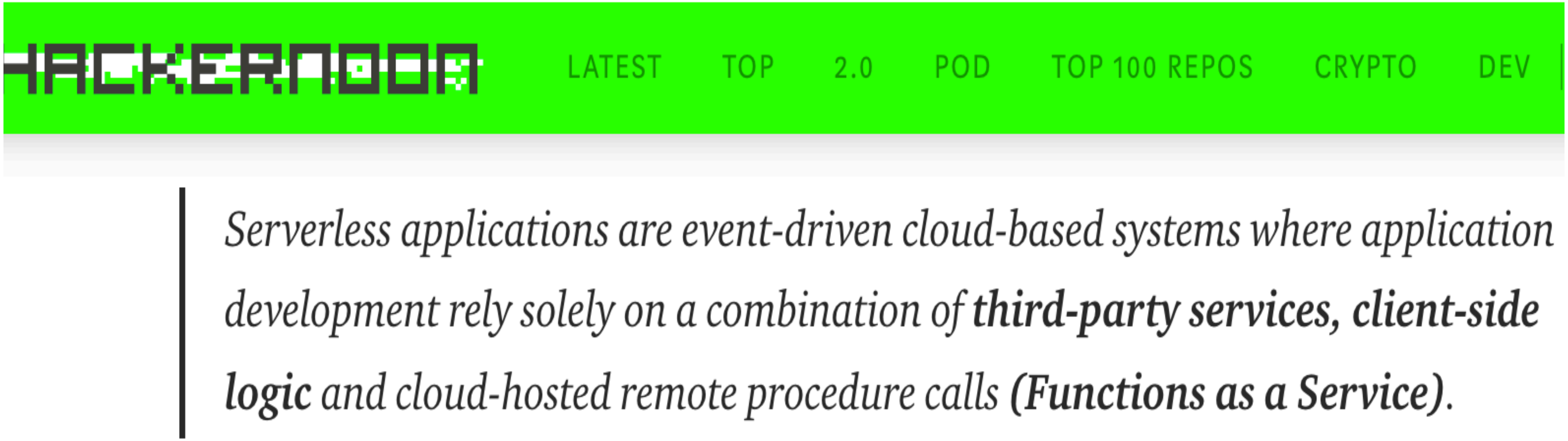
Agenda (approx 45 min)

History	(2 cases…)
Define Server(less)	(cheap shot…)
Why Serverless	(motivations)
A few topics..	(Quiz Zzz !!)
Lessons	(FWIW)
Discussion	(Do Challenge!)



Agenda. Why, what, how, Illustrations, comparisons.
30+ slides, 1min/slide.

What does WWW think…



> *Serverless applications are event-driven cloud-based systems where application development rely solely on a combination of* ***third-party services, client-side logic*** *and cloud-hosted remote procedure calls* ***(Functions as a Service)****.*

• Event Driven?	Hmmm…
• Cloud Based, 3rd party..	Yeah..
• Client-side-logic ?	Hmmm…
• FaaS ?	Hmmm…

Definition. And keywords: Lambda, Cloud Funcitons, Azure functions, Openwhisk
All exmaples seem vendor-bound… not sure if C1 and C2 want that (but C1 deffo is AWS-locked)

Two Customers: Serverless

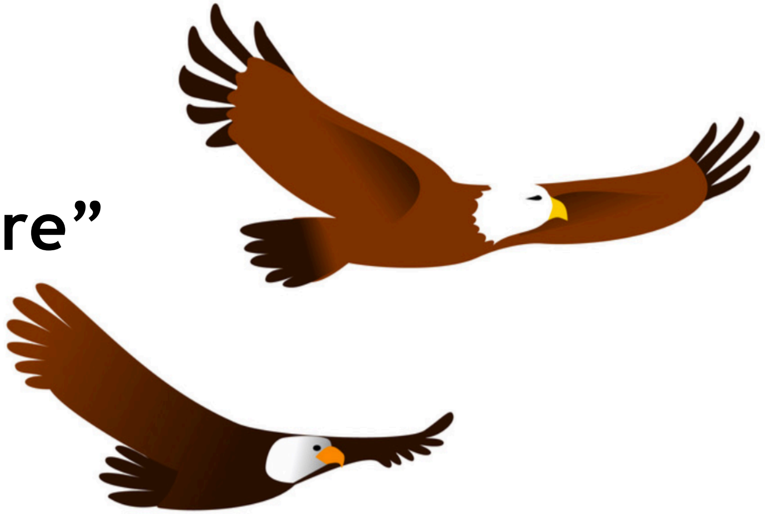
Why? …. : Freedom from “infra”, Less “admin”

- C1 : Amazon AWS. Docker. Lambdas. Some EC2.
- C2: K8s on “hardware in the basement”

Two cases, very different, but serverless. Same motivation, different Implementation. Both Success”
C1 = Buzzword driven, C2 = Goal Oriented. Pragmatic.

Commonalities…C1, C2

- Process Large number of B2B transactions (M/day)
- Variable workloads (day / week profiles)
- Changing environments, Adapt…
- Legacy Oracle (Forms ☺)
- Outsourced admin of “Infrastructure”



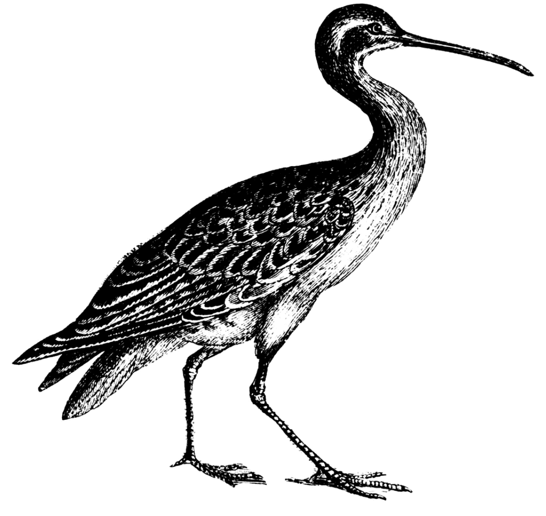
- Want To Get Rid of Oracle (!!!)
- Prevent lock-in to new vendors.
- Want to #Devops, use CI/CD Pipeline (Both Jenkins)

Processing is mostly OLTP, C1 has very strickt SLAs answer within 2sec, no dataloss.
Both now HATE Oracle. C1 also dislikes any “service provider”

Differences…C1, C2

- C1 : Guru + Buzzword driven.
  - DevOps (agile, git, Kanban, Netflix)
  - Cloud-Native, Serverless. Microservices.
  - Mistrust and ”Squeeze” providers
  - The killer: ”They're not DevOps Enough”

- C2 : JFDI, Just Make it Work.
  - Process focussed. Must Work Correctly.
  - Test-culture: Test Everyting (Automatic!)
  - Long term partnership with “infra provider”
  - Wants data “in house” (!)

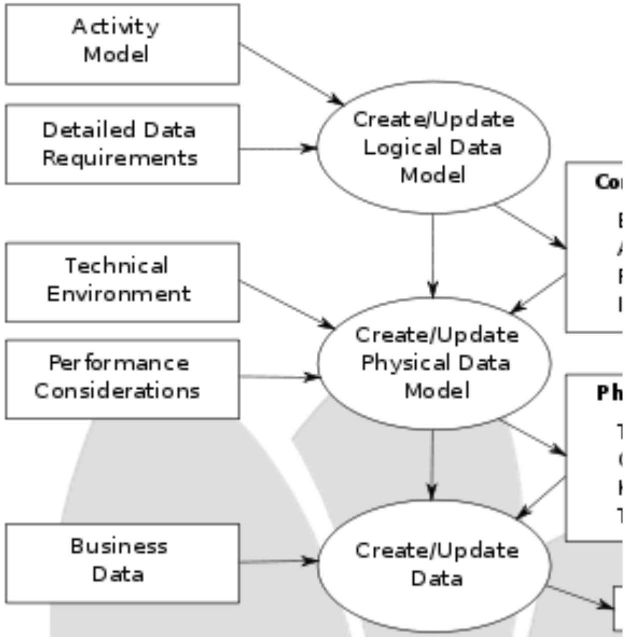


Two cases, very different, but serverless. Same motivation, different Implementation. Both Success”
C1 = Buzzword driven “You are Autonomous…”, C2 = Goal Oriented. Pragmatic.

What happened…

- C1 : Pioneering, Innovation!….
  - Vision: DevOps, Agile, Autonomous, AWS
  - Spends a lot of time in TerraForm
  - Lots of “discovering legacy”
  - (Brexit was good Learning Experience…)

- C2 : Pragmatic, Knowledgable.
  - Vision: K8s as platform - (next: Jetson?)
  - Close involvement of “infra provider”
  - Hardware on-site, on-demand, Flexible.
  - Smooth Database-clone process (PDBs)
  - Test, Test, Test + Deploy (automatic,..)



Activity Model

Detailed Data Requirements

Create/Update Logical Data Model

Co

Technical Environment

Performance Considerations

Create/Update Physical Data Model

Ph

Business Data

Create/Update Data

The pioniers : the one with arrows in their back.
C1 = Buzzword driven, incl Chaos Friday/month, C2 = Goal Oriented. Pragmatic.

C1: Low Freq. "oversight" Deployments..

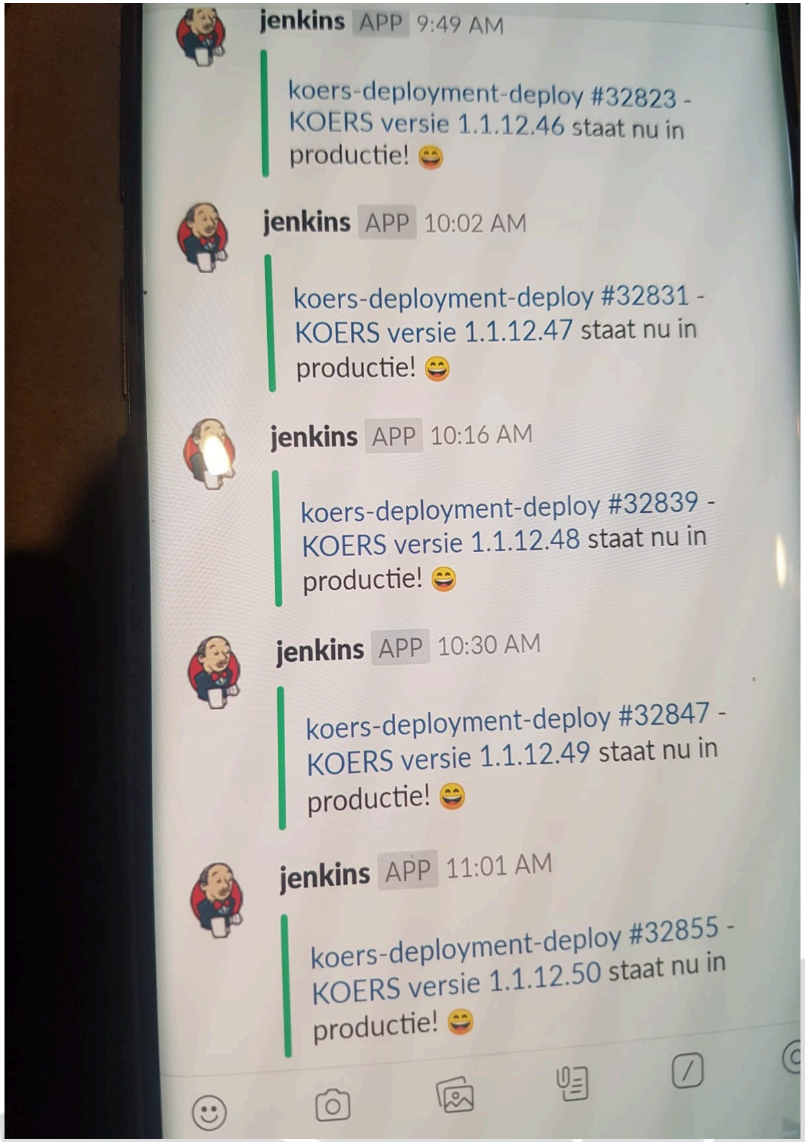
- <2 deployments / week
- Push-button Deployment (who …?)
- 4hrs prepare, 1hr Deploy @19:00
  - No downtime….normally.
- IMHO… No Serious Testing.
- "too complex" - Really ?
- Reality: Test != Sexy.

stock image

The "careful" deployment. Main reasons are : Complexity and no-automated-testing.
The 2/week "sweathy hands deployments", with occasional "fail + rollback"

C2: High Freq (CI/CD) Deployment..

- 10 deployments / day
- Automated-Everything.
- Test !!
- Setting up Tests = Dev-effort.
- 150 Test-databases, 1000+ cases
- (a lot of effort, ~ Boring!)
- “Trust” in CI/CD Pipeline…



jenkins APP 9:49 AM
koers-deployment-deploy #32823 - KOERS versie 1.1.12.46 staat nu in productie! 😄

jenkins APP 10:02 AM
koers-deployment-deploy #32831 - KOERS versie 1.1.12.47 staat nu in productie! 😄

jenkins APP 10:16 AM
koers-deployment-deploy #32839 - KOERS versie 1.1.12.48 staat nu in productie! 😄

jenkins APP 10:30 AM
koers-deployment-deploy #32847 - KOERS versie 1.1.12.49 staat nu in productie! 😄

jenkins APP 11:01 AM
koers-deployment-deploy #32855 - KOERS versie 1.1.12.50 staat nu in productie! 😄

The secret to success: Testing. No Excuses to create + maintain Test-cases..
Good, Automated testing => trust in Pipeline. Culture + Historty of Automatic-Testing.

Happy Topic #1 : DB-aaS ...

- The DB-aaS: Works!

- AWR-RDS: Very Happy!
- RDS - Needs Competition

- Managed DB in house…
- (HW + db-clones in house): Also Works

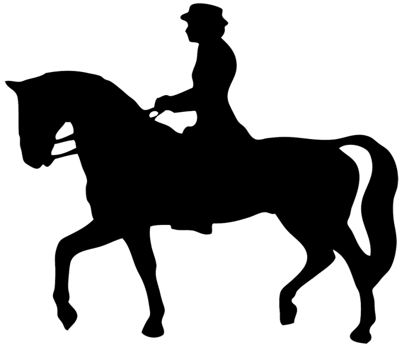
- Key point @ DB: Worries--, Productive++.

My Pet-Topic: Limit the DB-worries. Either AWS or the ”Service Provider” is taking care of all DB-stuff.
And the DBA / Data-expertise can “Contribute” instead of “hinder” progress.

Happy Topic #2 : Testing = Knowledge ...

- Did I mention…Testing !

- Create Test = Insight in your system
- Running Tests = Training.
- Automate Test = QA, Efficiency.
- Metrics = More Insight…



- Selenium or similar tool - Pipeline!
- And some DIY (you know your system best!)

- Slogan: Testing is Terrific…

BASETIDE

premium data solutions

Soso Happy: Thick Clients

- Both Containers and Lambda: Too Fat..
- Main Cause: Legacy “stacks”
- Notably: Oracle Client.
- “Workaround” => SQLcl
- Better: PSQL or PostgreSQL+Python

- Cold-starts too slow..
- Workaround: Provisioning.

- Net Effect: Container => small VM

- SINO: Serverless; In Name Only
  - Slow Everything Down…

Oracle Client largest obstacle, also: Oracle-bashing. Overweight Containers.. Slooowww.
One fix: use SQLcl to replace SQL*Plus

Soso Happy: Conn Pools / container

- App in Container needs connecion-pool.
- Some apps take 10+ conn (and use 1??)

- Day 2 after go-live: DDOS by Conn-storm.
- Lots of “dead” conn, not re-used.

- Fixes…
  - Re-configure conn-pools (+/-)
  - Alert on Conn/container.
  - And: Cycle Containers…
  - Find+fix “dead” connections !

Containers are small VMs, but each container = overhead. Re-configuer conn pools
The Dead-conn was a test-message-bug, eating connections from pools.

Not Happy: Infra-as-code…

- Infra as code : Excellent Concept… but…
- Terraform (!= documentation)
- Possibly: We don't "master" TF yet.
- TF takes too much time+effort.
- TF (stored in Git) lead to Confusion + Mistakes.
- ACC deployed into PROD..
  - 2hr outage + lots of anger.

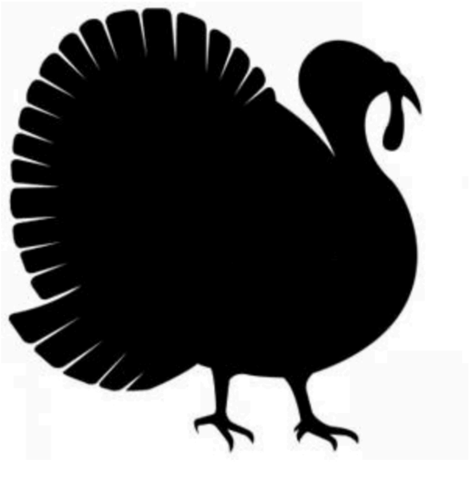
Terraform seems to take days+days of tine to get it right. Manual or "scripted" would be faster
One particular mistake caused 2 hrs un-necessary boutage. => needs more Work. (testing??)

Not Happy: Serverless-Fanatics

- Legacy requires a Virtual Machine (SINO…)

- Squeeze a VM into a Container ?
  - Legacy joke: win2003 --> a Container…??

- Trick: a VM is Not a Server…
  - VM can also be "provisioned", Agile ;-)



- If Container = small VM ?
- Then Use a VM…

- Ditto for too-fat Lambda: use a Container…
  - Don't Waste Effort on "impossible config"

If a VM is the "best" solution, dont hesitate to use a VM
Make life easier. A VM can be just as "agile" . A big Turkey is also a bird..

BASETIDE

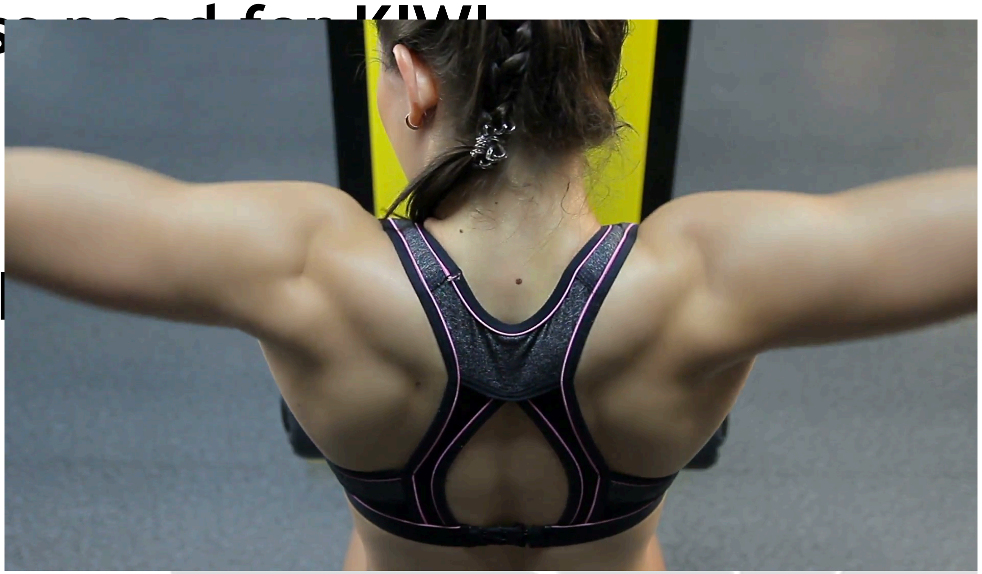
premium data solutions

Happy: Database, SmartDB.

- a DB is an Asset, not a problem: SmartDB
- Database can be ”provisioned”
- PL/SQL can be callable “service”
- Future: REST, ORDS. (=CRUD)

- Code is Easier to find+Fix in a DB
- Code Performs Better inside a DB.
- Overall more “efficient”, Less need for KIWI

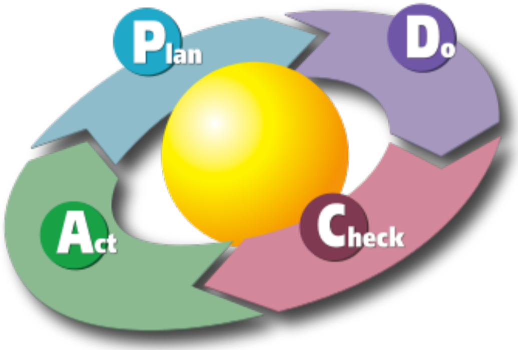
- Scale-Out: Not a real issue (a
- DB-cpu + storage are Cheap+
- (You are not Netflix…)



Quick plug for Smartdb, PL/SQL, PG/PSQL, GraalVM
Especially if a (micro)service needs State.. Consider putting it in the DB. (image: fat gorilla + athlete)

Lessons ?? (still early…) 1/2

- Automation - do I need to repeat this?



- Testing - not sexy, but … Important.
  - Testing needs Mindset, Design, Automation
  - Testing needs Work
  - Testing = Training + Learning + Metrics…

- Don’t be “Fanatic” on Serverless.
  - Accept that “legacy” can be ”tolerated” (VMs!)
  - Bi-Modal…

- Focus on Goals (not on Buzzwords)
  1. Do the Job(s) of the system.
  2. Be Flexible, be Automated, be QA-Focussed.

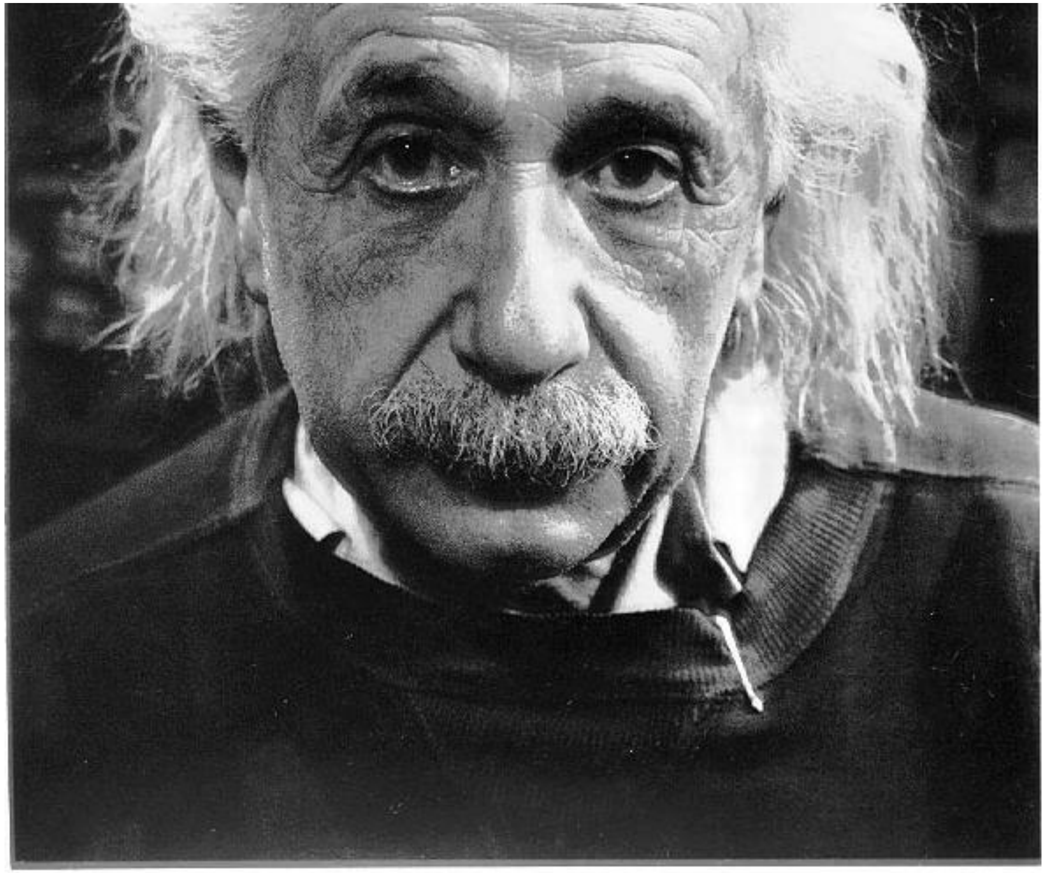
PDCA cycle.,.. N otably testing.
And “Latency = next challange”, you cannot buy speed of light…

Positive! 2/2

- Both Customers are Happy, Flying DevOps
- Both spend less time on ”admin” (and lower Co$$t)
- Both “feel” relatively Flexible.
- End Result = Positive.

Positive in both cases.
And the innovative. C1 is learning….

He got it …

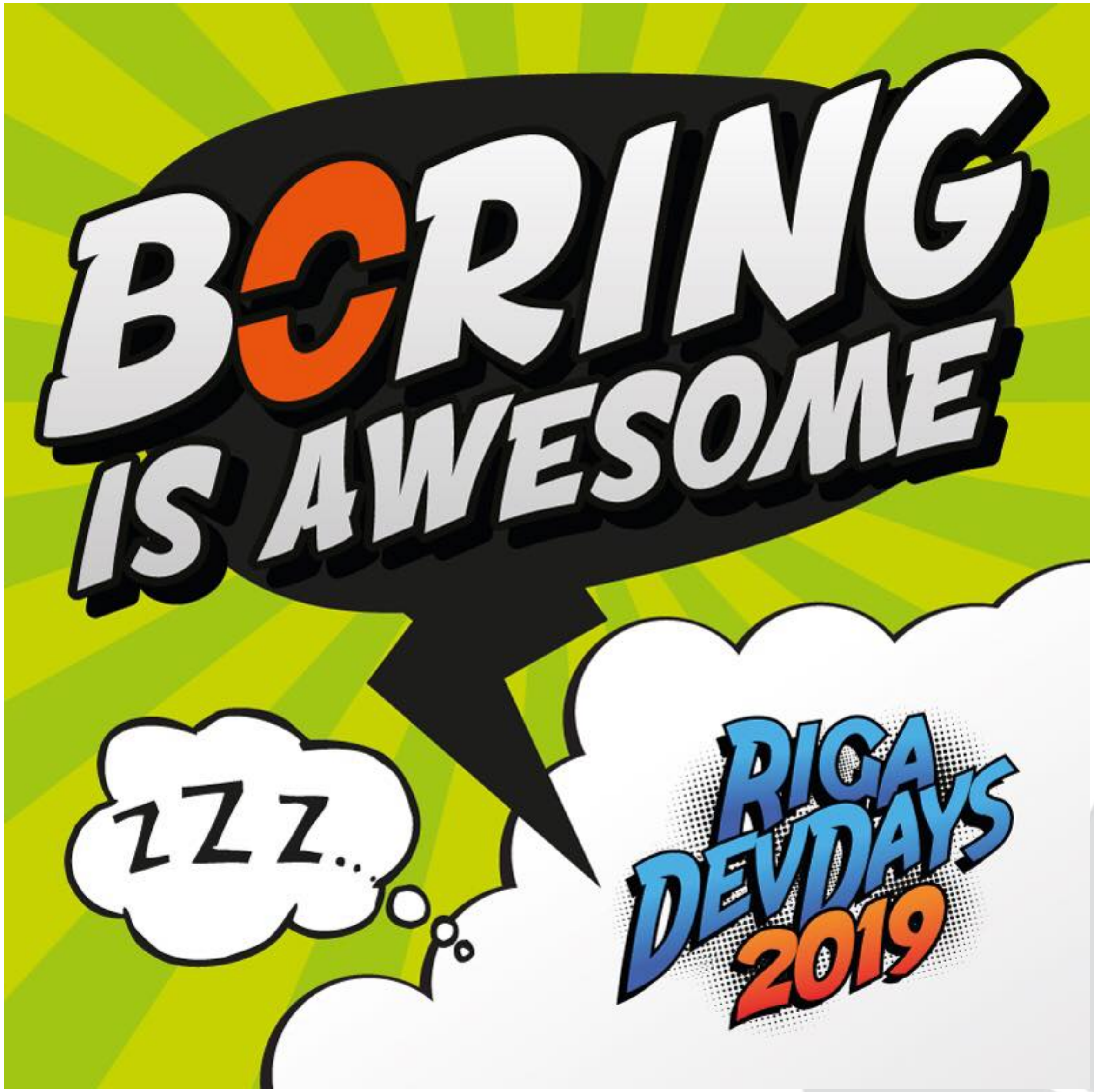


“If you can't explain it simply, you don't understand it well enough”

As Simple as Possible, but not too simple
Simplicity is a Requirement – but Comlexity just sells better (EWD).

Remember The Boss…

-



When you go home, and tweak your systems..
Boring is Always Best (who seen the movie) – and if Testing is "boring".. It is still needed.

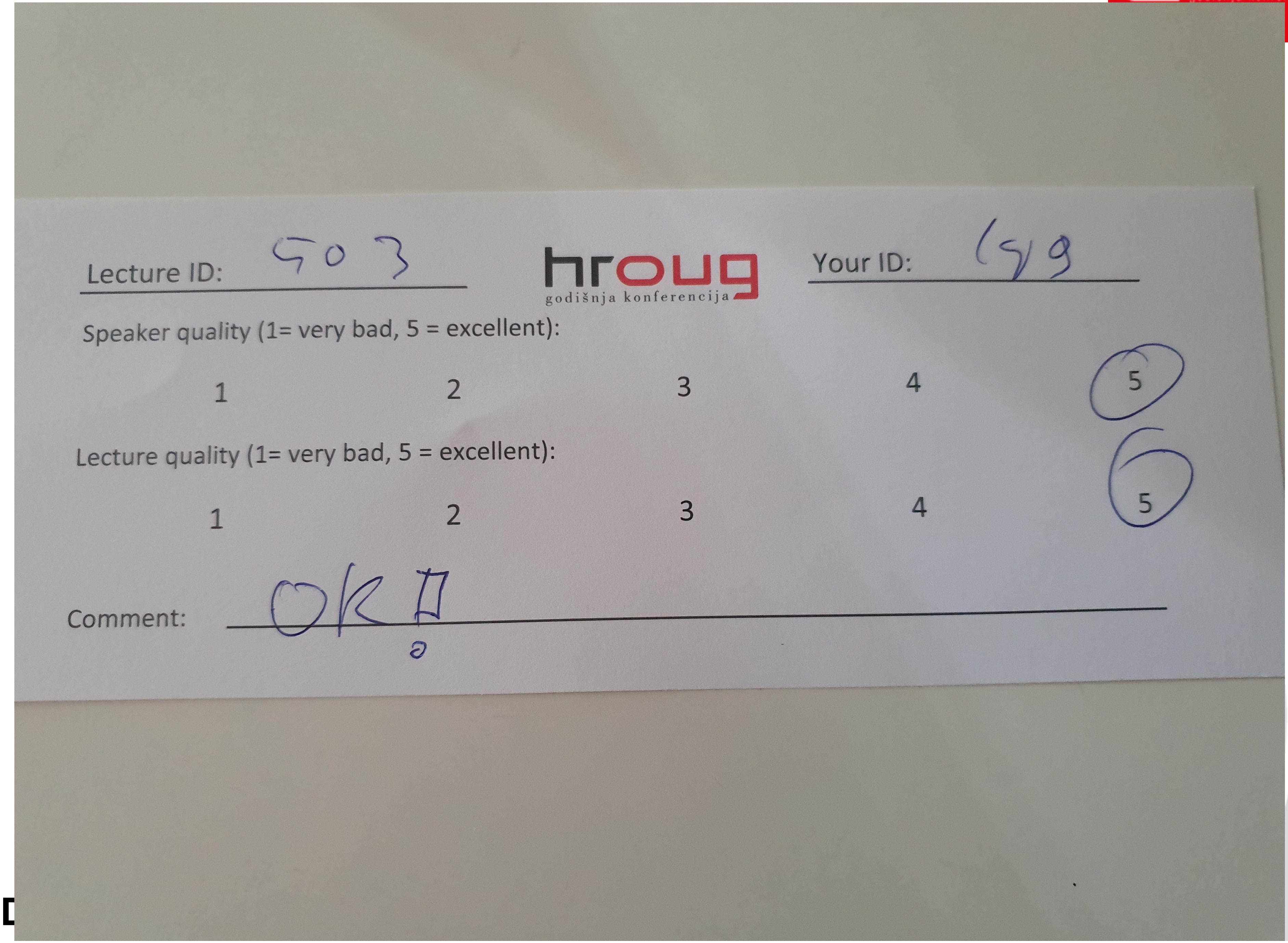
Quick Q & A (3 min ;-) 3 .. 2 .. 1 .. Zero

- Questions ?
- Reactions ?
- Experiences from the audience ?

PDVBV

Question and Answer time. Discussion welcome
Teach me something: Tell me where you do NOT AGREE.

(what about that Razor?)



Lecture ID: 503

Your ID: 159

Speaker quality (1= very bad, 5 = excellent):

1 2 3 4 [✓ 5]

Lecture quality (1= very bad, 5 = excellent):

1 2 3 4 [✓ 5]

Comment: OK

PD

Smart DB – Pink DB

#SmartDB #PinkDB

- Don’t Drag Data Around…
- Data-Gravity

#BringCodeToData

Chemtrails are clouds …

…