

METRO|NOM

**Use Docker, Go, Rest and
some more to build a
PostgreSQL
multi cloud self service**

PGCONF US, NEW YORK

MARCH 2020

About me

Heiko Onnebrink - DBA – developer - open-source contributor

300 relatives in China + 3 kids in Germany

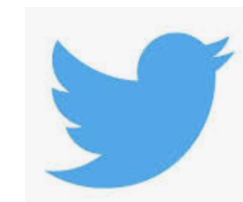
working 25 year as Oracle employee, trainer, consultant & freelancer

I decided 3 years ago to stop doing anything with Oracle products

Currently principal domain architect of core infrastructure @ METRO | NOM



heiko.onnebrink@metronom.com



@HeikoOnnebrink



THE BIGGEST SOFTWARE COMPANY YOU NEVER HEARD ABOUT

METRONOM – SETTING THE PACE IN FOOD AND TECHNOLOGY



METRO | NOM

150.000 Employees

€37 Billion

25 Countries

2000 Employees

IT-Services

IT-Solutions

Relational Database @ METRO | NOM

Facts

- **IT grown over decades**
- **3000+ Oracle databases**
- **ticket driven organisation**
- **slow delivery time**
- **low level of automatisisation**

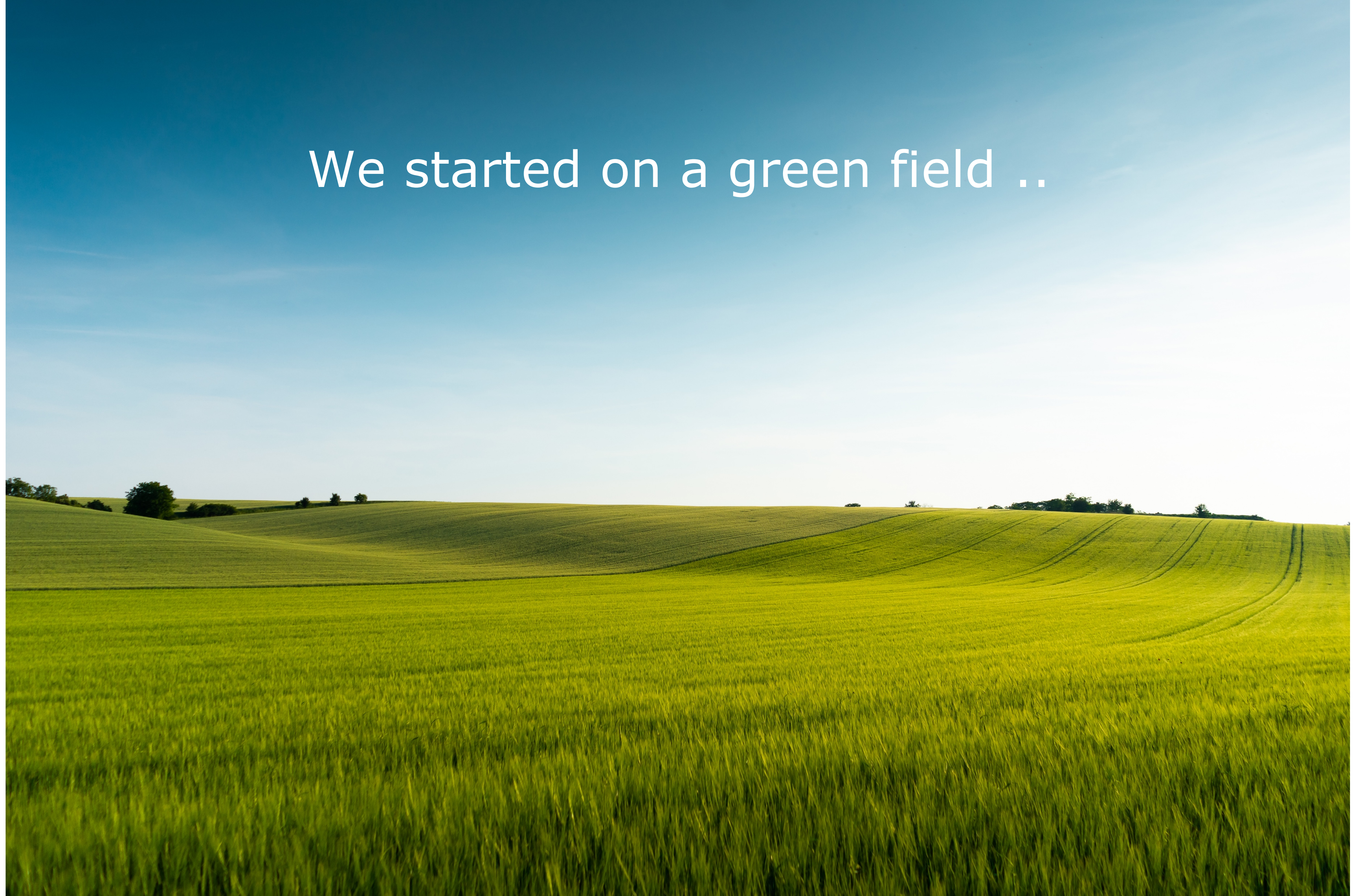
Challenges

- **Digital Readiness projects**
- **alternative RDBMS**
- **support DevOps / DBaaS**
- **full automation**
- **self service**
- **run across any cloud, on prem and public**

A pair of hands holds a white rectangular sign against a plain white background. The sign features the text "TIME FOR SOMETHING NEW!" in a bold, dark red, sans-serif font, arranged in three lines. The hands are positioned at the left and right edges of the sign, with fingers gripping the paper. The lighting is bright and even, highlighting the texture of the skin and the paper.

**TIME FOR
SOMETHING
NEW!**

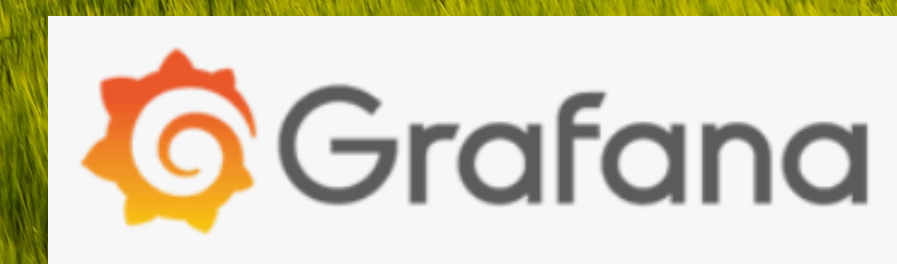
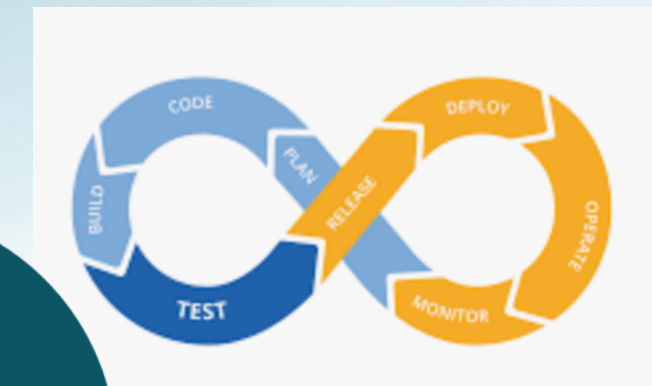
We started on a green field ..



.. and looked around what's out there



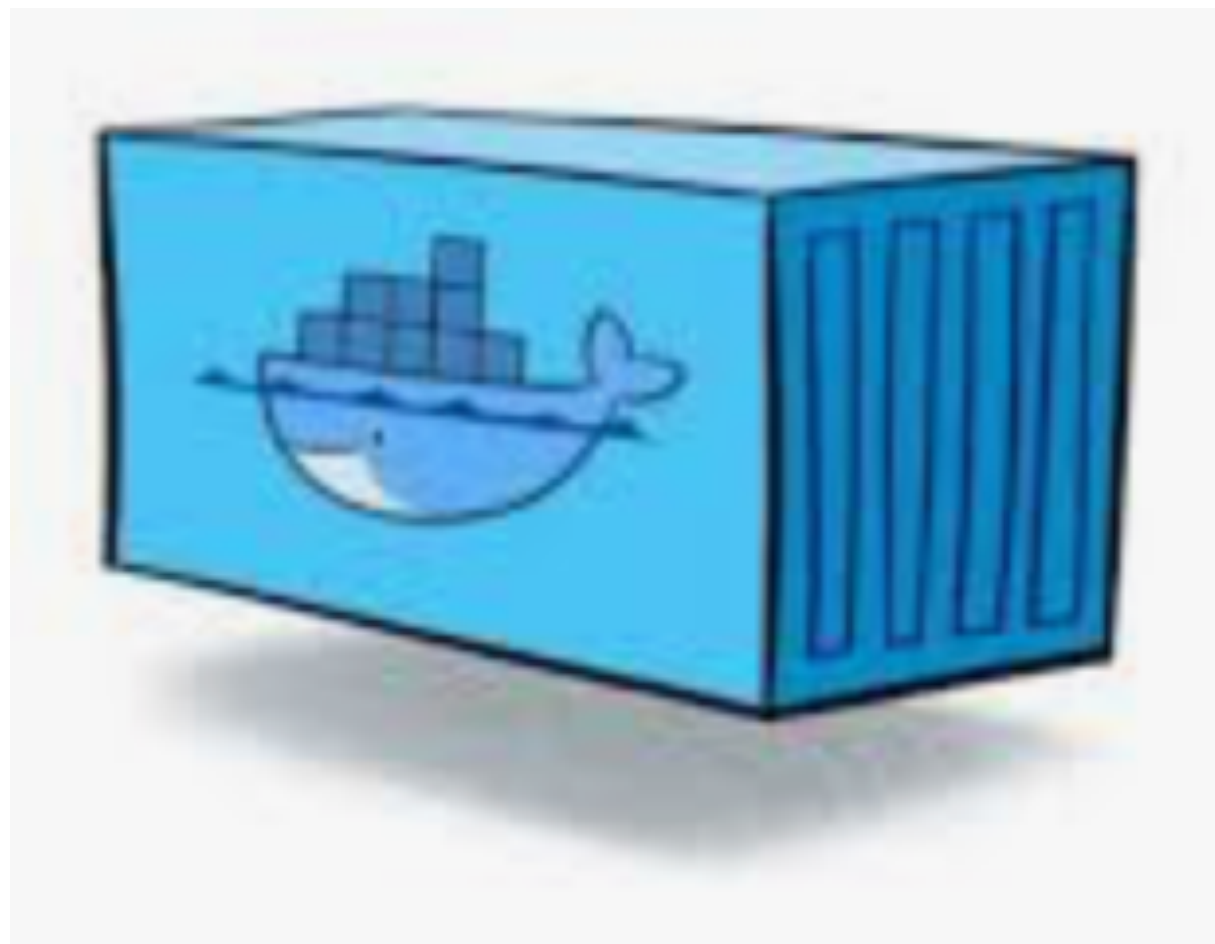
.. and looked around what's out there



CHAPTER # 1

Containers

We started and build some containers ..



The PostgreSQL container

Based on Ubuntu Image

PostgreSQL database with GIS

WAL-G for wal shipping

FDW for PostgreSQL and Oracle

Oracle instant client

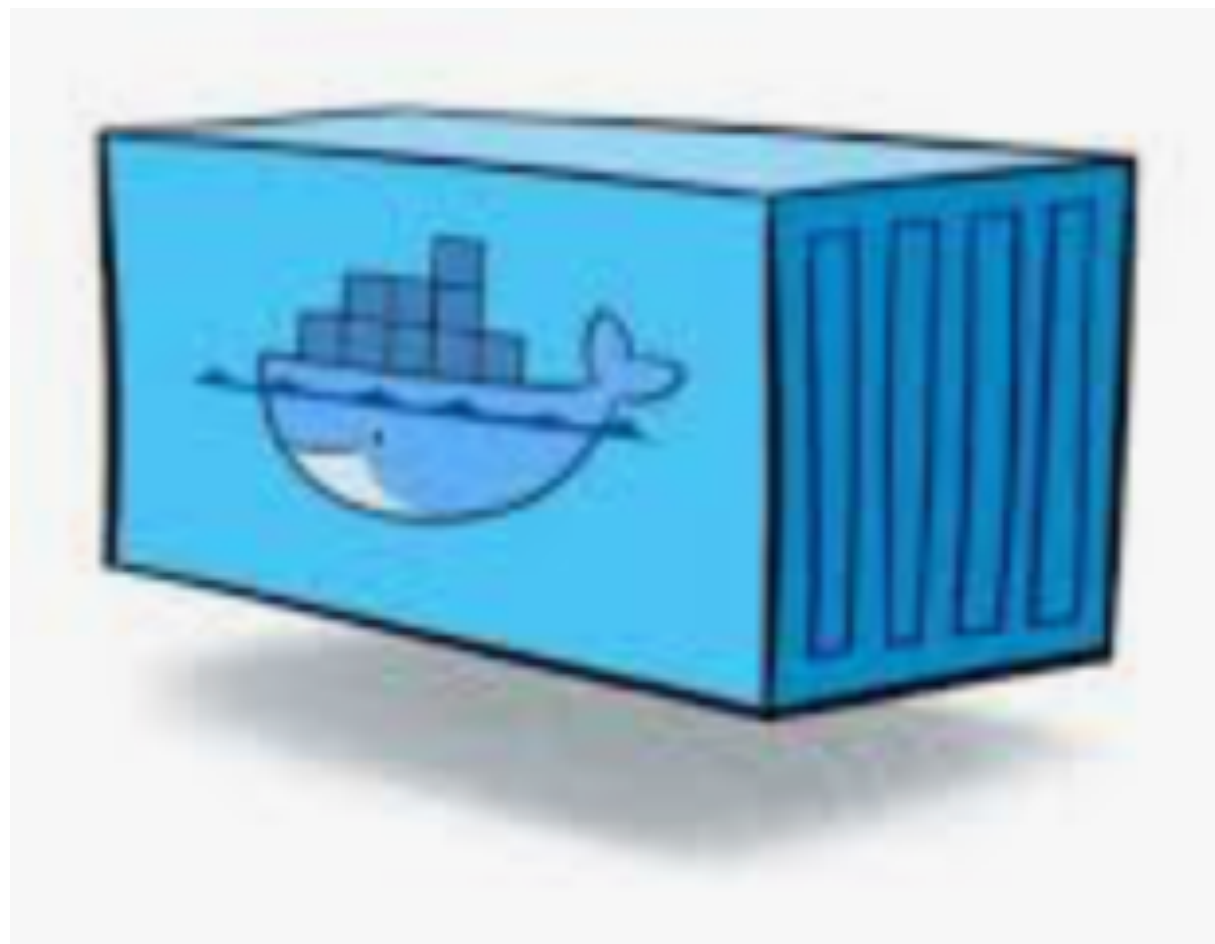
pg_activity

pg_audit

Logical Decoding plugins (Json/Protobuf) for Kafka Integration

Timescaledb & Prometheus Extension

... and an entrypoint / shell script which contains the logic



The PostgreSQL container (continued)

Features:

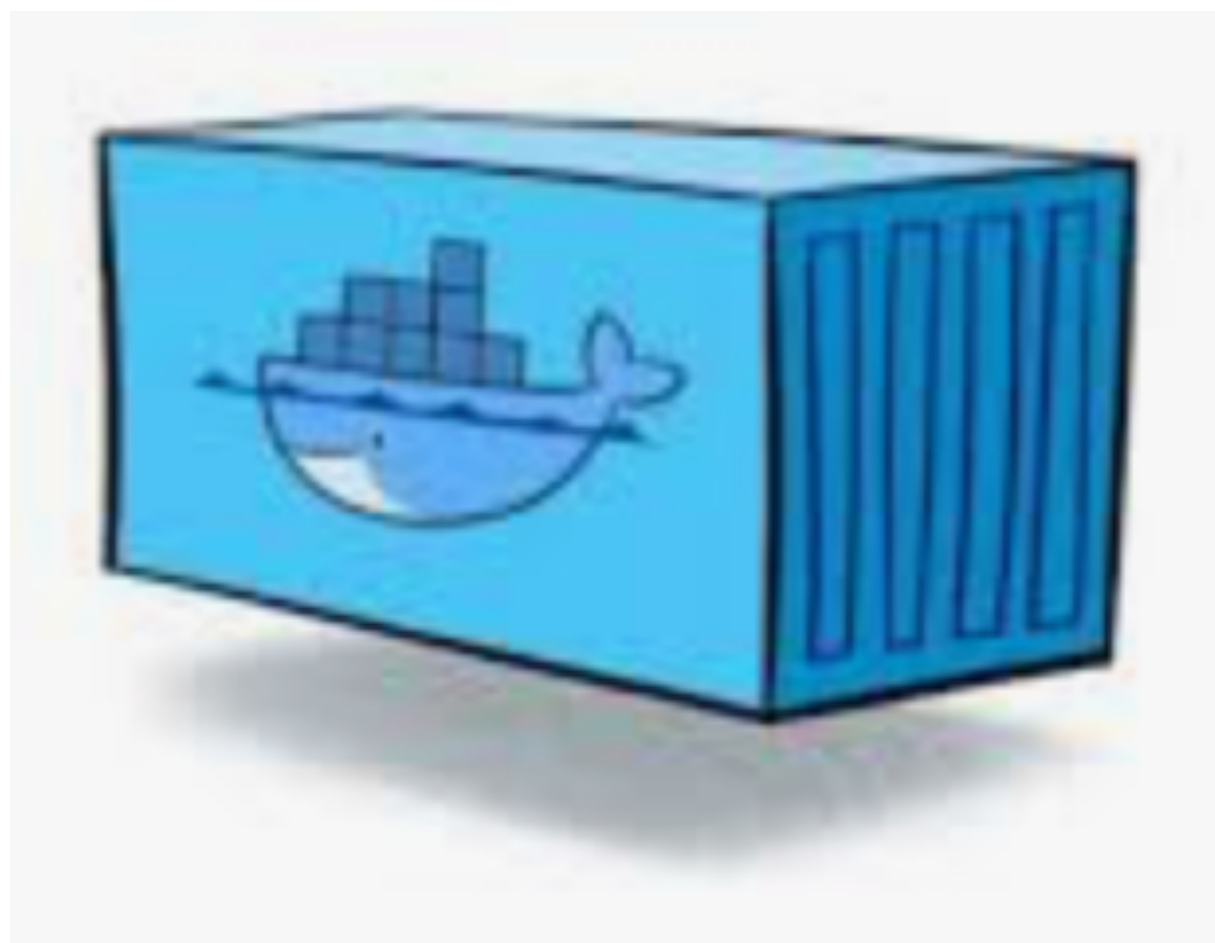
create a new database

- from scratch
- as a slave of some existing db
- as a clone of some existing db

run a major release upgrade of a database

run a full / PIT /restore-point based recovery

WAL-G



The Backup Container

Created from the PostgreSQL container, so it already contains WAL-G binaries.

.. just contains a different entrypoint / shell script with some logic

Features:

write backups straight to object storage (S3, SWIFT, GCS)

allow full or incremental backups

added functionality to allow self-contained backups for database not running wal shipping (by temporarily activating wal shipping during backup execution)

also used to do backup cleanup on object storage



The Metric Container

There was already something .. so lets use it ..

Features:

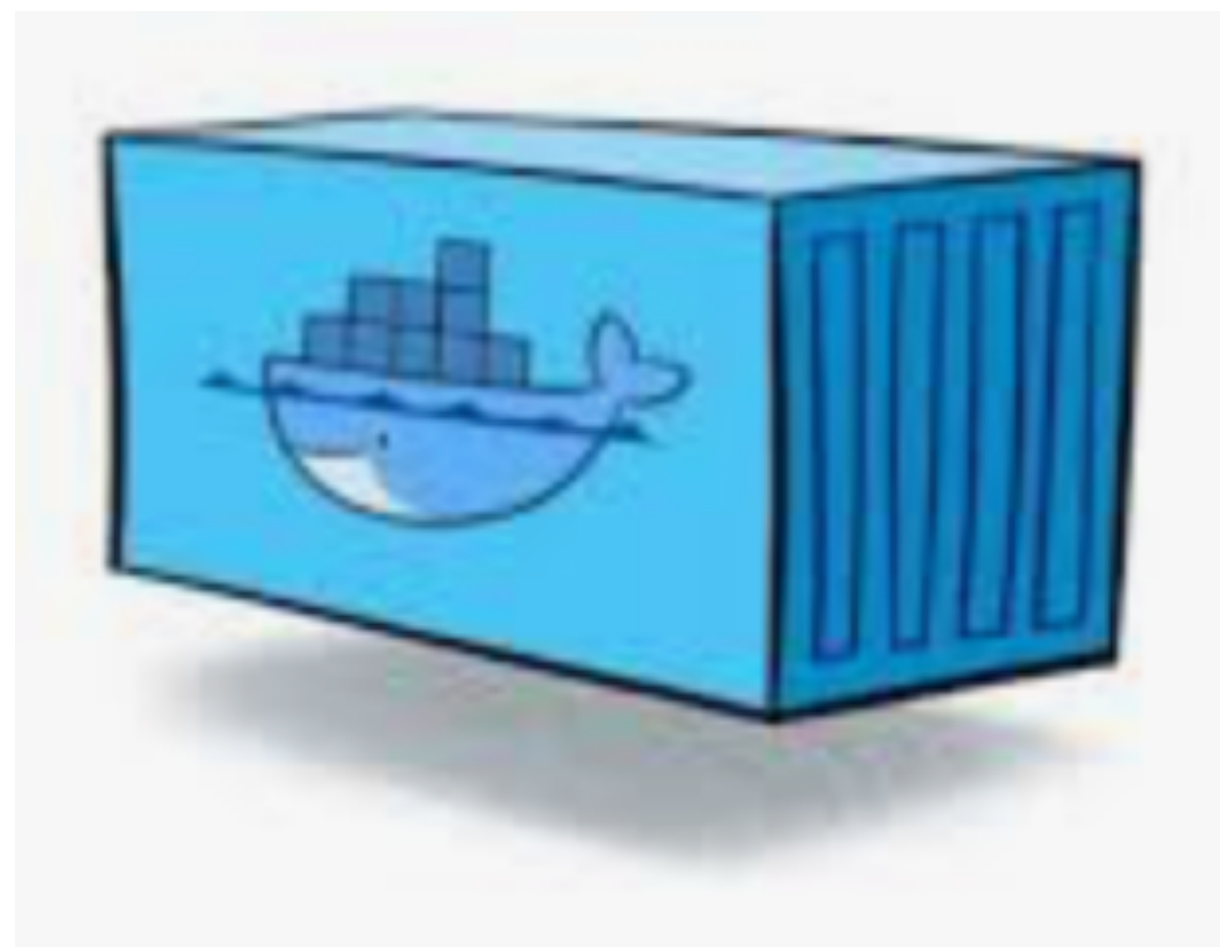
out of the box metric sources for most relevant OS level metrics

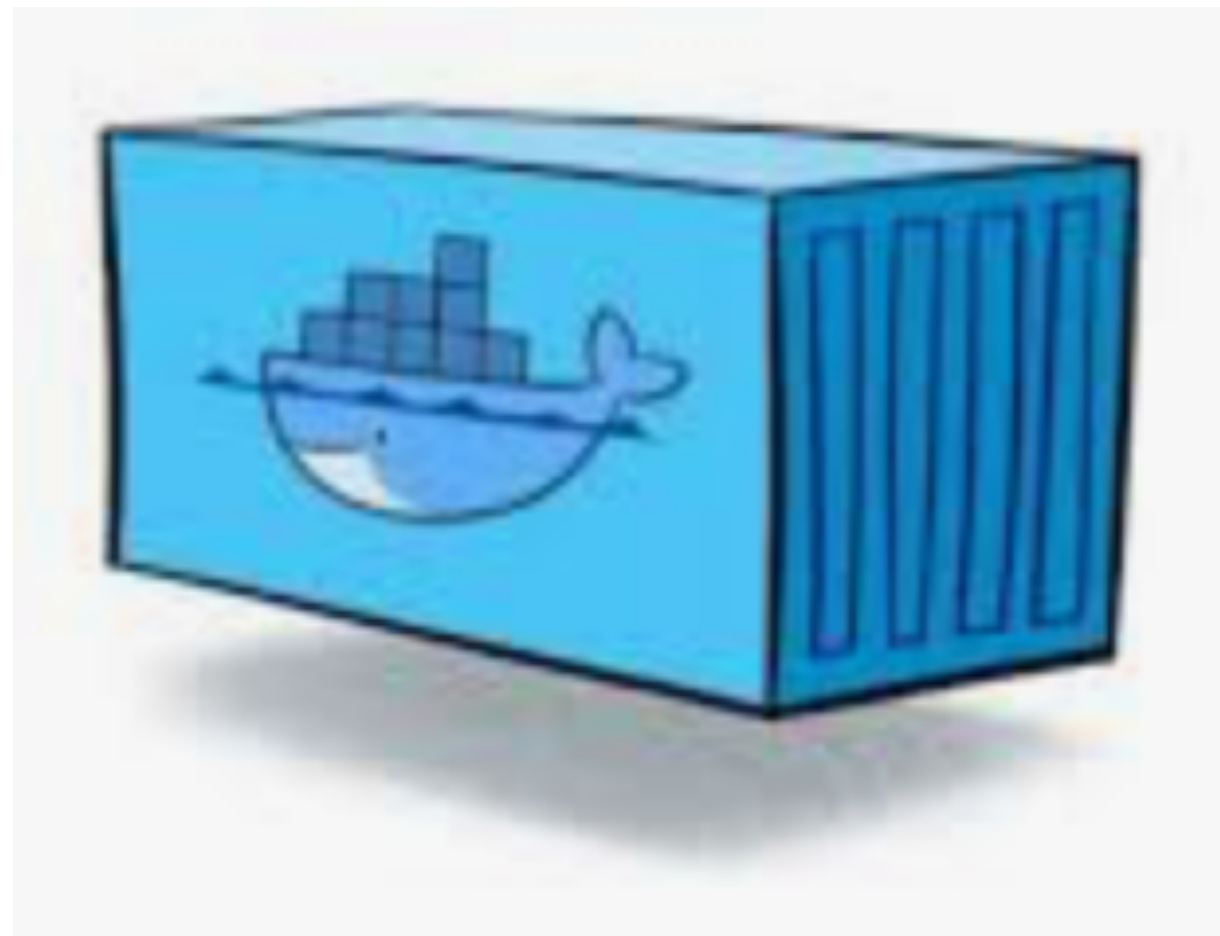
out of the box PostgreSQL metrics

easy extensible to add any custome metrics

easy to containerize

open architecture.. can be used also for any other metric source





The backend and helper container

used wherever possible ready build containers from docker hub

Features:

Influxdb : time series database to store db and host metrics

Grafana : web based dashboard to visualize metrics

pgAdmin : web based admin tool

pgBadger: transforms db logs into HTTP web content

goStatic : minimalistic HTTP server used to publish pgBadger reports

we were able now to ..

spin up some db like this:

```
docker run registry.metronom.com/rdb-dev/postgresql:11
```


we were able now to ..

create db with usr/pwd, mount volume inside and ship wal to S3 storage :

```
docker run --rm -d --name postgresql -d
-v /dockerdata/postgresql:/postgresqldata
-e username=heiko -e password=supersecret -db=appdb1
-e DBUUID=5b54d046-c8f3-447c-8e56-247b7493b4df
-e BACKUP_TYPE=S3
-e BACKUP_URI=//backup.postgresqlbackup.dus30hcp01.asf.madm.net
-e BACKUP_CREDENTIALS="someid:somesecret"
-e PG_WRITE_ARCHIVE=true
registry.metronom.com/rdb-dev/postgresql:11
```


and we were able to do all the other day2 operations :

- take a backup from a running db to some object store
- run a major PostgreSQL release upgrade
- restore/recover a db with all recovery options
- collect metrics and ship them to some metric repo
- publish metrics using Grafana container
- offer pgAdmin web
- publish pgBadger reports to web using HTTP container on DB node

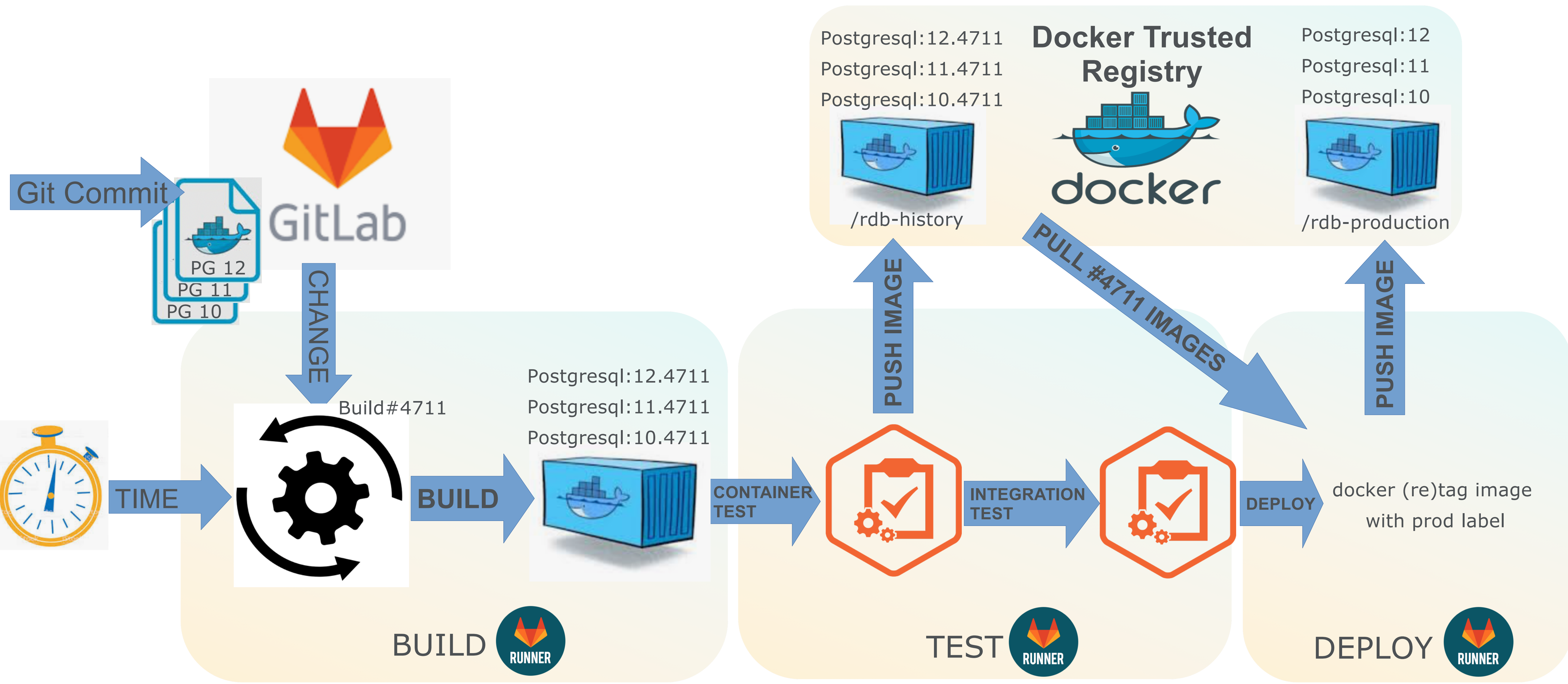
**But who will continuously build and
test all these containers ..**



CHAPTER # 2

The container build pipeline

We created next some container build pipeline ..



Container build pipeline (continued)

The screenshot shows the GitLab CI/CD Pipelines page for the project 'pg-container'. The interface includes a top navigation bar with 'GitLab' and various menu items like 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. A search bar is present on the right. On the left, a sidebar contains navigation options: Project, Repository, Issues (0), Merge Requests (0), CI/CD (selected), Pipelines, Jobs, Schedules, Charts, Operations, Wiki, Snippets, and Settings.

The main content area displays a list of pipeline runs. At the top, there are filters for 'All 333', 'Pending 0', 'Running 0', and 'Finished 328'. There are also buttons for 'Run Pipeline', 'Clear Runner Caches', and 'CI Lin >'. The pipeline runs are listed in a table with columns for Status, Pipeline, Commit, Stages, and duration.

Status	Pipeline	Commit	Stages	Duration	Age
passed	#88160 by latest	PROD4.1 -> 88972eec update README for pipeline tri...	✓ ✓ >> ✓	00:33:10	1 day ago
passed	#88074 by API	PROD4.1 -> 2fe6f960 change integration test report t...	✓ ✓ ✓ ✓ ✓	00:42:00	2 days ago
passed	#86357 by API	PROD4.1 -> 2fe6f960 change integration test report t...	✓ ✓ ✓ ✓ ✓	00:22:13	1 week ago
passed	#85731 by	PROD4.1 -> 2fe6f960 change integration test report t...	✓ ✓ >> ✓	00:23:46	1 week ago
passed	#84479 by API	PROD4.1 -> 28e5d9e7 fix integration test report proble...	✓ ✓ ✓ ✓ ✓	00:23:56	2 weeks ago
passed	#82926 by API	PROD4.1 -> 28e5d9e7 fix integration test report proble...	✓ ✓ ✓ ✓ ✓	00:23:36	3 weeks ago
passed	#81684 by	PROD4.1 -> 28e5d9e7 fix integration test report proble...	✓ ✓ ✓ ✓ ✓	00:34:59	4 weeks ago
passed	#81497 by API	PROD4.1 -> 28e5d9e7 fix integration test report proble...	✓ ✓ ✓ ✓ ✓	00:39:44	4 weeks ago
passed	#81445 by	PROD4.1 -> 28e5d9e7 fix integration test report proble...	✓ ✓ >> ✓	00:21:45	1 month ago

Container build pipeline (continued)

The screenshot shows the GitLab CI/CD interface for a project named 'pg-container'. The top navigation bar includes 'GitLab', 'Projects', 'Groups', 'Activity', 'Milestones', 'Snippets', and a search bar. The left sidebar contains navigation options: Project, Repository, Issues (0), Merge Requests (0), CI/CD (selected), Pipelines, Jobs, Schedules, Charts, Operations, Wiki, Snippets, and Settings.

The main content area displays the pipeline details for 'pg-container' with pipeline ID '#88074'. A green 'passed' badge indicates the pipeline's status. The commit message is 'change integration test report to simple html as temporary solution to prevent pwd offering'. The pipeline summary shows '8 jobs for PROD4.1 in 42 minutes (queued for 5 seconds)'. A commit hash '2fe6f960' is visible.

The pipeline stages and jobs are as follows:

- Cleanup**: cleanup-docker...
- Build**: build-pg-10, build-pg-11, build-pg-12
- Image-test**: test-only-pg
- Integration-test**: integration-test...
- Deploy-prod**: deploy-prod-sc...

All jobs in the pipeline are marked with a green checkmark, indicating they passed successfully.

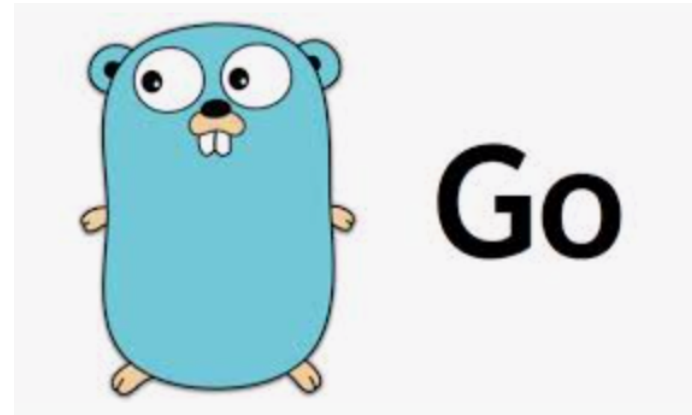
How do we make now a Self Service out of it



CHAPTER # 3

The API first approach !

We build some REST API ..



RESTful

follows micro service principals

written in Go



consumes other REST APIs of cloud and storage providers

ultra high availability as it is runs across several DCs



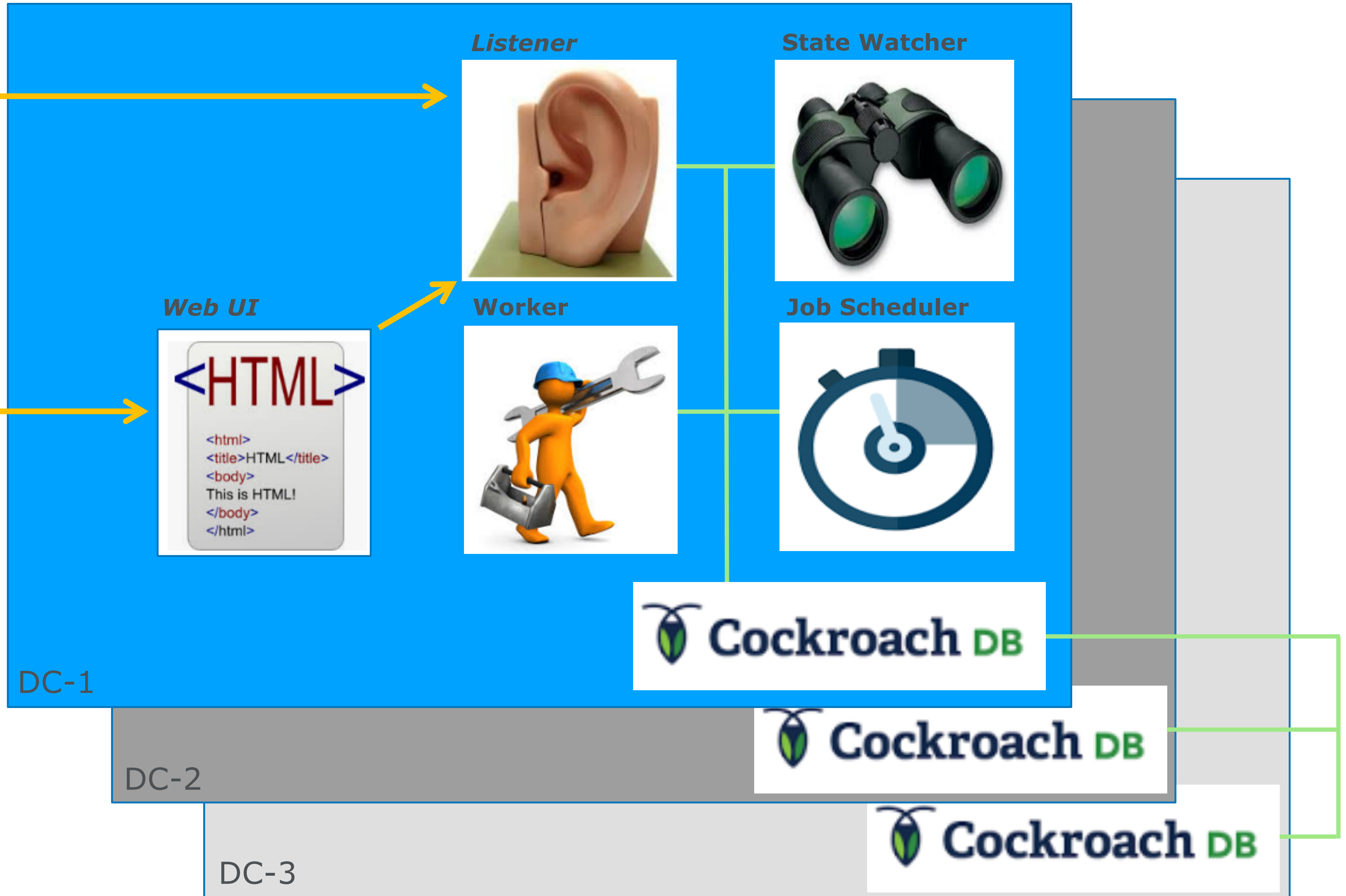
backed by Cockroach DB (solving for us all the distributed problems)

secured by OAuth 2.0

terminated by a distributed load balancer

PostgreSQL DBaaS Restful - API

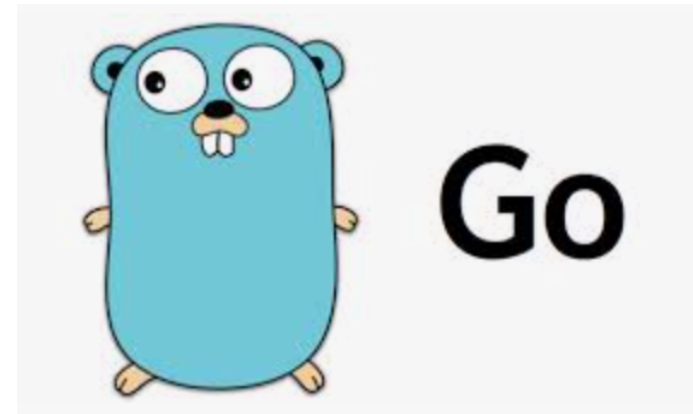
{ REST }



CHAPTER # 4

The self service

In the end we build the **Self Service portal** ..



written in Go (using Go HTML templates, jQuery, Twitter bootstrap)

designed with the target to have a great user experience



wizard for db creation incl. jobs and reports

allow self-service config/backup/restore/recovery



expose dashboards and pgBadger reports to DevOps teams

pay per use

integrate customer feedback so that we better learn how to improve our service

Intuitive to use..

M|RDB Heiko Onnebrink | ADMIN ↻

Databases (PostgreSQL)

Search: [New Database](#)

DB Name	Curr. State	Description	Data Center / Avail. Zone	IP Address	Port	Repl. Status	Tenant	
coreroller	running	CoreRoller DB	DUS11 /	10.96.147.223	5432		RDB_ASM	...
coreroller	running	coreroller DB	DUS11 /	10.96.145.156	5432		RDB_ASM	...
coreroller-flatcar	running	coreroller DB for flatcar linux	DUS11 /	10.96.146.239	5432		RDB_ASM	...
coyo	running	United Slave Database	FFM11 /	10.98.144.72	5432	SLAVE	RDB_UNITED	...
coyo	running	United Database	DUS11 /	10.96.145.132	5432	MASTER	RDB_UNITED	...
coyopp	running	UnitedPP Master - resiez DB	FFM11 /	10.98.144.43	5432	MASTER	RDB_UNITED	...
coyopp	running	UnitedPP Master - resiez DB	DUS11 /	10.96.148.69	5432	SLAVE	RDB_UNITED	...
coyoppdev	running	United DEV database - 28102019	FFM11 / Rack1	10.98.148.33	5432		RDB_UNITED	...
druckerdb	running	Liste aller Drucker am Campus DUS	DUS11 / Rack1	10.96.148.89	5432		RDB_DRUCKERDB	...
fsd-mm-sdb-odoo12	running	ODOO12 MWM session db DEV	DUS21 / Rack1	10.97.180.120	5432	MASTER	RDB_ODOO	...

Previous Page 3 / 12 Next Page size 10

Time Zone: Europe/Berlin | Refresh in 4 seconds Auto Refresh

wizard driven DB (and job) creation..

M RDB Heiko Onnebrink | ADMIN ↗

New Database

- 1 Database Details**
- 2 Sizing
- 3 Access
- 4 Backup & Recovery
- 5 Miscellaneous
- 6 Summary

Database Details

Data Center	Availability Zone	PostgreSQL Version
GCW1	europa-west1-c	11
DB Name	Description	
DEMO	some demo db	
MARC Product		
C1290		
Tenant		
ADMIN		

Cancel Next

Time Zone: Europe/Berlin

summary screen ..

M RDB Heiko Onnebrink | ADMIN ↻

New Database

- ✓ Database Details
- ✓ Sizing
- ✓ Access
- ✓ Backup & Recovery
- ✓ Miscellaneous
- 6** Summary

Summary (Please review all data before you continue!)

DB Name	Description	PostgreSQL Version
DEMO	some demo db	11

Data Center / Avail. Zone	Product	Tenant
GCW1 / europe-west1-c	C1290	ADMIN

Storage Type	Volume (GB)	Flavour
LOCAL	375	developer

Port	User
5432	myuser

Backup job Data Protection

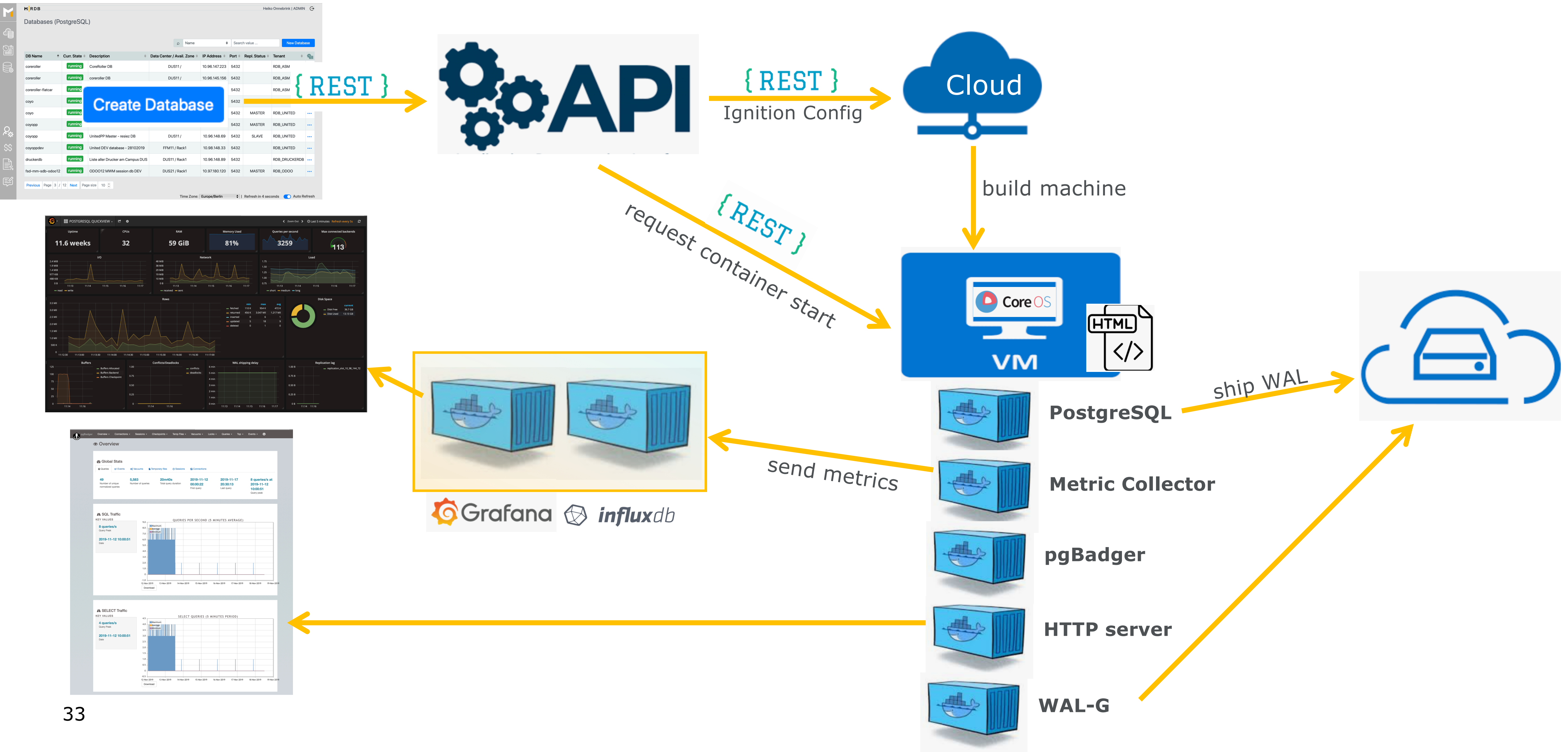
Frequency	Next Run (date & time)
DAY	2019-11-18 22:00 CET

Automatic patching job PgBadger Job

Frequency	Next Run (date & time)	Frequency	Next Run (date & time)
4 WEEKS	2019-11-24 22:00 CET	1 HOUR	2019-11-18 10:00 CET

Cancel Previous Create Database

What happens in the background when a DB gets created



DB details with request and job views ..

M RDB
Heiko Onnebrink | ADMIN [↻](#)

Database List

DB Name (Type & Version)

fsd-mm-sdb-odoo12 (PostgreSQL 10)

Stop
Edit
Config
Slave
Clone
Recover
Statistics
Delete

Description

ODOO12 MWM session db DEV

Data Center / Avail. Zone

DUS21 / Rack1

MARC Product

A1309

Storage Type

LOCAL

Volume (GB)

75

Flavour

smallbusiness

Tenant

RDB_ODOO

IP

10.97.180.120

Port

5432

Replication Status

MASTER

Data Protection

Current State

running

Desired State

running

DB ID

c5bea3c0-c126-4f8a-a069-6fb84cc36b9d

Requests
Jobs

Requests List

Request Type

CREATE-DB

Requested on

2019-11-06 11:47:42 CET

State

PROCESSED

Request Key

501241816420024324

Last Step

Last Error

Body

```

{id": "c5bea3c0-c126-4f8a-a069-6fb84cc36b9d",
"state": "",
"dcName": "DUS21",
"version": "10",
"name": "fsd-mm-sdb-odoo12",
"description": "ODOO12 MWM session db",
"userName": "odoo",
"password": "****masked****",

```

Trace

Timestamp	Action
2019-11-06 11:47:49 CET	start
2019-11-06 11:47:49 CET	validate
2019-11-06 11:47:49 CET	create object storage bucket
2019-11-06 11:47:50 CET	create server
2019-11-06 11:48:17 CET	reboot server
2019-11-06 11:50:27 CET	contact container engine
2019-11-06 11:51:09 CET	ship db container
2019-11-06 11:51:47 CET	ship remaining container
2019-11-06 11:51:56 CET	end

Time Zone: Europe/Berlin [↕](#)
Refresh in 1 seconds Auto Refresh

cost details per tenant ..

M|RDB Database Costs
Heiko Onnebrink | ADMIN

DB Deployments

DB Flavour Legend:

developer smallbusiness powerbusiness highend

DUS11

DUS21

Costs

Reporting Time Period:

Year: 2019

Entire Fiscal Year:

Month: 11

[Get Costs](#)

Data for current month 11

000-KCENTER	
000-UNITED	
217-4539	
241-6286	
241-6290	
392-7021	
801-4299	
801-6273	
991-0717	
991-0767	

Previous Page 1 / 2 Next Page size 10

Costs Details For PSC 241-6286

Collect Name	Amount	Price / Unit (EUR)	Cost (EUR)
DB flavor analytics with local storage	6009 h	0.7924	2022.51
DB flavor developer with local storage	419 h	0.0926	38.80
DB flavor highend with local storage	419 h	0.0926	38.80
DB flavor powerbusiness with local storage	838 h	0.23	135.17
DB flavor smallbusiness	419 h	0.078	32.68
DB flavor smallbusiness with local storage	3063 h	0.1422	366.76
Network attached disk space	41900 GB/h	0.000702	29.41

Cancel

Total Costs: 10861.10 EUR

Costs (EUR)	
289.11	...
1396.90	...
6.72	...
2664.14	...
1343.52	...
354.12	...
20.75	...
38.89	...
939.86	...
2290.33	...

and we value customer feedback..

The screenshot shows a database management interface for PostgreSQL. A feedback modal is open in the center, titled "Thank you for giving us a feedback!". The modal contains the following text: "Please express your overall experience with our product by clicking the corresponding tile below. If you like, you can also write some comments or just request that we contact you." Below the text are five emoji tiles: a red sad face, a red frowny face, a yellow neutral face, a yellow happy face, and a green happy face. A text input field is labeled "Please enter your comments here". At the bottom of the modal, there is a "Contact me" toggle switch (currently off) and two buttons: "Not now" and "Submit".

The background interface shows a list of databases with columns for DB Name, Curr. State, and a table of database details. The user is logged in as Heiko Onnebrink | ADMIN.

DB Name	Curr. State
coreroller	running
coreroller	running
coreroller-flatcar	running
coyo	running
coyo	running
coyopp	running
coyopp	running
coyoppdev	running
druckerdb	running
fsd-mm-sdb-odoo12	running

Repl. Status	Tenant
	RDB_ASM
	RDB_ASM
	RDB_ASM
SLAVE	RDB_UNITED
MASTER	RDB_UNITED
MASTER	RDB_UNITED
SLAVE	RDB_UNITED
	RDB_UNITED
	RDB_DRUCKERDB
MASTER	RDB_ODOO

LAST CHAPTER

lessons learned

- **automize everything you do more than once**
- **use automated testing from the very beginning**
- **know your customer, build what he needs and make him productive**
- **start with a minimal viable product (MVP)**
- **grow from here feature by feature**
- **it takes some time to get there.. but its worth the effort**

once you got there you will gain time, quality and happy customer..

.. and have much more time to go to conferences and give a talk 😊

**Thank you for
your attention!**

the PostgreSQL team @
METRO|NOM