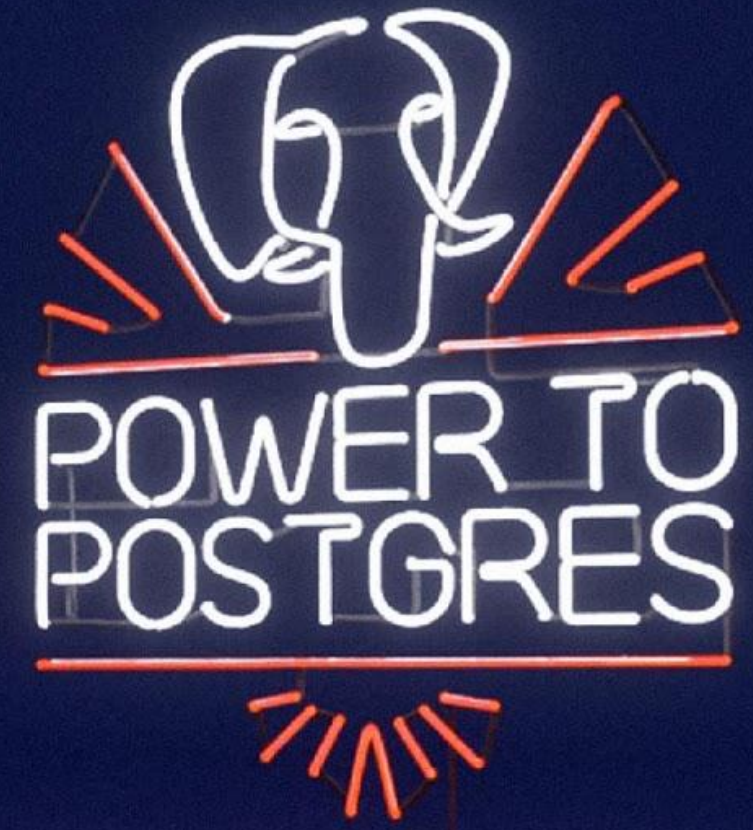
Fuel the DevOps movement and innovate faster with Cloud Native Postgres

Jan Karremans & Gabriele Bartolini

April 2022



EDB

“Postgres is
the most transformative tech
since Linux”

Marc Linster - EDB CTO

Lay of the land

Postgres

Cloud

Cloud Native

Your moderators

Gabriele Bartolini

- EDB's VP for Cloud Native
- Co-founder of 2ndQuadrant (2008-2020)
- Postgres, Kubernetes, and DevOps enthusiast
- Co-founder of PostgreSQL Europe
- Founding member of Barman

Jan Karremans

- EDB Global director of Product Marketing
- 30 years of database technology
- 15 years of consulting, sales & management
- 10 years of community service
- Oracle ACE Alumni
- EDB Postgres Advanced Server professional

Agenda

- Postgres and EDB Postgres
- Monolith to agile
- Dora

Why Postgres?


Miller
Hallenmiller &
Dowling
Architects
Chicago, Ill.
1919

Why did PostgreSQL win?

It does everything...

- Migration
- New App Development
- Replatforming to Cloud and Containers
- System of Record
- System of Analysis
- System of Engagement

It works everywhere...

- Public Cloud - IaaS
- Public Cloud - DBaaS
- Private Cloud
- Virtual Machines
- Containers

and doesn't lock you in

Where are you on your journey with Postgres?

Strategic

- PostgreSQL is a “standard”
- Tier 1 apps are in production
- Scalability, HA, and automation required

Expanding

- Going to production
- Looking for better tools and automation
- Security is increasingly important

Emerging

- Kicking the tires
- Looking at a single project
- Test/dev

Why EDB?

We're the PostgreSQL experts

1986
The design of PostgreSQL

1996
Birth of PostgreSQL

2004
EDB is founded

2007
2ndQuadrant launched

2020
EDB acquires 2ndQuadrant

Key PostgreSQL Contributions

EDB

- Heap Only Tuples (HOT)
- Materialized Views
- Parallel Query
- JIT Compilation
- Serializable Parallel Query

2ndQuadrant

- Hot Standby
- Logical Replication
- Transaction Control in Procedures
- Generated Columns

No company has contributed more to PostgreSQL

We have the most PostgreSQL experts

EDB TEAM INCLUDES:

- 300+ PostgreSQL technologists
- 26 PostgreSQL community contributors and committers
- Including founders and leaders like

Michael Stonebraker
"Father of Postgres" and EDB Advisor

Bruce Momjian
Co-founder, PostgreSQL Development Corp and PostgreSQL Core Team

Peter Eisentraut
PostgreSQL Core Team member

Robert Haas
PostgreSQL Major Contributor and Committer

Simon Riggs
PostgreSQL Major Contributor, Founder of 2ndQuadrant

Monolithic to Agile

Cloud changes much more than just your deployment method

Enabling speed and innovation

From buzzword bingo to business benefits

Agile

Ensure development and deployment teams can (re)deploy and test quickly and seamlessly

Microservices

Transform traditional monolithic applications to cloud native, microservices based solutions

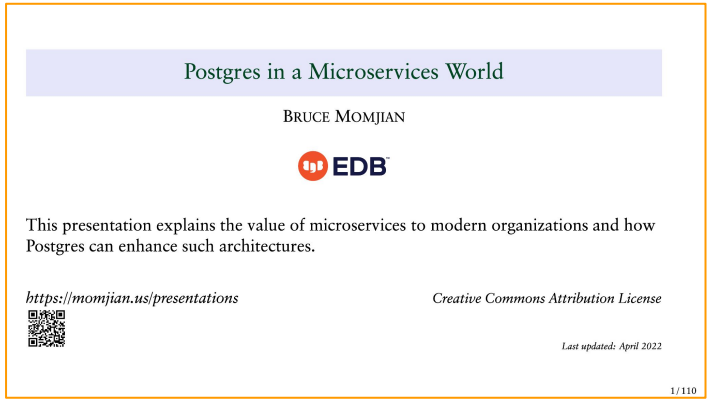
DevOps

Development and deployment of applications are no longer disjointed operations but fully integrated

CI/CD

New value for your solution, released quickly and securely in short and safe deployment cycles

Look out for **Bruce Momjian**'s talk



Coming to a conference near you soon

Why DevOps?

Introduce DORA

Dev**O**ps **R**esearch and **A**ssessment

The longest running academically rigorous research investigation of its kind

Providing an independent view into the practices and capabilities

That drive high performance in technology delivery and organizational outcomes

https://www.devops-research.com/research.html

Version control

Software should be developed using a version control system, like Git

- Help organize source code, test and deployment scripts, configuration information, and more
- Keep track of changes over time
- Enables reproducibility and traceability
- Benefits:
  - response to defects
  - disaster recovery
  - auditability
  - capacity management
  - higher quality
- Strong requirement for automated CI/CD pipelines

https://cloud.google.com/architecture/devops/devops-tech-version-control

Trunk based development

Development happens in the “main” branch (aka the trunk)

- No more than a couple of active branches in the repository
- Work in small batches
- Avoid code freezes!

https://cloud.google.com/architecture/devops/devops-tech-trunk-based-development

Continuous integration (CI)

Effectively coordinate code updates at scale

- Developers regularly integrate all their work into the trunk - possibly daily
- **Goal:** fast feedback on code changes to improve quality in the product
  - Even the smallest change can have unintended side effects on the rest of the system
  - Accentuated when working with multiple developers
- Ideally, each commit in a CI system should trigger:
  - An automated build
  - An automated set of tests
- A Test Driven Development (TDD) mindset helps

https://cloud.google.com/architecture/devops/devops-tech-continuous-integration

Deployment automation

It is no longer just Ops problem!

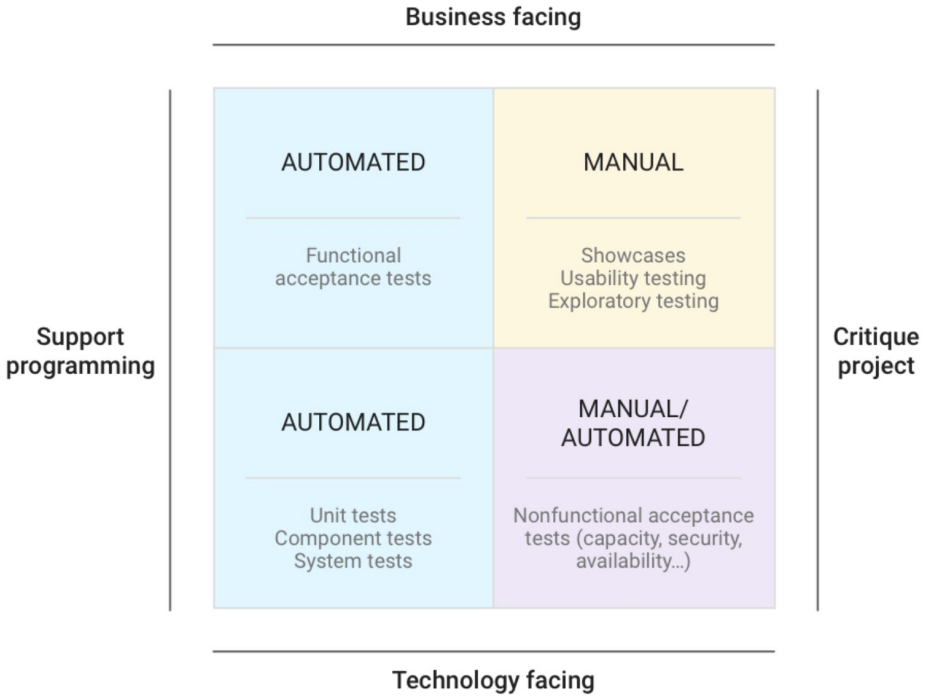
- Automate manual idempotent tasks
- Relly on artifacts (packages or container images) built by CI
- Deploy them to ANY environment – including production - in an automated way
- These also include scripts for configuring, deploying and testing
- Same process, only different environment specific information

https://cloud.google.com/architecture/devops/devops-tech-deployment-automation

Continuous testing

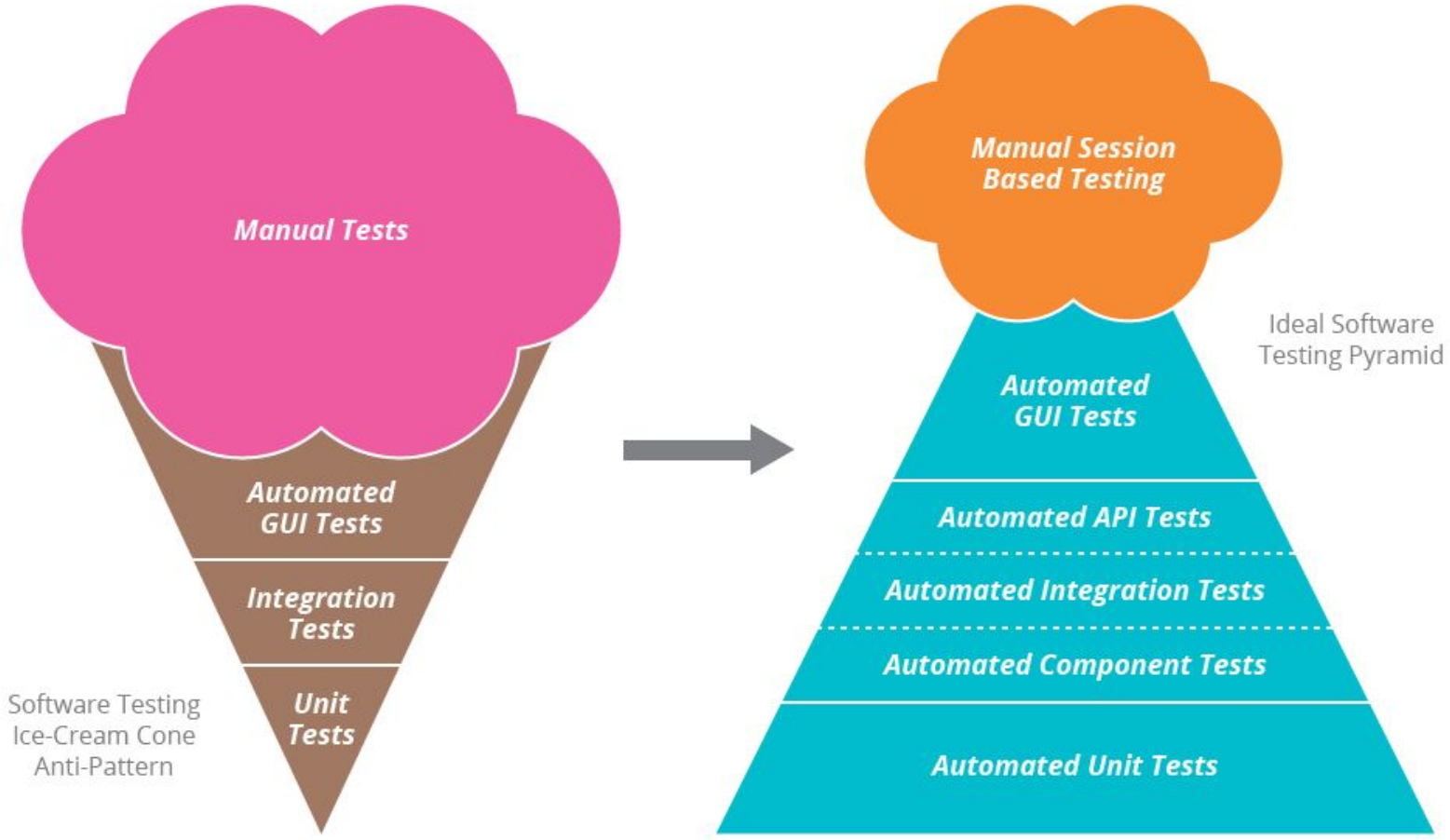
Getting fast feedback on the impact of changes

- Getting feedback early and fast is crucial
- Eliminate human errors in testing
- Fast feedback increases code quality – the learning effect
  - Code baseline quality
  - Testability of code
- Continued test documentation evolution



https://cloud.google.com/architecture/devops/devops-tech-test-automation

Run away from the inverted testing pyramid



Continuous delivery

The ability to release changes of all kinds on demand quickly, safely, and sustainably

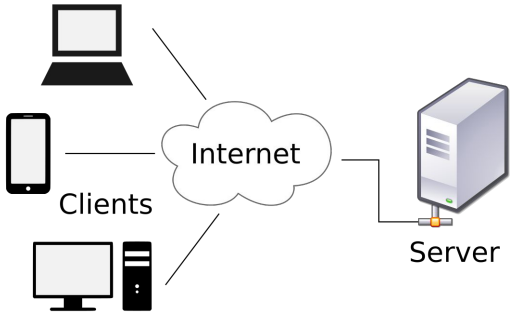
- Continuous Deployment ≠ Continuous Delivery!!
  - Continuous delivery → Can we do it
  - Continuous deployment → Are we doing it
- Is the software deployable throughout its lifecycle!
- Deployability over new features
- Can you deploy the system to production, or to end users, at any time, on demand?
  - **This is when we generate value for the business** (superset of customer value)
- State of mind, based on my experience

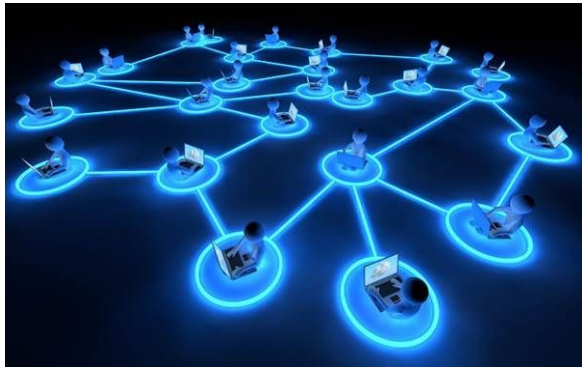
https://cloud.google.com/architecture/devops/devops-tech-continuous-delivery

Architecture

Capabilities that drive higher software delivery and organizational performance

- Choosing the right architecture for your system
- Monolithic? Microservices?
- Depend on the organization and the team
  - You can't ignore Conway's law and cognitive load considerations





https://cloud.google.com/architecture/devops/devops-tech-architecture

Empowering teams to choose tools

Flash news: it is not about getting more tools!

The obvious way is not always the best way

- Periodically assess the technology stack
  - Establish a cross-team baseline
- Define clear APIs between teams and components
- Periodically investigate and evaluate tools and tool usage
- Schedule time to experiment and document the experiments
  - Don't shy away from exceptions
  - Present and discuss new tools
- It’s all about the people

https://cloud.google.com/architecture/devops/devops-tech-teams-empowered-to-choose-tools

Test data management

The result of a test is only as good as the foundation is was run on...

- Good test data lets you
  - Validate common or high value user journeys
  - Test for edge cases
  - Reproduce defects
  - Simulate errors
- One of the more undervalued core areas

https://cloud.google.com/architecture/devops/devops-tech-test-data-management

Shifting left on security

Security is not a last-minute quick-fix

Security is *everyone's* responsibility.

- Get InfoSec involved in software design
- Built preapproved code
- Integrate security reviews
- Automatically test for security

https://cloud.google.com/architecture/devops/devops-tech-shift-left-on-security

Database change management

Every change is a migration

- "Database administration" is a continued joint effort between DBA's and Dev's
  - Establish effective communication of database changes
- Treat database schema changes as migrations
- Create zero-downtime database changes
  - Decouple database and application changes
  - Evaluate schema-less methodologies (JSON) where it makes sense

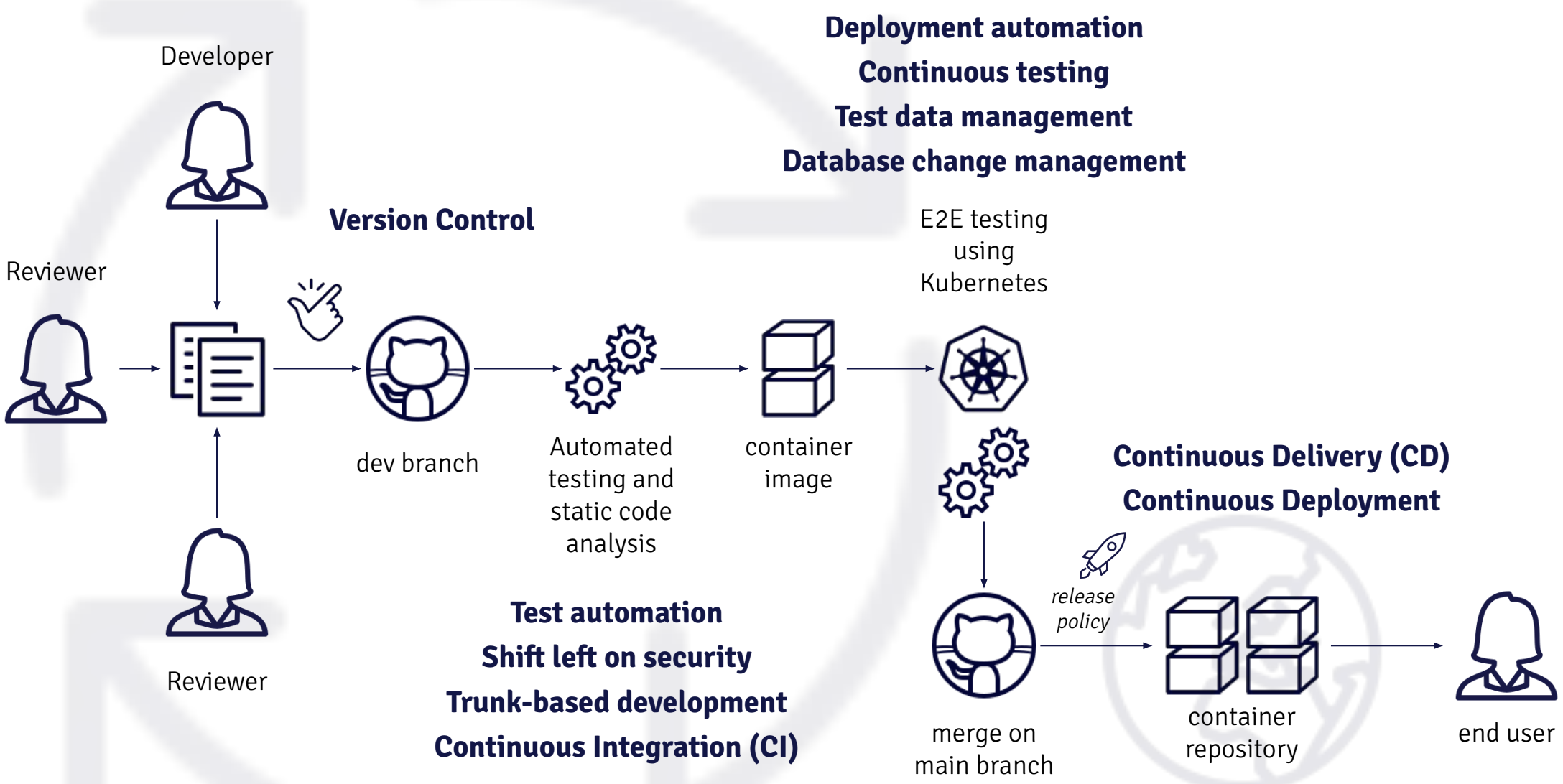
https://cloud.google.com/architecture/devops/devops-tech-database-change-management

Cloud infrastructure

Do not use traditional datacenter practices for Cloud

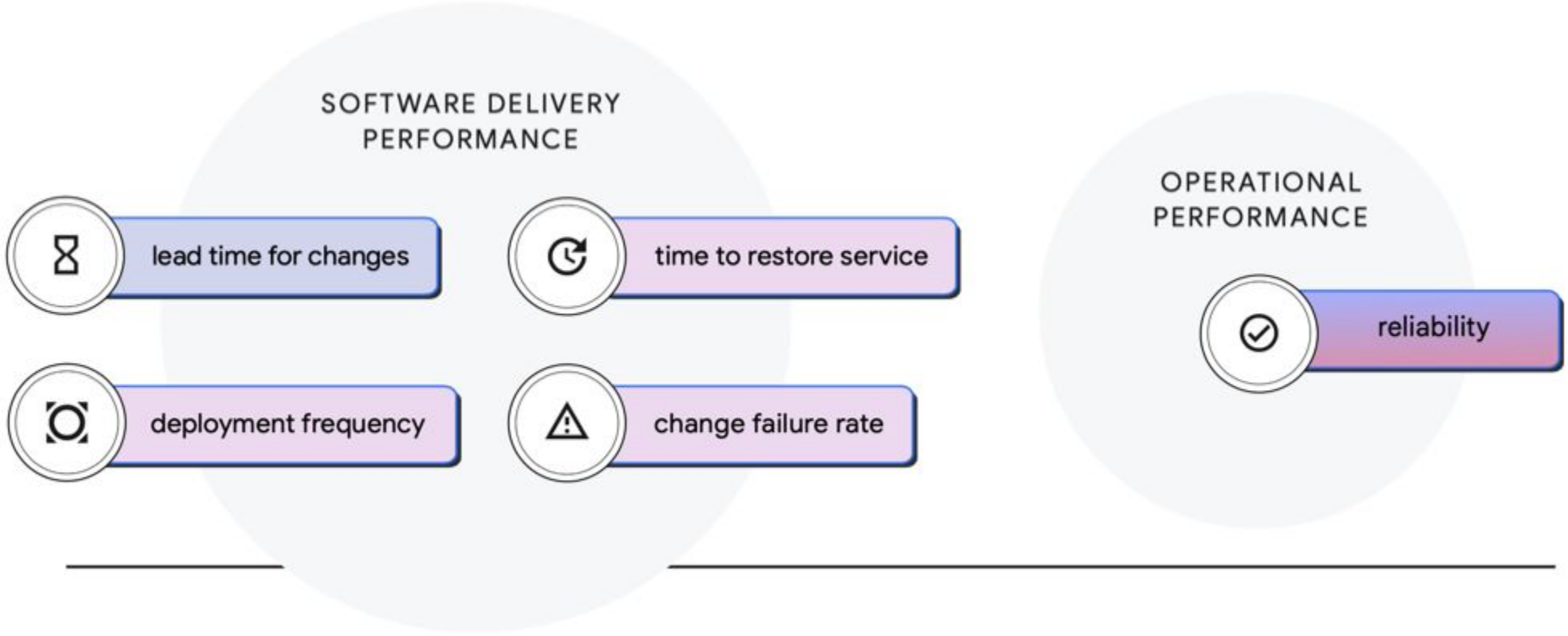
- Improved software delivery performance
- Better service availability
- Improved cost visibility
- Access unparalleled elasticity
- Ephemeral infrastructure (mention) - Kubernetes - cattle vs pets?

https://cloud.google.com/architecture/devops/devops-tech-cloud-infrastructure



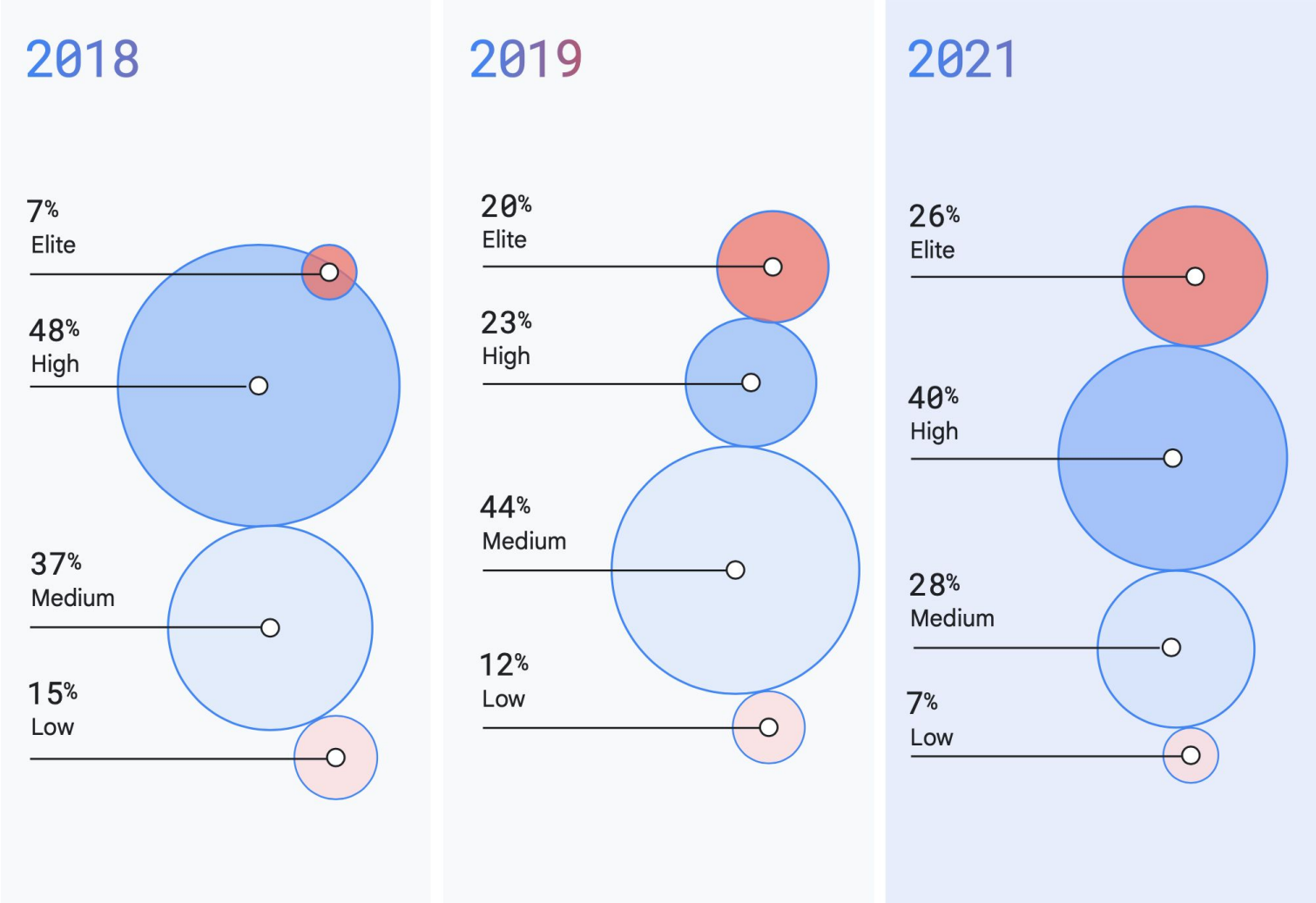
From availability to reliability

It is all about the metrics



Report: *Accelerate state of DevOps 2021*

The industry continues to accelerate



Report: *Accelerate state of DevOps 2021*

Elite performers

973x

more frequent
code deployments

6570x

faster lead time
from commit to deploy

(yes, you read that correctly)

3x

lower change failure rate
(changes are ⅓ less likely to fail)

6570x

faster time to recover
from incidents

https://www.devops-research.com/research.html

Report: *Accelerate state of DevOps 2021*

*Cloud native technologies empower organizations to **build** and **run scalable applications** in modern, **dynamic environments** such as public, private, and hybrid **clouds**. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*

*These techniques enable **loosely coupled systems** that are **resilient**, **manageable**, and **observable**. Combined with r**obust automation**, they allow engineers to make **high-impact changes frequently** and **predictably** with **minimal toil**.*

*The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining **an ecosystem of open source**, **vendor-neutral** projects. We **democratize** state-of-the-art patterns to make these innovations **accessible for everyone**.*

EDB Cloud Native Postgres capabilities

Enabling true DevOps with Postgres

Deploy anywhere

Lightweight, immutable Postgres containers

Automate DBA Tasks

Failover, switchover, backup, recovery, and rolling updates

Avoid lock-in

Operator and images are portable to any cloud

Save the Date

KubeCon

CloudNativeCon

Europe 2022

16 – 20 MAY
VALENCIA, SPAIN + VIRTUAL