
EDB + Red Hat
Plasma il Futuro: Postgres, Kubernetes e Architetture Cloud-Neutral
10 June | Florence, Italy

Agenda

10:00 – 10:30	Arrival & Welcome Coffee ☕
10:30 – 10:45	Opening Remarks
10:45 – 11:15	Keynote: Running PostgreSQL on Bare-Metal Kubernetes
	Gabriele Bartolini, VP, Chief Architect, Kubernetes, EDB
11:15 – 11:45	The Role of Kubernetes in Modern IT Infrastructures
	Natale Vinto, Technical Evangelism Director, Red Hat
12:15 – 12:30	Break & Networking ☕
12:30 – 13:00	PostgreSQL as an AI Data Management Platform for OpenShift AI
	Natale Vito, Technical Evangelism Director, Red Hat
	David Tammaro, Principal Sales Engineer, EDB
13:00 – 13:30	Panel Discussion
13:30–14:30	Networking Lunch
14.30 - 17.30	Postgres on Openshift workshop
	Natale Vito, Technical Evangelism Director, Red Hat
	David Tammaro, Principal Sales Engineer, EDB
	Sergio Romera, Senior Manager, Sales Engineering, EDB

Panel Discussion

Cloud-Neutral Strategies for Data Sovereignty & Portability

Andrea Rizzi

VP, Sales EMEA South

EDB

Moderator

Hervé Timsit

CRO

EDB

Speaker

Franck Sidi

CTO

EDB

Speaker

Natale Vinto
Technical Evangelism Director

Red Hat

Speaker

Piergiorgio Spagnolatti

Head of Infrastructure

BPS

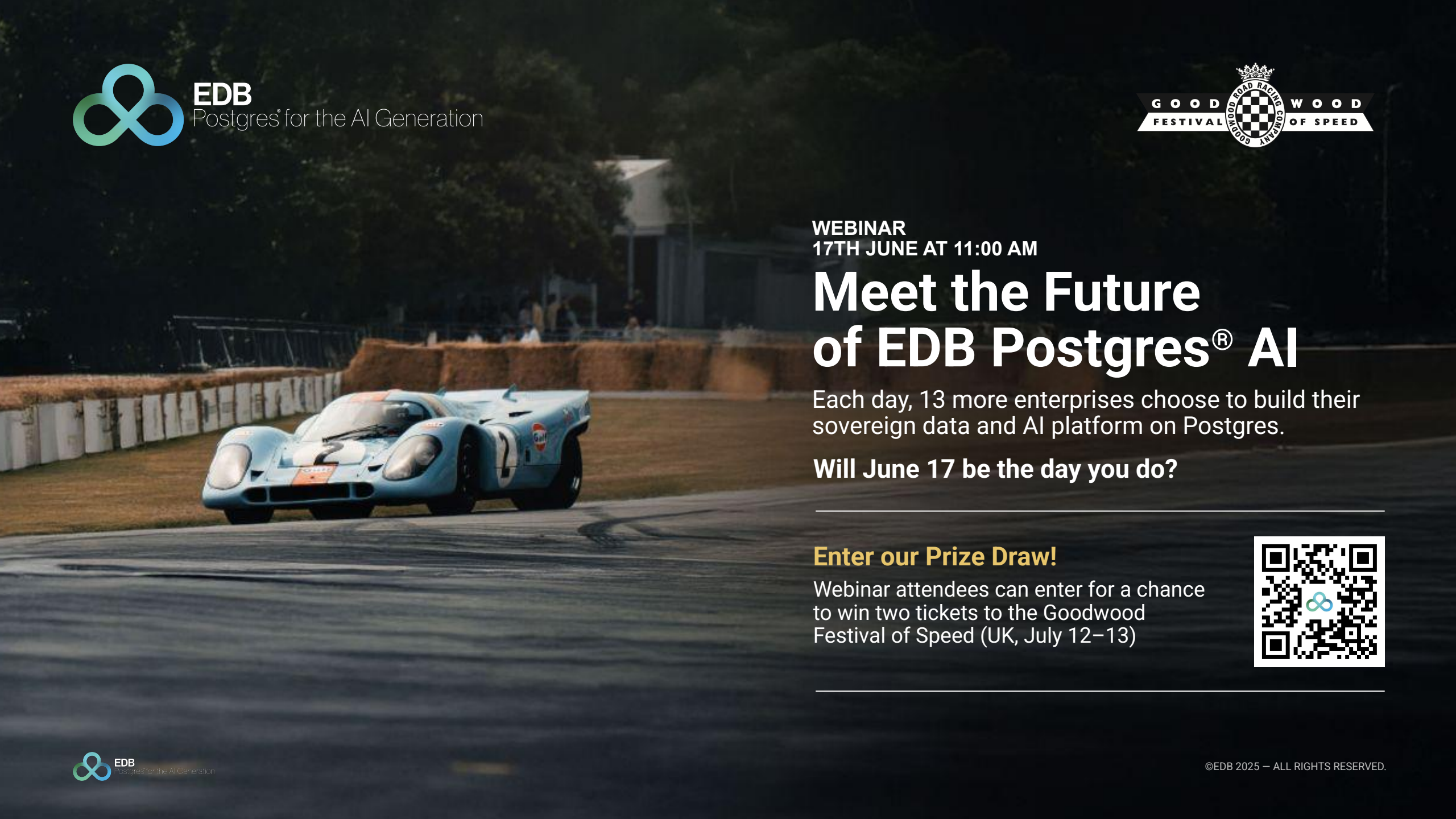
Speaker

Gianni Brandani

CTO

Quid

Speaker



EDB
Postgres® for the AI Generation

GOODWOOD FESTIVAL OF SPEED

WEBINAR
17TH JUNE AT 11:00 AM

Meet the Future of EDB Postgres® AI

Each day, 13 more enterprises choose to build their sovereign data and AI platform on Postgres.

Will June 17 be the day you do?

Enter our Prize Draw!

Webinar attendees can enter for a chance to win two tickets to the Goodwood Festival of Speed (UK, July 12–13)

Running PostgreSQL on Bare-Metal Kubernetes

Gabriele Bartolini
VP, Chief Architect, Kubernetes at EDB
Firenze, 10 giugno

Why shall I consider running Postgres on bare-metal nodes in Kubernetes?

(what!?!?! Bare-metal!?!?!)

Gabriele Bartolini

VP, Chief Architect of Kubernetes at EDB

PostgreSQL user since ~2000

Ex 2ndQuadrant (Co-founder)

PostgreSQL Contributor

DoK Ambassador

DevOps evangelist

Open Source contributor

- Barman (2011)
- CloudNativePG (2022)

Blog: gabrielebartolini.it @_GBartolini_

Disclaimer

For simplicity, "Kubernetes" is used as a general term to refer to concepts that also apply to Red Hat OpenShift.

EDB
Postgres® for the AI Generation

Agenda

- Architectures Overview
- Postgres recommendations
- The Bare-Metal opportunity
- Migrating Postgres
- Key takeaways

EDB
Postgres® for the AI Generation

The Context

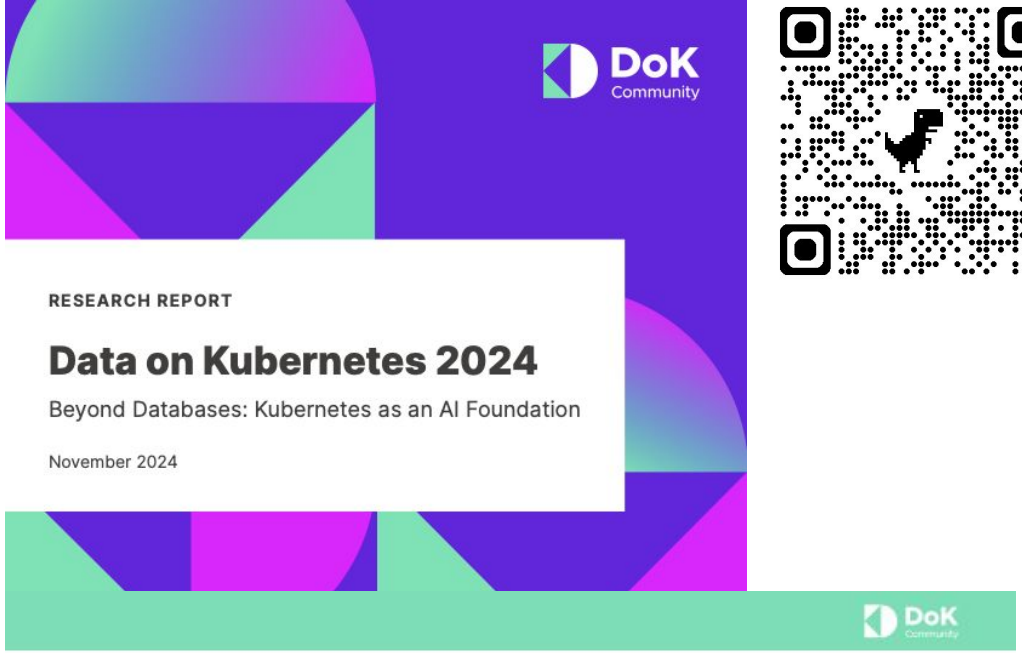
Today's Landscape

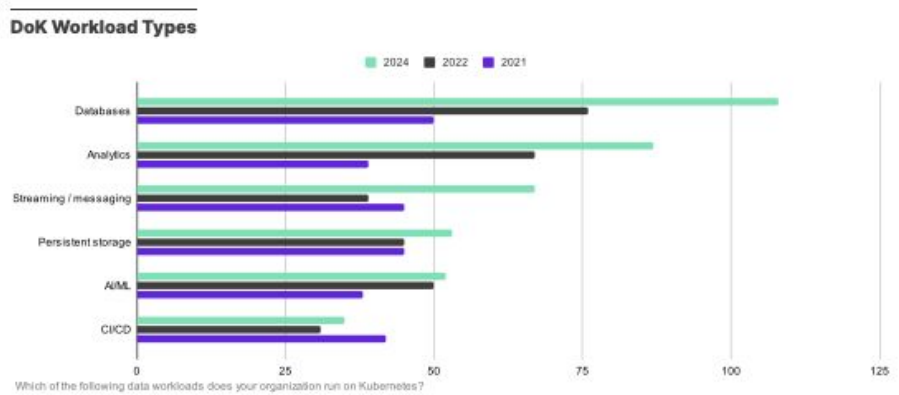
Innovation, data sovereignty, cloud neutrality, AI, data portability, vendor lock-in mitigation, and TCO reductions are fueling the growing adoption of Postgres databases on Kubernetes.

Data on Kubernetes Community

Databases are #1 workload in Kubernetes

The Future of DBaaS on Kubernetes - M. Logan, S. Pronin, D. Sigireddi, G. Bartolini
Kubecon NA 2024 - Panel





Database Workloads: The Steady Foundation

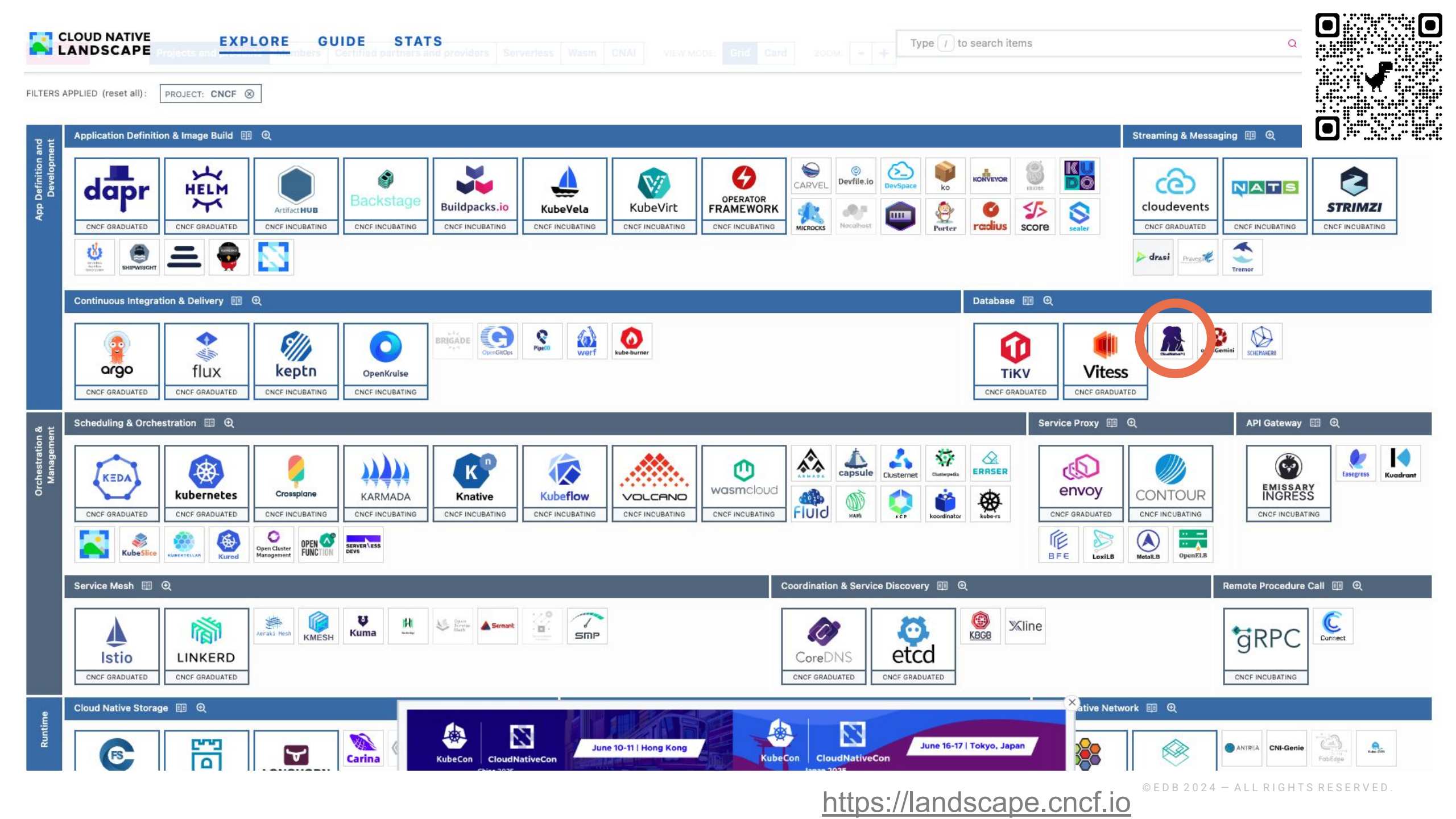
Databases continue to be the cornerstone of DoK deployments. For the third consecutive year, databases remain the most common DoK workload, demonstrating the platform's reliability for critical data services. The consistency in database workload adoption demonstrates:

1. **Platform Reliability:** Organizations trust Kubernetes for critical data services.
2. **Operational Standardization:** Growing comfort with running databases on Kubernetes.
3. **Deployment Confidence:** Increased willingness to run production database workloads.

Evolution of PostgreSQL in containers

From Docker system containers to Kubernetes native databases with CloudNativePG

- **2013/3**: Docker is released. Postgres runs mainly for testing in system containers
- **2015/7**: Kubernetes 1.0 is released. Stateless applications only.
- **2016/11**: Operator pattern by CoreOS
- **2017/3**: Crunchy Data releases the first Postgres operator based on Patroni
- **2017/12**: Statefulsets are introduced in Kubernetes 1.9 (*1 year after beta in 1.5*)
- **2018/8**: Zalando releases their operator
- **2019/4**: Local persistent volumes are introduced in Kubernetes 1.14
- **2019/8**: The Cloud Native initiative at EDB (2ndQuadrant at that time) begins
- **2021/2:** EDB launches Cloud Native Postgres
- **2022/5: EDB open sources CloudNativePG**
- **2024/10**: CloudNativePG reaches 4500 stars on GitHub (#1 Postgres operator)
- **2025/1**: **CloudNativePG** becomes a **CNCF** project entering the **Sandbox**



https://landscape.cncf.io

CloudNativePG

Certified

CloudNativePG is a Kubernetes operator that covers the full lifecycle of a PostgreSQL database cluster with a primary/standby architecture, using native streaming replication

CloudNativePG is a Kubernetes operator that covers the full lifecycle of a PostgreSQL database cluster with a primary/standby architecture, using native streaming replication

→ Self-Healing and automated failover

In case of detected failure on the primary, the operator will change the status of the cluster by setting the most aligned replica as the new target primary. As a consequence, the instance manager in each alive pod will initiate the required procedures to align itself with the requested status of the cluster, by either becoming the new primary or by following it. In case the former primary comes back up, the same mechanism will avoid a split-brain by preventing applications from reaching it, running `pg_rewind` on the server and restarting it as a standby. Self-healing is enhanced by the automated recreation of a standby: in case the pod hosting a standby is removed, the operator initiates the procedure to recreate a standby server.

EDB Postgres for Kubernetes

Certified

PostgreSQL Operator for mission critical databases in Openshift Container Platform

EDB Postgres for Kubernetes is an operator designed, developed, and supported by EDB that covers the full lifecycle of a highly available Postgres database clusters with a primary/standby architecture, using native streaming replication. The operator has been renamed from EDB Cloud Native PostgreSQL. It is based on the open source CloudNativePG operator, and provides additional value such as compatibility with Oracle using EDB Postgres Advanced Server, additional supported platforms such as IBM Power and OpenShift. EDB Postgres for Kubernetes uses the Restricted SCC.

→ Self-Healing and automated failover

In case of detected failure on the primary, the operator will change the status of the cluster by setting the most aligned replica as the new target primary. As a consequence, the instance manager in each alive pod will initiate the required procedures to align itself with the requested status of the cluster, by either becoming the new primary or by following it. In case the former primary comes back up, the same mechanism will avoid a split-brain by preventing applications from reaching it, running `pg_rewind` on the server and restarting it as a standby. Self-healing is enhanced by the automated recreation of a standby: in case the pod hosting a standby is removed, the operator initiates the procedure to recreate a standby server

The PostgreSQL `Cluster` resource

CloudNativePG

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: clapton
spec:
  instances: 3
  postgresql:
    synchronous:
      method: any
      number: 1
  storage:
    size: 40Gi
  walStorage:
    size: 10Gi
```

EDB Postgres for Kubernetes

```
apiVersion: postgresql.k8s.enterprisedb.io/v1
kind: Cluster
metadata:
  name: clapton
spec:
  instances: 3
  postgresql:
    synchronous:
      method: any
      number: 1
  storage:
    size: 40Gi
  walStorage:
    size: 10Gi
```

IMPORTANT

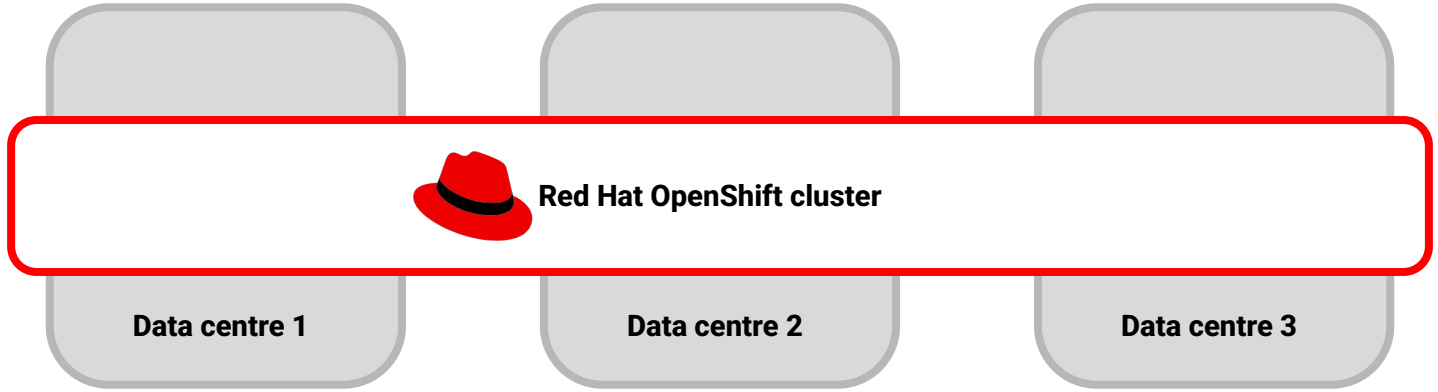
EDB Postgres for Kubernetes is a light fork of CloudNativePG. EDB provides Long Term Support on selected versions.
Differentiation on top of an open core will be in the form of **operands**, **extensions**, **plugins**.

EDB
Postgres® for the AI Generation

Architectures Overview (for Kubernetes)

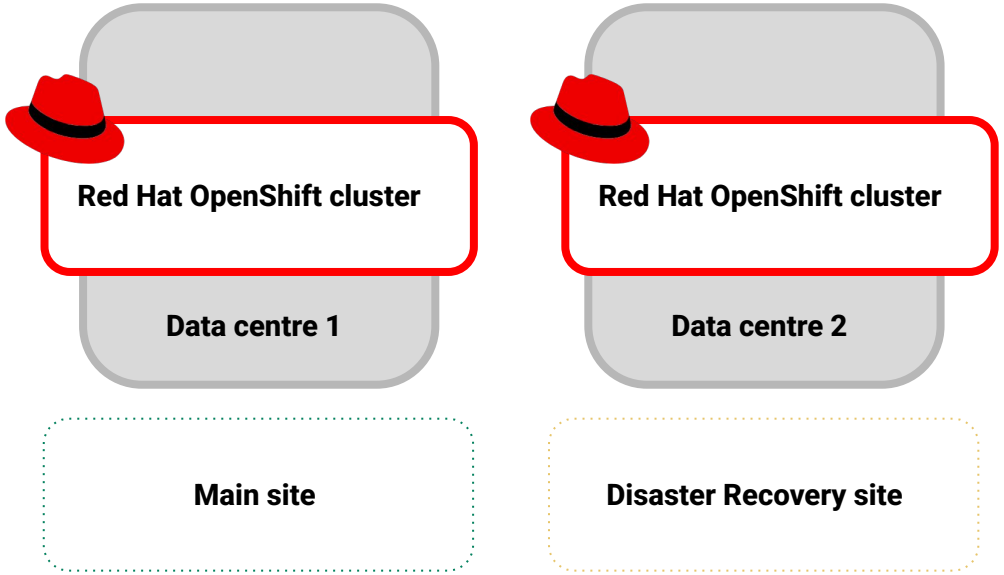
Public Cloud Environment

A typical Red Hat OpenShift “stretched” cluster in cloud environments with 3+ availability zones

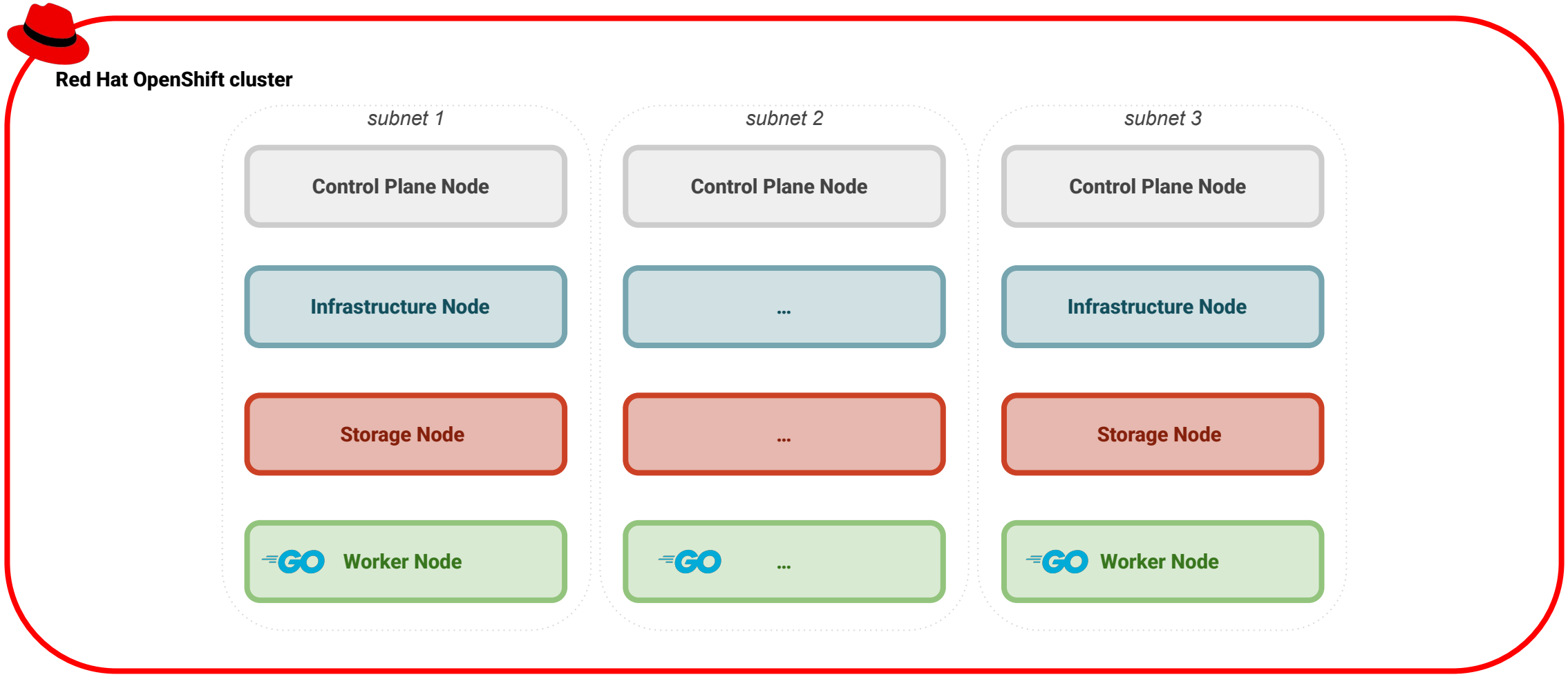


Private Cloud / On-premises Environment

A typical Red Hat OpenShift on-premises deployment with separate clusters for each data centre

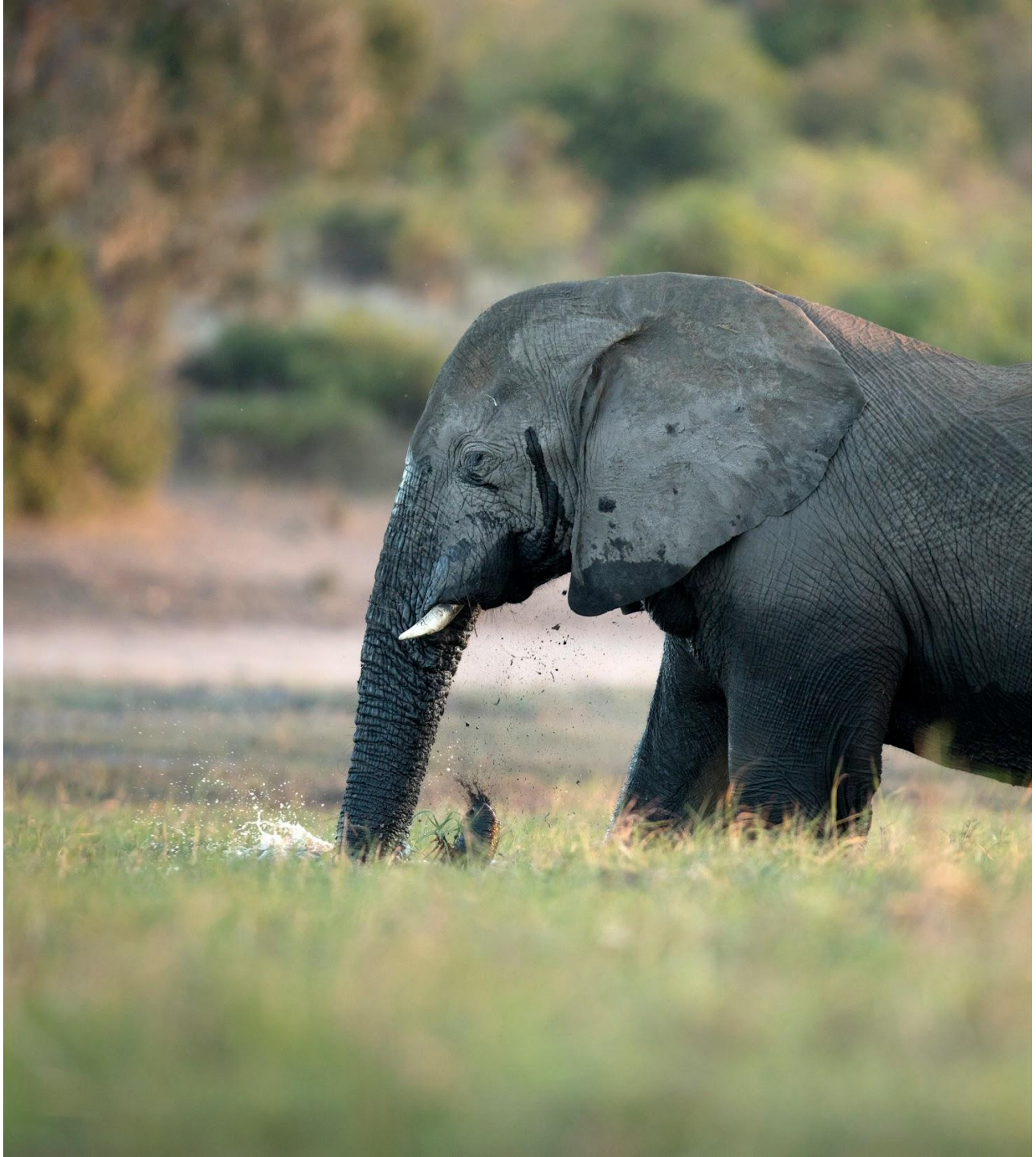


Example of hardware required in a data centre (“cattle approach”)



Kubernetes nodes are typically virtual machines

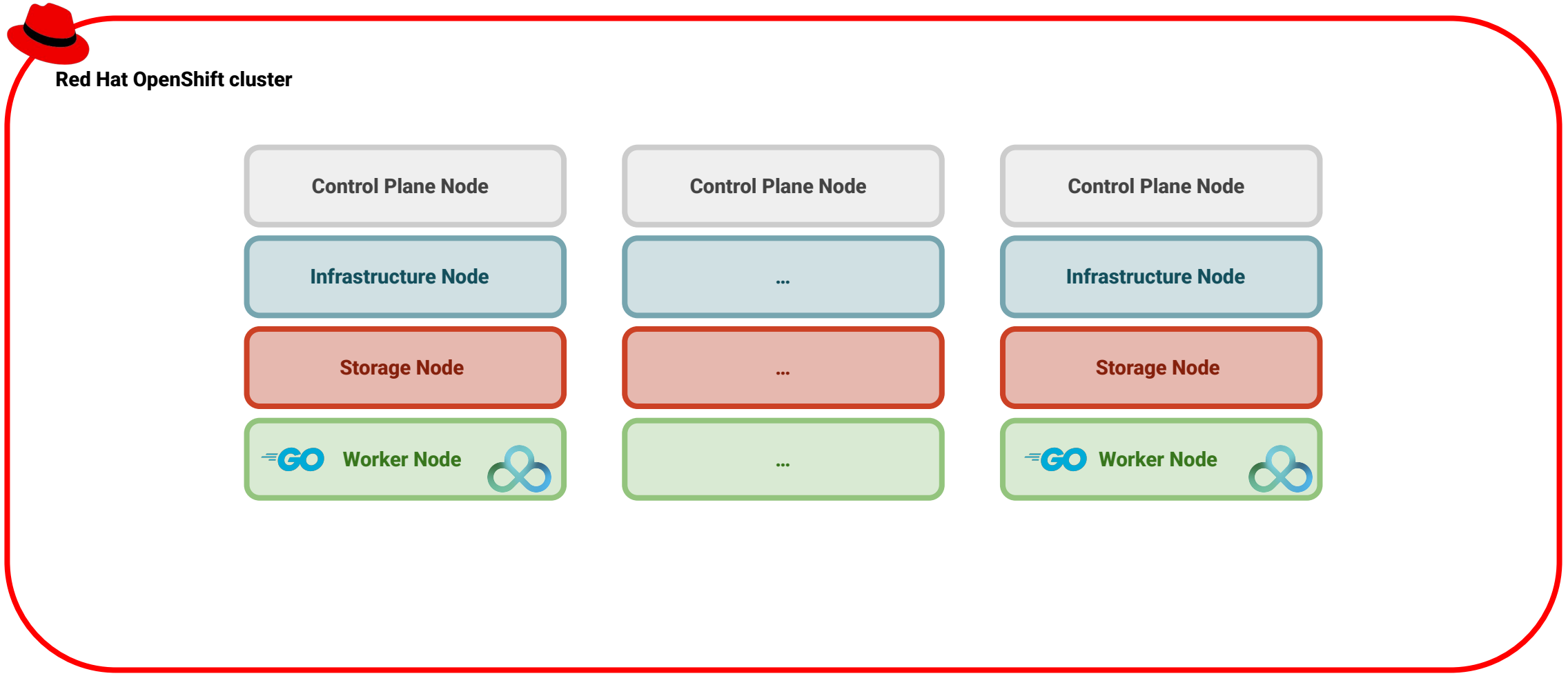
How can we add PostgreSQL in an existing Kubernetes cluster?



EDB
Postgres® for the AI Generation

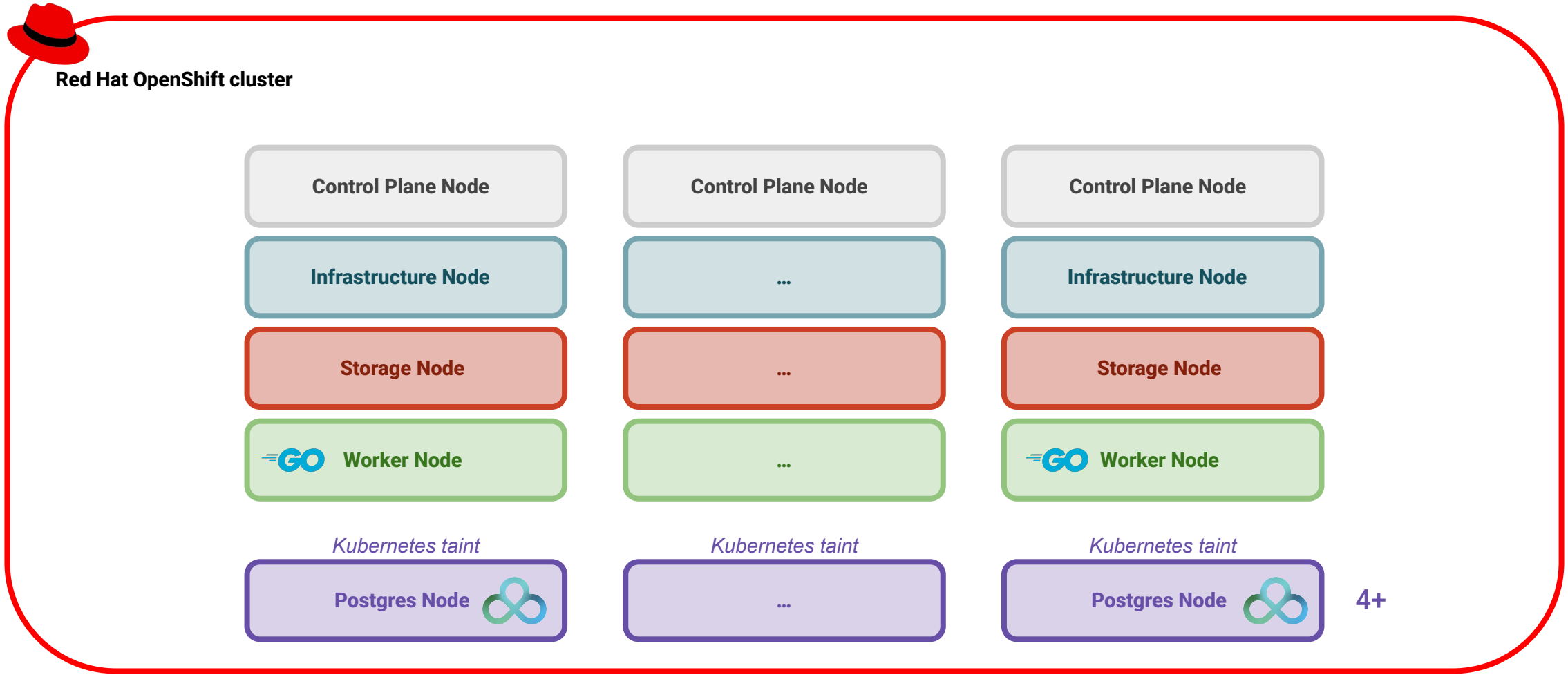
Postgres recommendations (for Kubernetes)

Common perception: shared workloads (“cattle” approach)



Kubernetes nodes are typically virtual machines

Recommendation: isolate Postgres workloads (“elephant herd” approach)



Kubernetes nodes are typically virtual machines

Cattle or **pets**? Better … **herds**

“Pinnawala orphanage provides a lifeline to the orphaned baby elephants and orphaned elephants lost in the wilderness” by Puviraj Diluckshan

From “cattle” to “elephants”: node labels and taints

```
oc label ${node} \
    node-role.kubernetes.io/postgres="" \
    node-role.kubernetes.io/worker-

oc adm taint node ${node} \
    node-role.kubernetes.io/postgres=:NoSchedule
```

Reserving nodes for Postgres workloads

```
# <snip> “Cluster” resource
 affinity:
   enablePodAntiAffinity: true
   topologyKey: kubernetes.io/hostname
   podAntiAffinityType: required
   nodeSelector:
     node-role.kubernetes.io/postgres: ""
   tolerations:
   - key: node-role.kubernetes.io/postgres
     operator: Exists
     effect: NoSchedule
 # <snip>
```

EDB
Postgres® for the AI Generation

The Bare-Metal opportunity

CloudNativePG leverages PostgreSQL's native physical replication to synchronise states across different locations, including cascading and synchronous replication at the transaction level, as well as Hot Standby.

File storage replication in PostgreSQL has gradually become obsolete, starting in 2005 with the introduction of Warm Standby.

Although CloudNativePG is storage-agnostic, **relying on storage replication for PostgreSQL in Kubernetes is considered bad practice.** It also leads to “write-amplification” when used with Postgres replicas.

Allow read only connections during recovery, known as Hot Standby.

Browse files

```
Enabled by recovery_connections = on (default) and forcing archive recovery using a recovery.conf.
Recovery processing now emulates the original transactions as they are replayed, providing full
locking and MVCC behaviour for read only queries. Recovery must enter consistent state before
connections are allowed, so there is a delay, typically short, before connections succeed. Replay of
recovering transactions can conflict and in some cases deadlock with queries during recovery; these
result in query cancellation after max_standby_delay seconds have expired. Infrastructure changes have
minor effects on normal running, though introduce four new types of WAL record.

New test mode "make standbycheck" allows regression tests of static command behaviour on a standby
server while in recovery. Typical and extreme dynamic behaviours have been checked via code inspection
and manual testing. Few port specific behaviours have been utilised, though primary testing has been
on Linux only so far.

This commit is the basic patch. Additional changes will follow in this release to enhance some aspects
of behaviour, notably improved handling of conflicts, deadlock detection and query cancellation.
Changes to VACUUM FULL are also required.

Simon Riggs, with significant and lengthy review by Heikki Linnakangas, including streamlined redesign
of snapshot creation and two-phase commit.

Important contributions from Florian Pflug, Mark Kirkwood, Merlin Moncure, Greg Stark, Gianni Ciolli,
Gabriele Bartolini, Hannu Krosing, Robert Haas, Tatsuo Ishii, Hiroyuki Yamada plus support and
feedback from many other community members.
```

master

REL_17_BETA2 ... REL8_5_ALPHA3

simonat2ndQuadrant committed on Dec 19, 2009

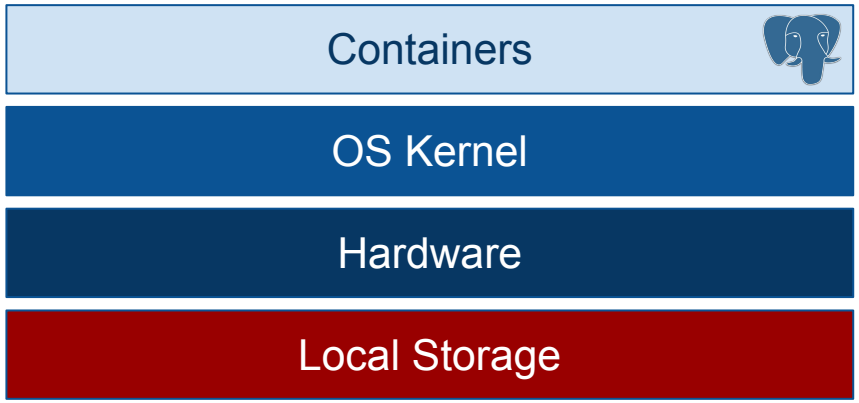
1 parent 78a0914 commit efc16ea

EDB

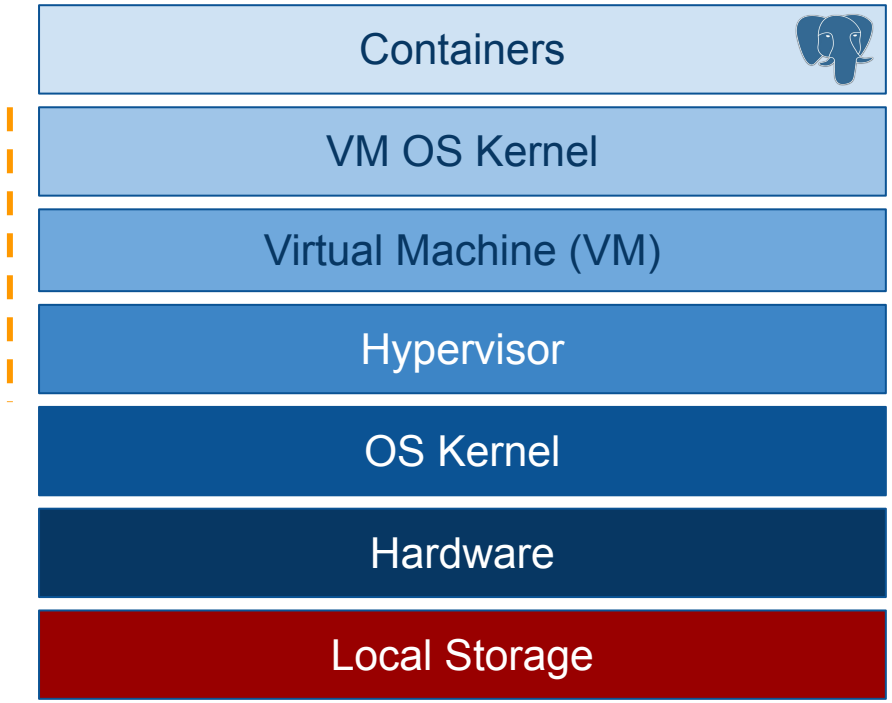
You can run Kubernetes on bare metal nodes

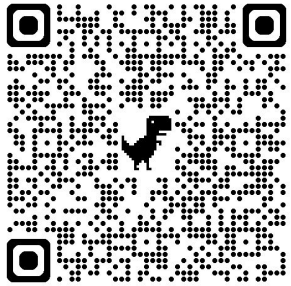
With locally attached and dedicated storage. Migrating “Postgres on VMs” to CloudNativePG on bare metal Kubernetes nodes.

Bare metal



VMs





Installing on bare metal

OPENSHIFT CONTAINER PLATFORM 4.18

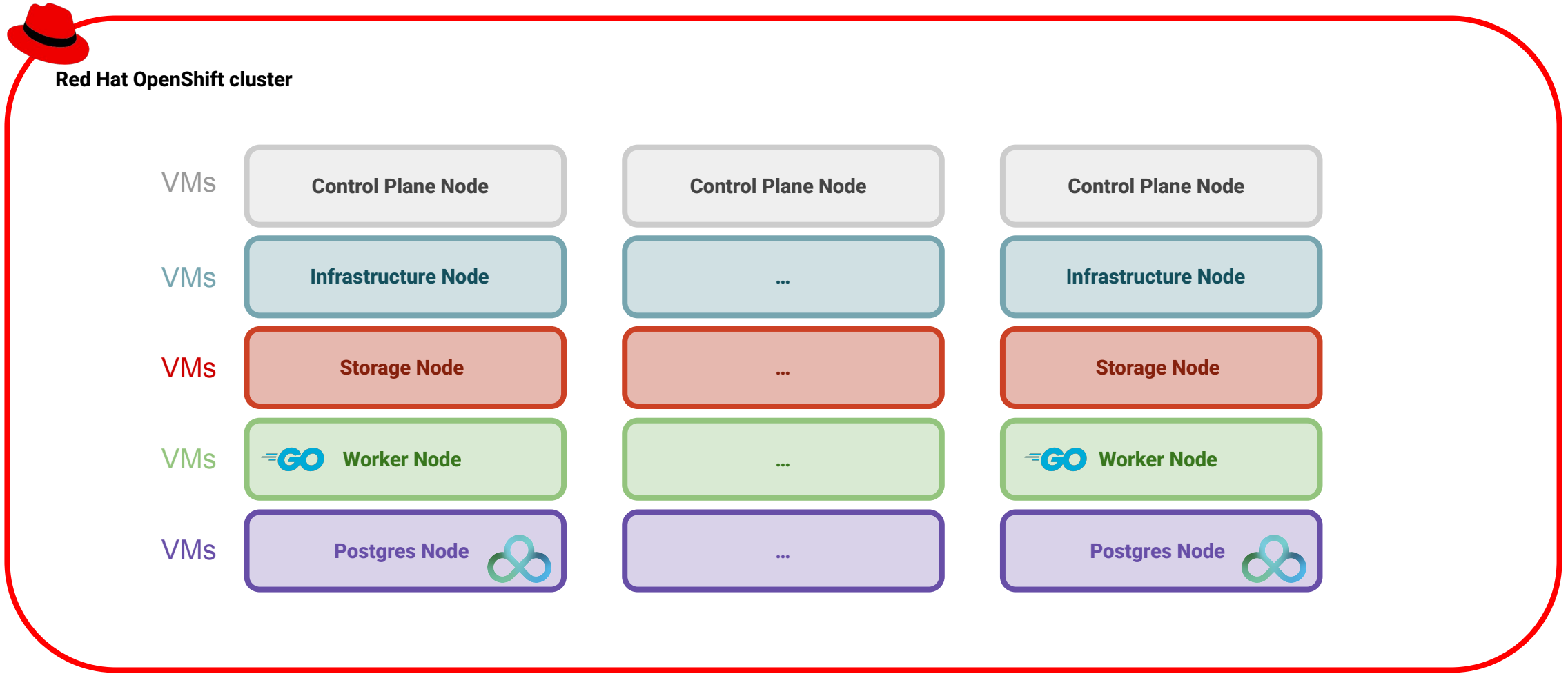
Installing OpenShift Container Platform on bare metal

Red Hat OpenShift Documentation Team

Abstract

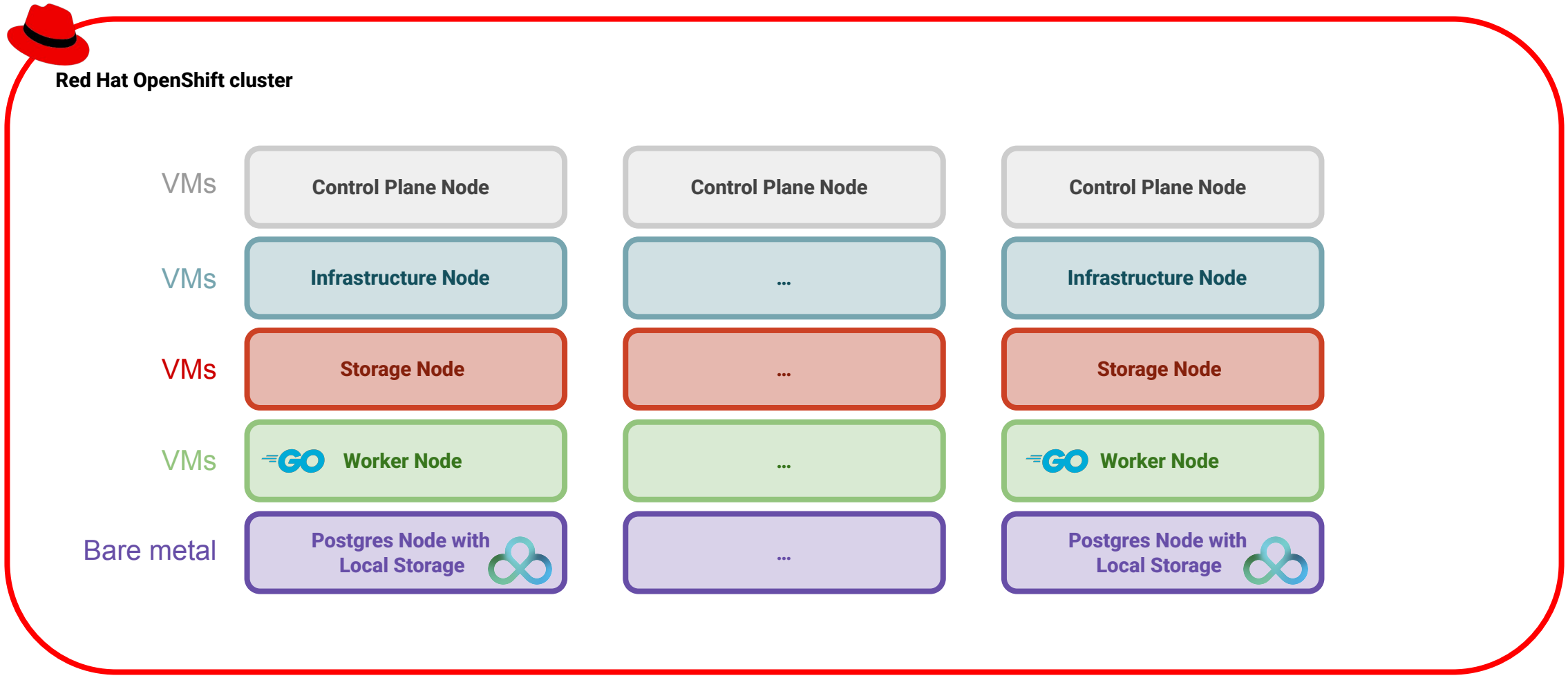
This document describes how to install OpenShift Container Platform on bare metal.

Basic approach: use VMs for Postgres nodes (dedicated compute, shared storage)



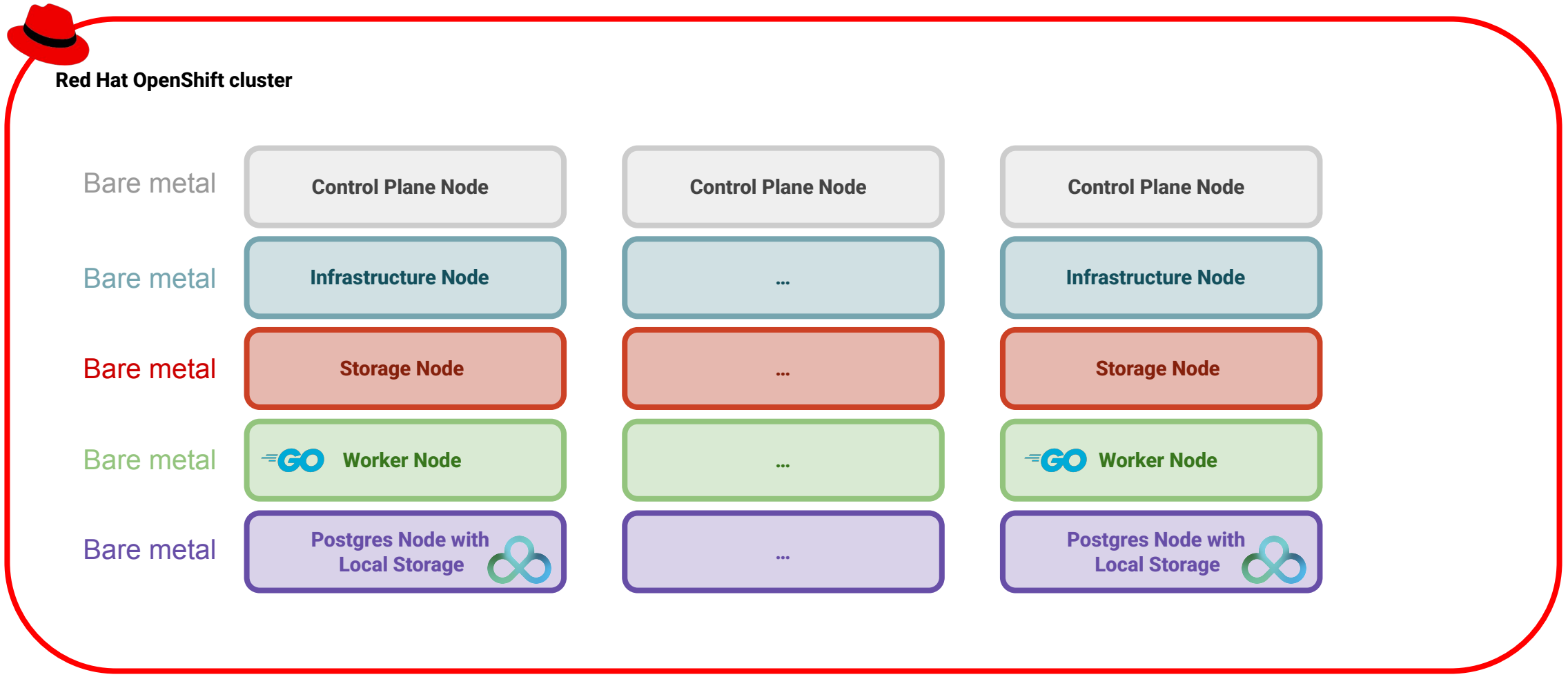
More resources, more VMs, more licenses

Recommended approach: shared-nothing architecture



Mindshift required, but better predictability, stability, durability, scalability

Recommended approach: full bare-metal with shared-nothing architecture for Postgres



Use OpenShift to consolidate your workloads, without an underlying hypervisor

4.12. Persistent storage using local storage

4.12.1. Local storage overview

You can use any of the following solutions to provision local storage:

- HostPath Provisioner (HPP)
- Local Storage Operator (LSO)
- Logical Volume Manager (LVM) Storage

Warning

These solutions support provisioning only node-local storage. The workloads are bound to the nodes that provide the storage. If the node becomes unavailable, the workload also becomes unavailable. To maintain workload availability despite node failures, you must ensure storage data replication through active or passive replication mechanisms.

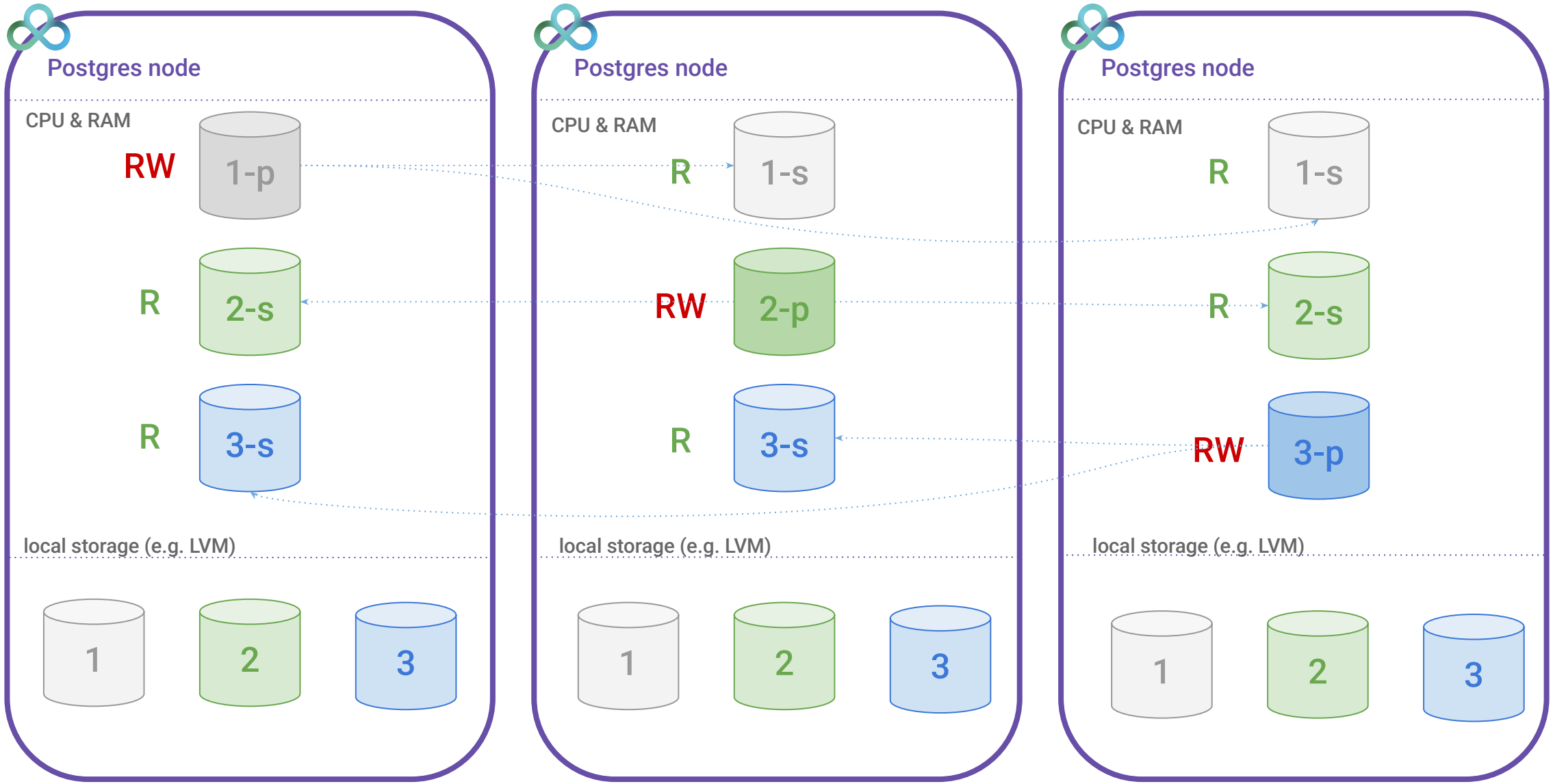
4.12.1.1. Overview of HostPath Provisioner functionality

You can perform the following actions using HostPath Provisioner (HPP):

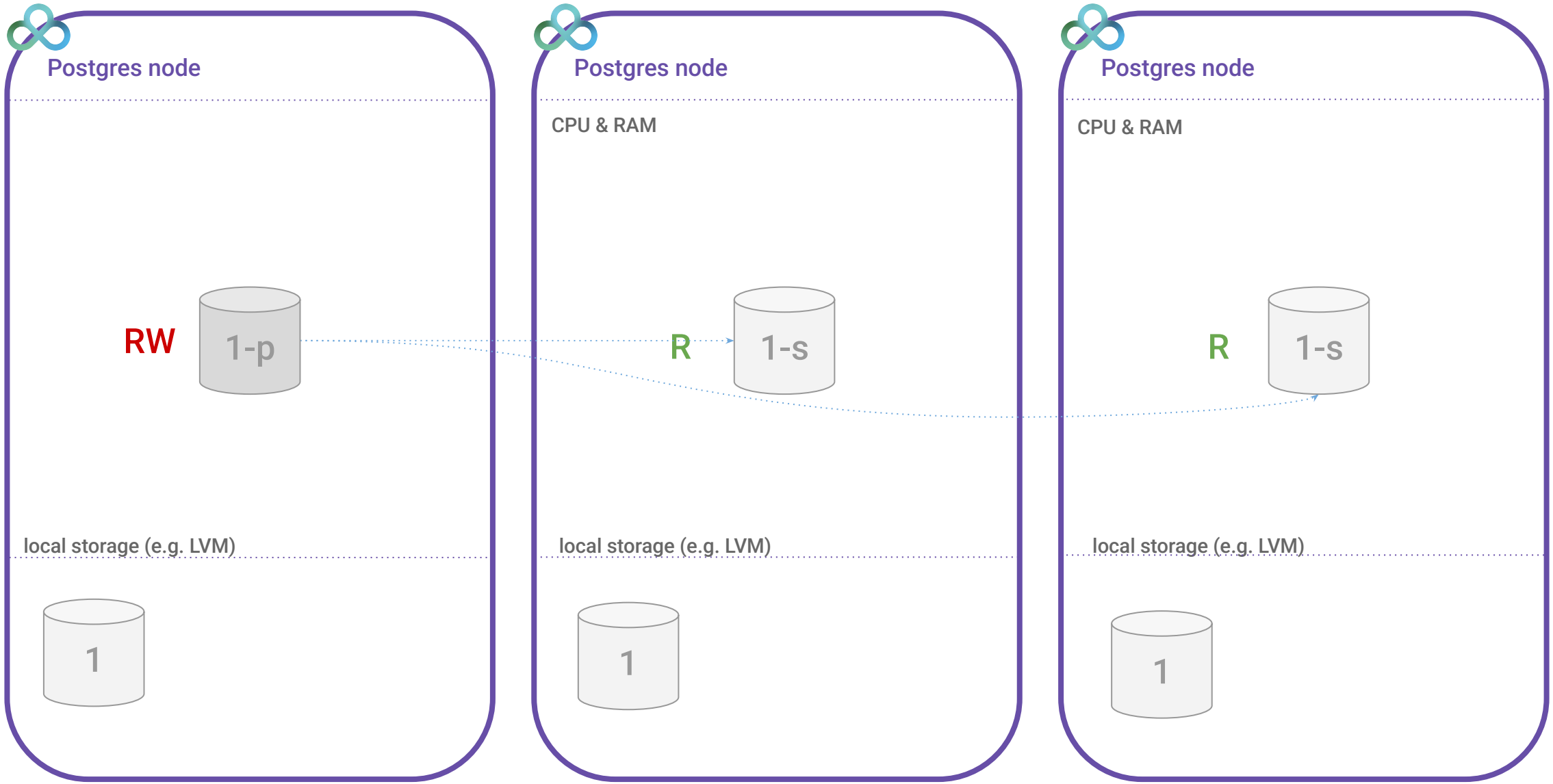
- Map the host filesystem paths to storage classes for provisioning local storage.
- Statically create storage classes to configure filesystem paths on a node for storage consumption.
- Statically provision Persistent Volumes (PVs) based on the storage class.

The actual scheduling order might change

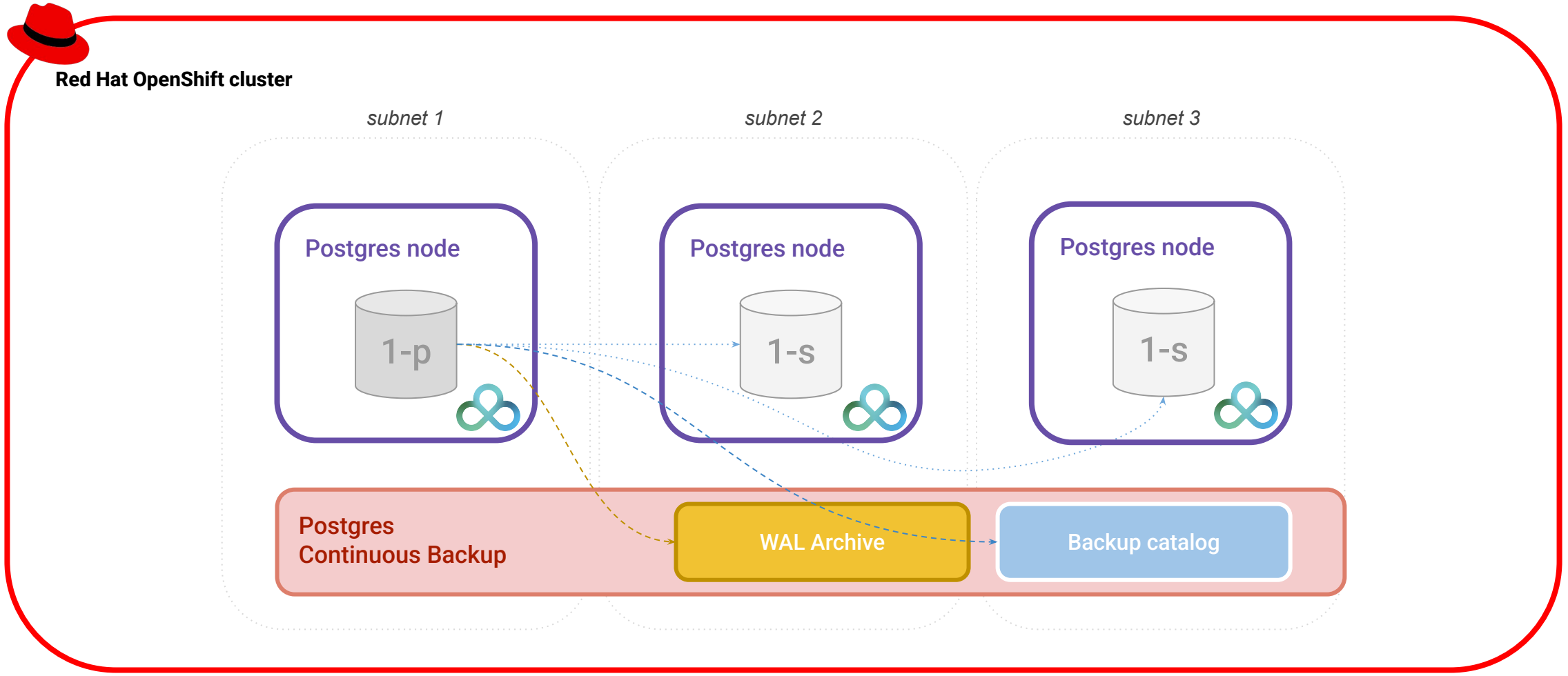
How CloudNativePG uses Postgres worker nodes



Let’s examine in more detail a single Postgres cluster lifecycle and architecture

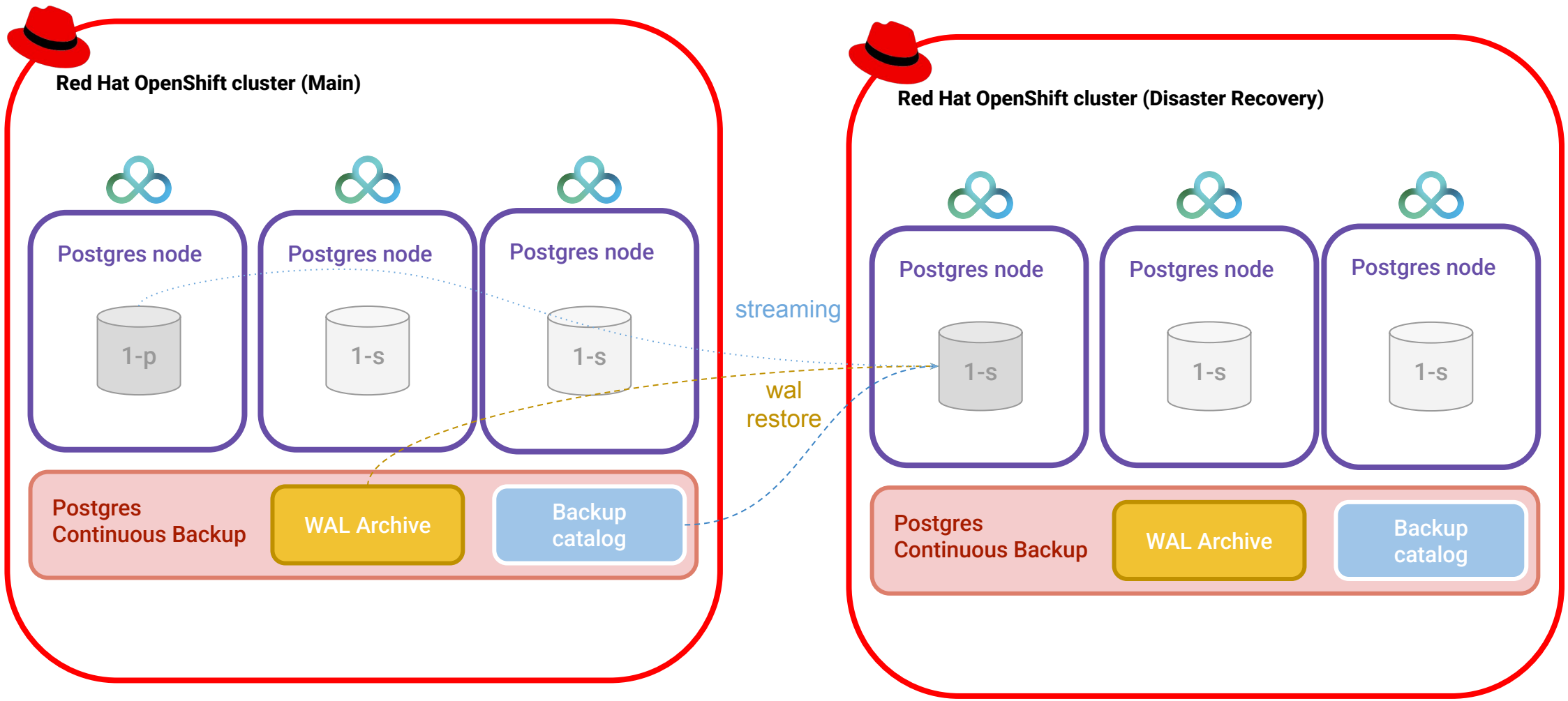


Let’s examine in more detail a single Postgres cluster lifecycle and architecture



Full HA and DR within a Red Hat OpenShift Cluster

How Disaster Recovery across two Red Hat OpenShift clusters is addressed



Declarative demotion, Declarative promotion (no automated failover)

EDB
Postgres® for the AI Generation

Migrating Postgres (to Kubernetes)

Moving Postgres to Kubernetes

It is not a lift-and-shift transition, it requires a mindshift

- **Break the silos**
  - Multi-disciplinary team with dev, infrastructure and DBA SMEs
  - You need Kubernetes and Postgres skills in your broader team
    - **That's where Red Hat and EDB can accelerate your process and reduce risks**
- **Application and Postgres database are in the same Kubernetes cluster**
  - Possibly in the same namespace
  - Separate Kubernetes clusters for databases are common (sigh!)
    - Silo culture "devs vs ops" no security gains, more operational complexity
- **Start with a pilot project**
  - Start with "cattle" approach, if the "elephant" is not possible
- **Based on success of the project, start planning for the "elephant" approach**
  - Isolate Postgres nodes, consider bare metal and local storage

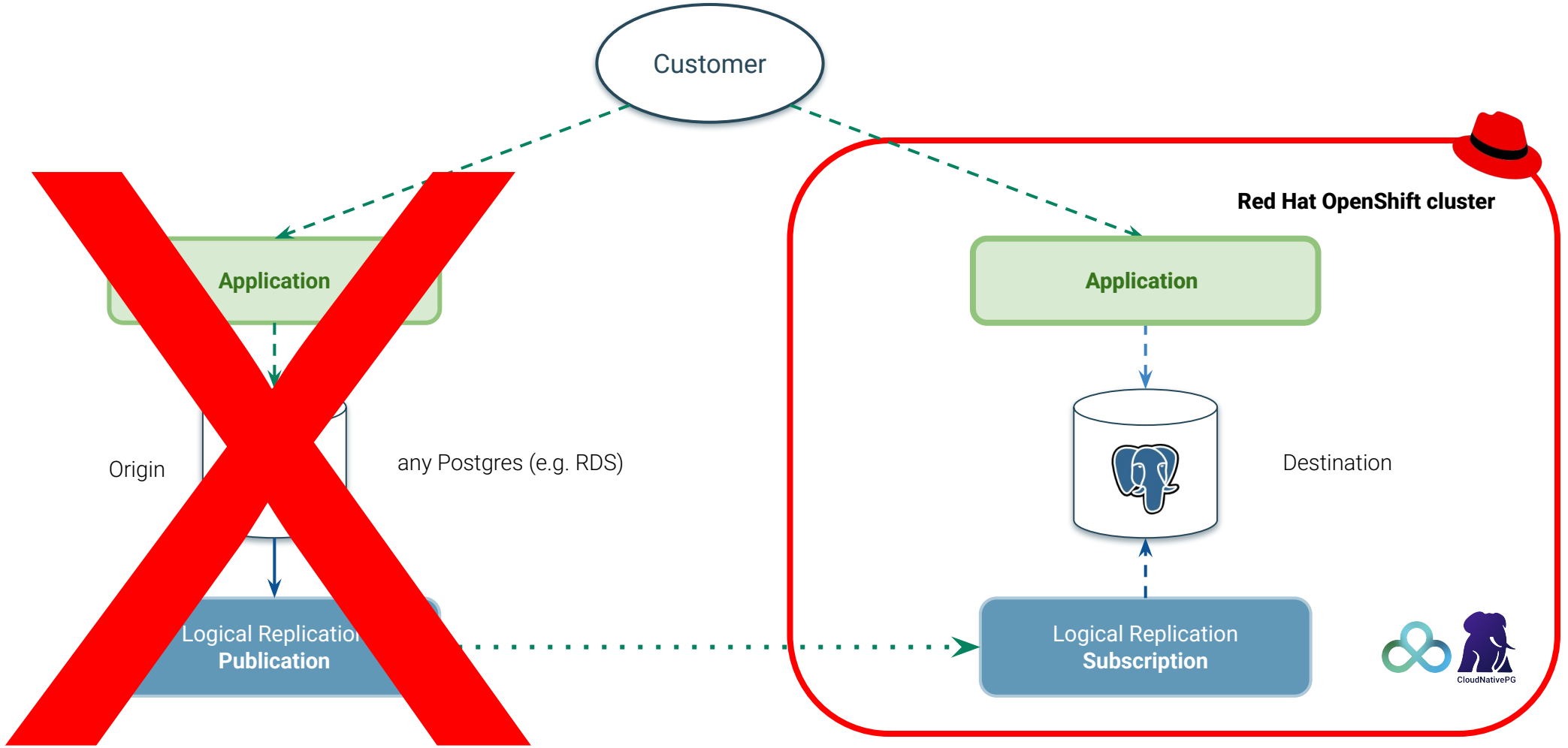
Benchmarking

Prior to production, both the storage and the database must be benchmarked

- Storage:
  - fio (Flexible I/O tester)
  - Shortcut: “oc cnpg fio”
    - Creates a Kubernetes job that benchmarks the storage class
- Database:
  - pgbench
  - Shortcut: “oc cnpg pgbench”, fully customisable
    - Creates a Kubernetes job that runs pgbench with custom options
- Examples:
  - github.com/gbartolini/postgres-kubernetes-playground/tree/main/tablespaces

~Zero cutover migrations from any Postgres anywhere

Use Postgres native logical replication and CloudNativePG declarative subscriptions for blue/green migrations with ~0 downtime



EDB
Postgres® for the AI Generation

Key takeaways

Reference architecture for Red Hat OpenShift

EDB + Red Hat

High Availability & Disaster Recovery of PostgreSQL Databases with **EDB and Red Hat OpenShift**

Reference architecture for a single Red Hat OpenShift cluster

enterprisedb.com

2. **Isolate PostgreSQL workloads:** Dedicate nodes specifically for PostgreSQL workloads, separating them from other applications. Use Red Hat OpenShift features such as node labels, selectors, taints, and tolerations to enforce this separation through declarative configuration. Reserve at least three nodes for PostgreSQL in each Red Hat OpenShift cluster, distributed evenly across the availability zones/subnets. Scale in multiples of three to maintain balance and resilience. In some cases, you might dedicate three Red Hat OpenShift worker nodes to a single PostgreSQL cluster, each hosting a PostgreSQL instance (primary or replica).
3. **Trust PostgreSQL replication over storage replication:** Unlike many cloud-native applications that synchronize state at the storage level, PostgreSQL handles state synchronization independently through its built-in physical replication capabilities, based on write-ahead log (WAL) shipping. These capabilities, used successfully in production by millions of users worldwide for more than a decade, include asynchronous and synchronous streaming replication over the network, as well as asynchronous file-based log shipping, typically used as a fallback option (e.g., storing WAL files in an object store). Standby servers, or replicas, can also handle read-only workloads thanks to the hot standby feature.

The result is a complete and transparent physical separation of PostgreSQL workloads from other applications, with dedicated nodes for PostgreSQL, termed "Postgres nodes." These nodes can be easily added to existing Red Hat OpenShift clusters and scaled up as needed. This setup is illustrated in the following figure:

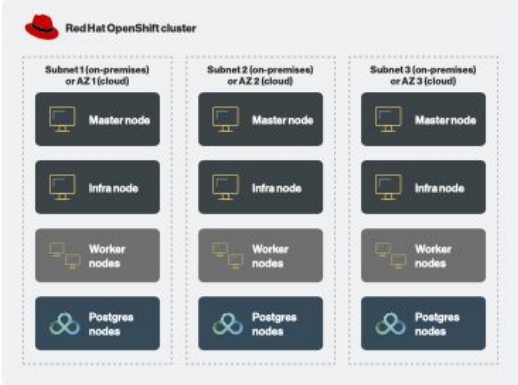


Figure 4. The four primary types of nodes in a typical EDB PG4K deployment on Red Hat OpenShift

To designate specific worker nodes as PostgreSQL nodes within your OpenShift cluster, we recommend using the `node-role.kubernetes.io/postgres` label. This label helps identify the nodes that should handle PostgreSQL workloads. You can apply the label to your chosen nodes using the following command:

```
$ oc label node <node-name> node-role.kubernetes.io/postgres=""
```

You can set appropriate taints on these nodes to ensure that only PostgreSQL workloads are scheduled on your designated Postgres nodes. This will prevent other types of workloads from being scheduled on them unless they have the matching tolerations. For example, you can add the following taint to a PostgreSQL node:

```
$ oc adm taint nodes <node-name> node-role.kubernetes.io/postgres=:NoSchedule
```

By applying this taint, the node will only accept pods with a corresponding toleration, limiting it to PostgreSQL workloads. When creating them, remember to set the proper tolerations on your PostgreSQL clusters so they can be scheduled on the tainted nodes. This approach enhances resource isolation and ensures that your PostgreSQL workloads run in the most suitable environment.

By design, in the above case, every committed transaction in the database is guaranteed to be written to a replica using PostgreSQL's native quorum-based synchronous replication, ensuring an RPO of zero for high availability. Given that Red Hat OpenShift can immediately detect a failure on a primary, the failover time is typically just the time it takes for the most advanced replica to be promoted to primary status, usually within a minute in total. This makes achieving 99.99% uptime a realistic goal, given a solid underlying infrastructure.

Regarding disaster recovery, EDB PG4K allows you to set up continuous backups of a PostgreSQL cluster through base backups and the WAL archive, ensuring, by default, a maximum RPO of five minutes.

In the simplest scenario, both components can reside in a local object store, either using the native provider solution if you are in the cloud (e.g., AWS S3, Azure Blob Storage, Google Cloud Storage) or a specialized product such as Red Hat OpenShift Data Foundation (ODF) object storage. If your storage class supports them, EDB PG4K also allows you to perform backup and recovery using volume snapshots. Hybrid strategies with both object stores and volume snapshots having different retention policies are also possible.

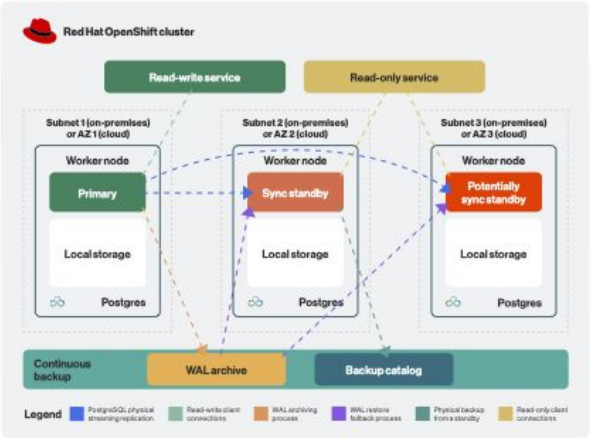
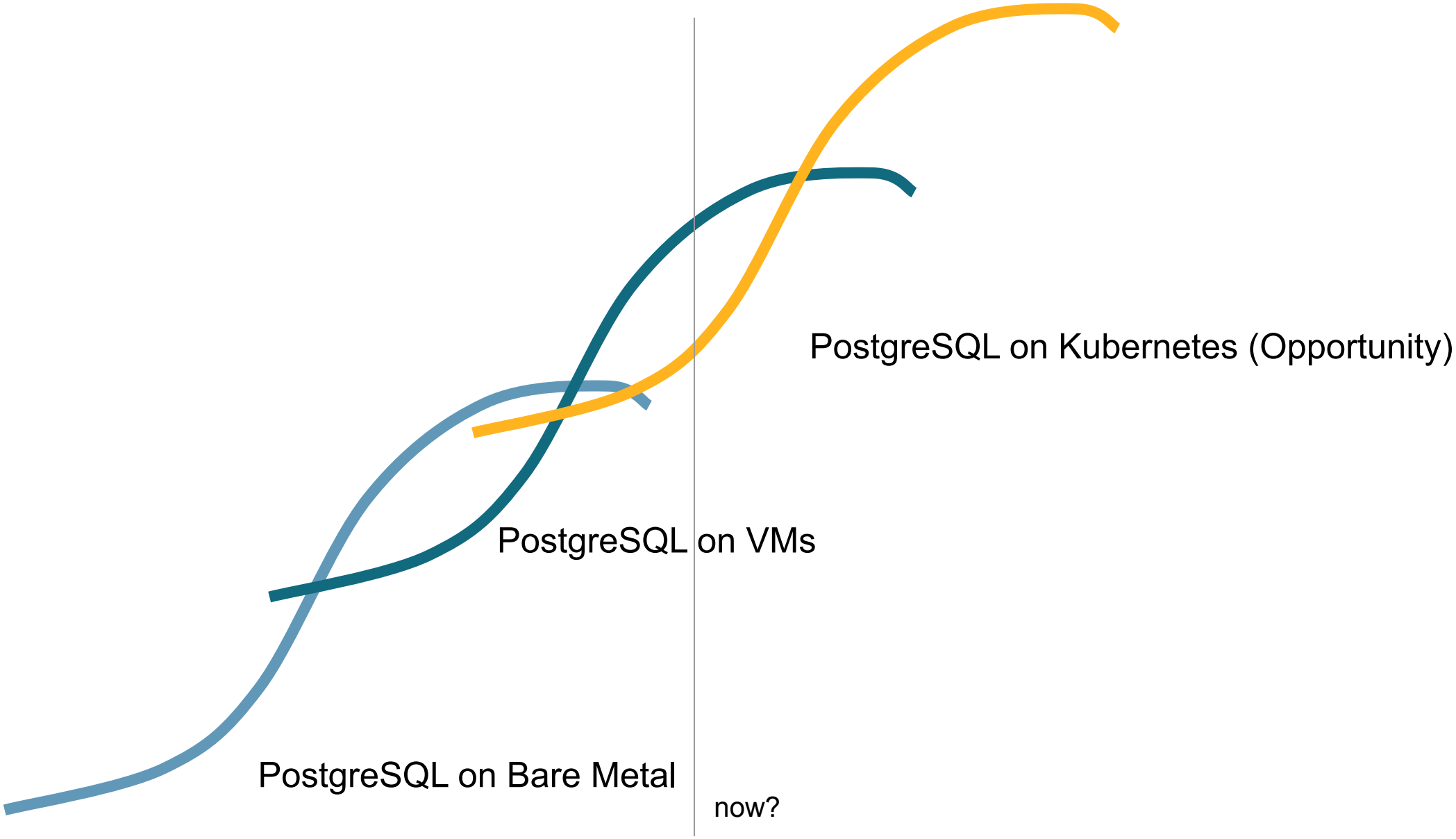


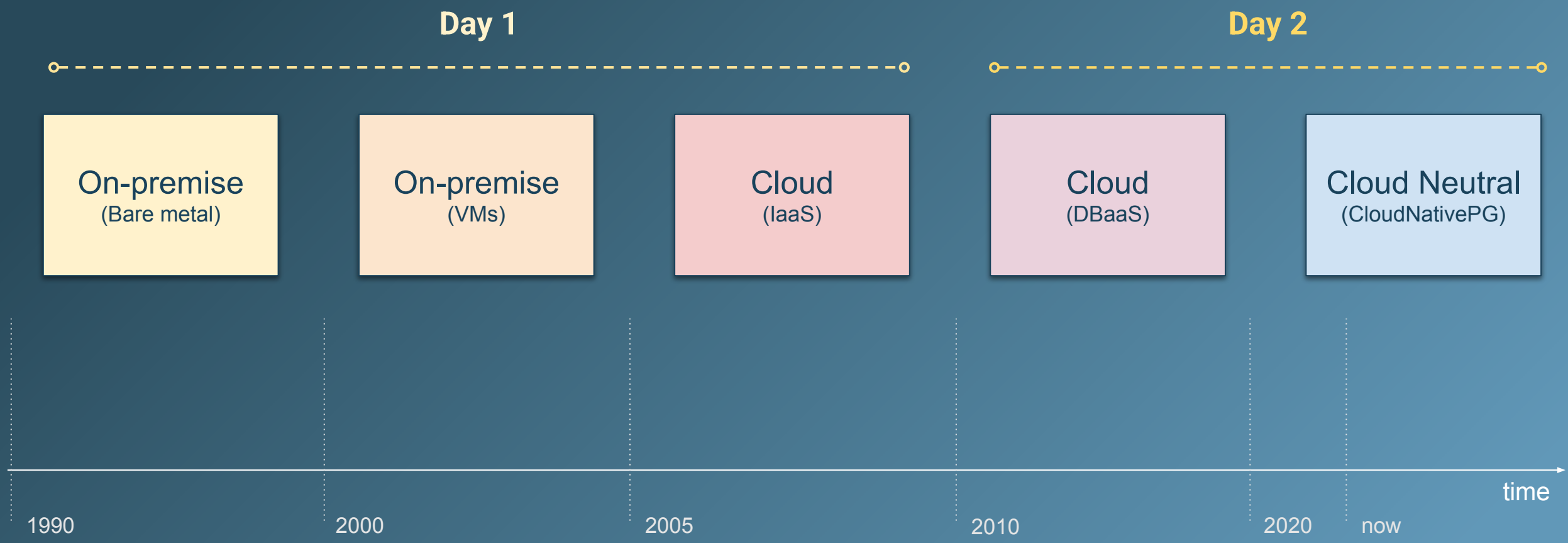
Figure 6. Typical architecture of a high-availability PostgreSQL cluster, highlighting services and continuous backup



EDB

The evolution of Postgres use cases

From Handcrafted PostgreSQL to Cloud-Neutral Automation with GitOps & K8s



Suggested reading from the CNCF blog

Cloud Neutral Postgres Databases with Kubernetes and CloudNativePG

Three options, your choice

Choose what works best for you, based on how critical Postgres is

- **Shared approach: Postgres like any other workload (“cattle”)**
  - Run Postgres on any worker node, with shared storage
- **Mixed approach: Dedicated compute, shared storage (“elephants”)**
  - Run Postgres on dedicated nodes, with shared storage
- **Shared-nothing approach: Dedicated compute and storage (“elephants”)**
  - Run Postgres on dedicated nodes with dedicated storage
    - Local storage on bare metal Postgres nodes
- **Reduce VM license costs with Red Hat OpenShift on bare-metal**
  - Planning is required due to hardware procurement
  - Mix VMs and bare-metal nodes to suit your needs
  - Consider local storage with NVMe RAID options too

CLOUD NATIVE DAYS ITALY

We’re proud sponsors of Cloud Native Days Italy

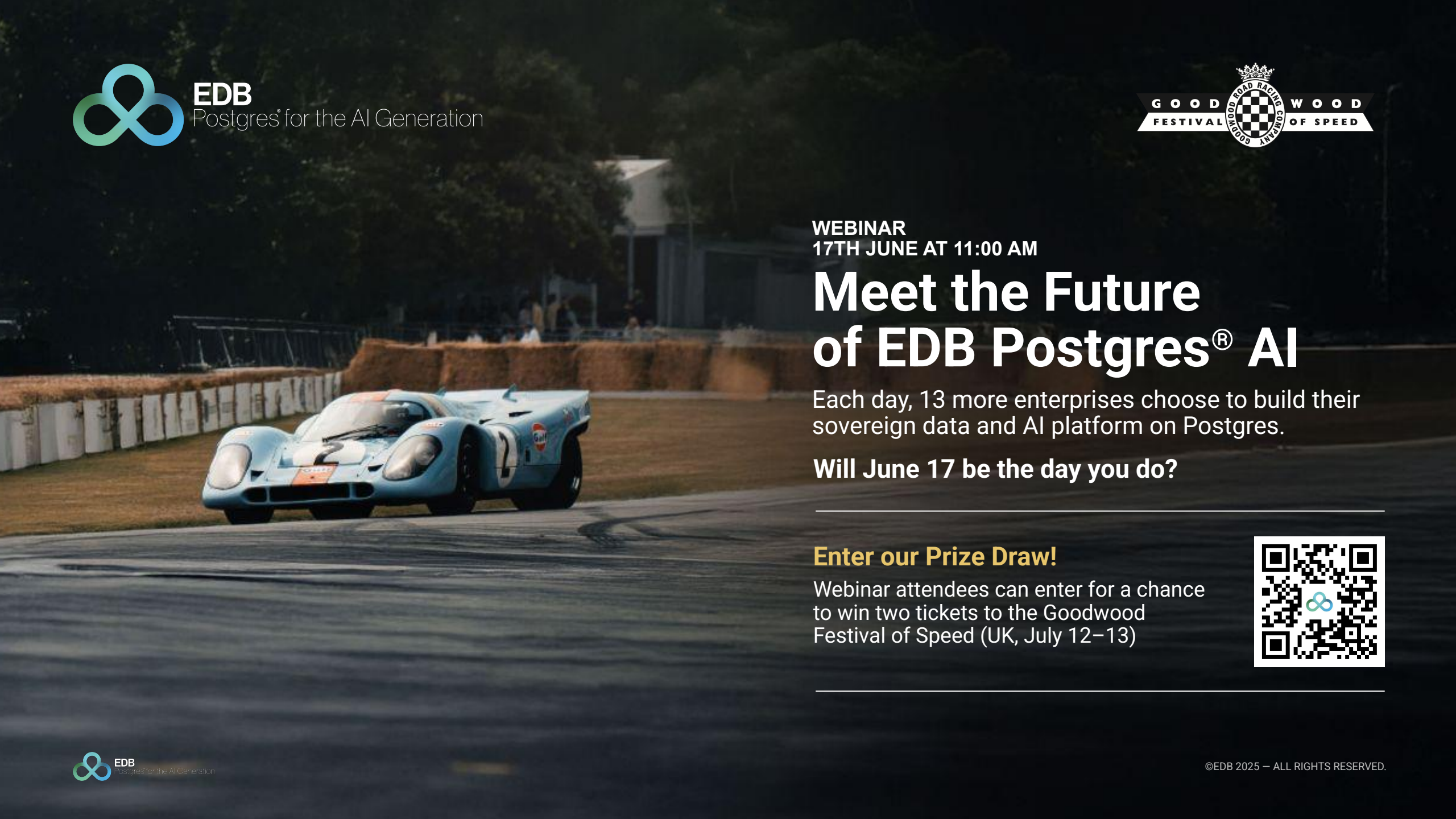
cloudnativedaysitaly.org

June 23-24

Gabriele Bartolini

Jonathan Battiato

Jonathan Gonzalez



EDB
Postgres® for the AI Generation

GOODWOOD FESTIVAL OF SPEED

WEBINAR
17TH JUNE AT 11:00 AM

Meet the Future of EDB Postgres® AI

Each day, 13 more enterprises choose to build their sovereign data and AI platform on Postgres.

Will June 17 be the day you do?

Enter our Prize Draw!

Webinar attendees can enter for a chance to win two tickets to the Goodwood Festival of Speed (UK, July 12–13)

Questions?

gabrielebartolini.it

@_GBartolini_

Creators of  CloudNativePG

enterprisedb.com

cloudnative-pg.io

CNCF Slack

@CloudNativePG@mastodon.social

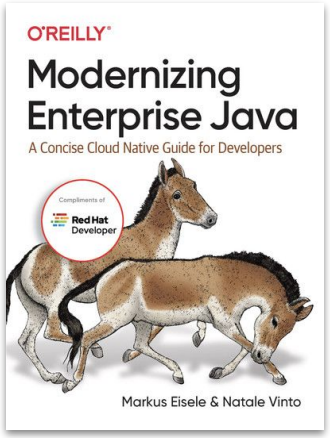
@CloudNativePG.bsky.social

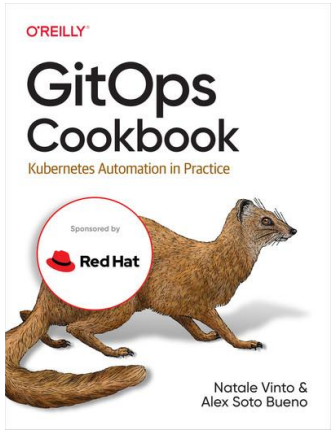
Red Hat OpenShift

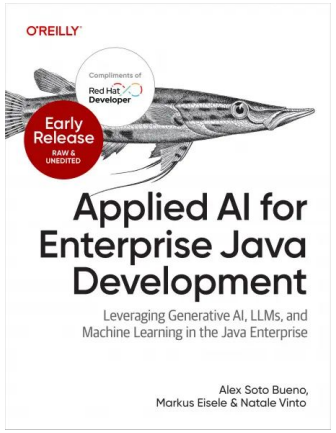
Kubernetes in modern IT infrastructures

Natale Vinto
Technical Evangelism Director, Red Hat
@natalevinto@mastodon.uno

Red Hat


O'REILLY
Modernizing Enterprise Java
A Concise Cloud Native Guide for Developers
Compliments of
Red Hat Developer
Markus Eisele & Natale Vinto


O'REILLY
GitOps Cookbook
Kubernetes Automation in Practice
Sponsored by
Red Hat
Natale Vinto &
Alex Soto Bueno


O'REILLY
Compliments of
Red Hat Developer
Early Release
RAW & UNEDITED
Applied AI for Enterprise Java Development
Leveraging Generative AI, LLMs, and Machine Learning in the Java Enterprise
Alex Soto Bueno,
Markus Eisele & Natale Vinto


PROGRAM COMMITTEE MEMBER
2021

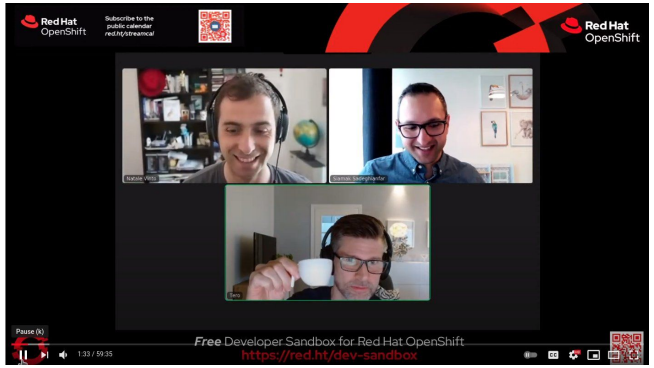

PROGRAM COMMITTEE MEMBER
2022


PROGRAM COMMITTEE MEMBER
2023


CD.FOUNDATION
AMBASSADOR
2023


KCD
GROWING CLOUD NATIVE TOGETHER
ORGANIZER
2024


RED HAT
CERTIFIED
ENGINEER
Red Hat

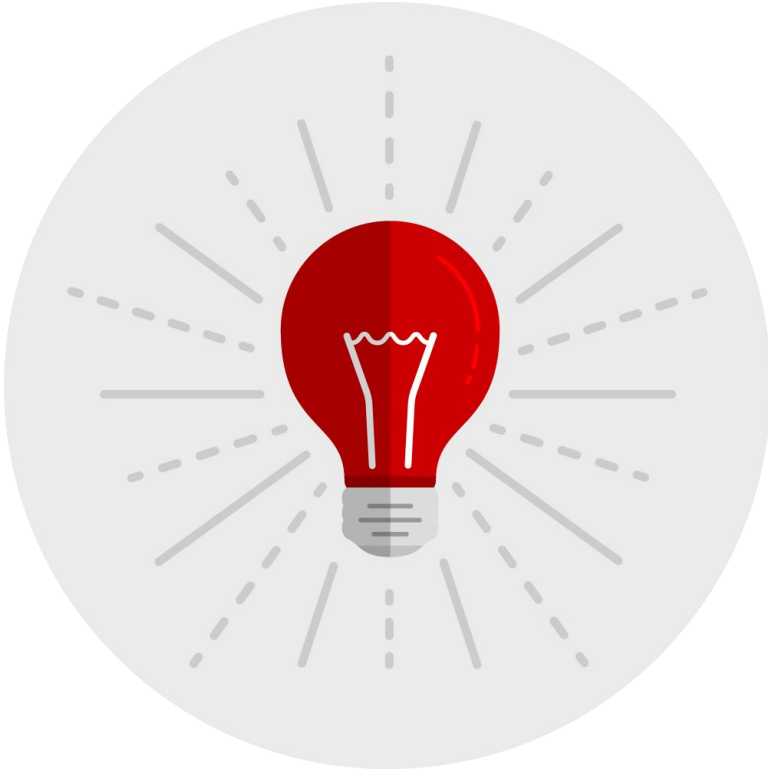

Red Hat OpenShift
Subscribe to the public calendar
Red Hat OpenShift
Free Developer Sandbox for Red Hat OpenShift
https://red.ht/dev-sandbox


KubeCon
CloudNativeCon
Europe 2024
SPEAKER
2024


KubeCon
CloudNativeCon
Europe 2025
SPEAKER
2025

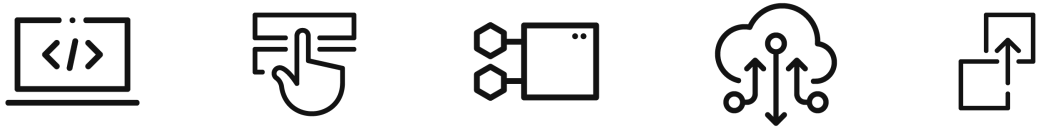
Agenda

- Why Kubernetes
- Kubernetes & Hybrid Cloud
- Red Hat OpenShift



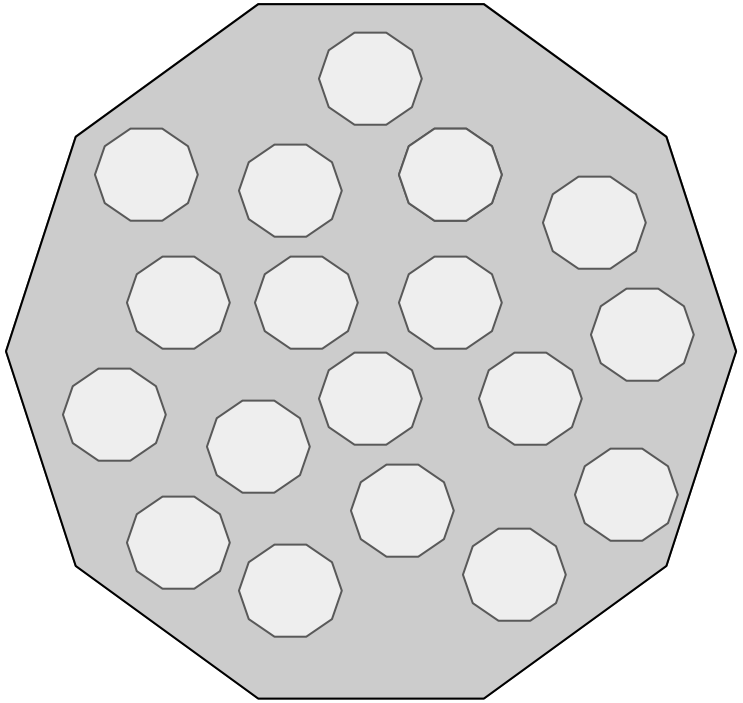
Red Hat

Why Kubernetes

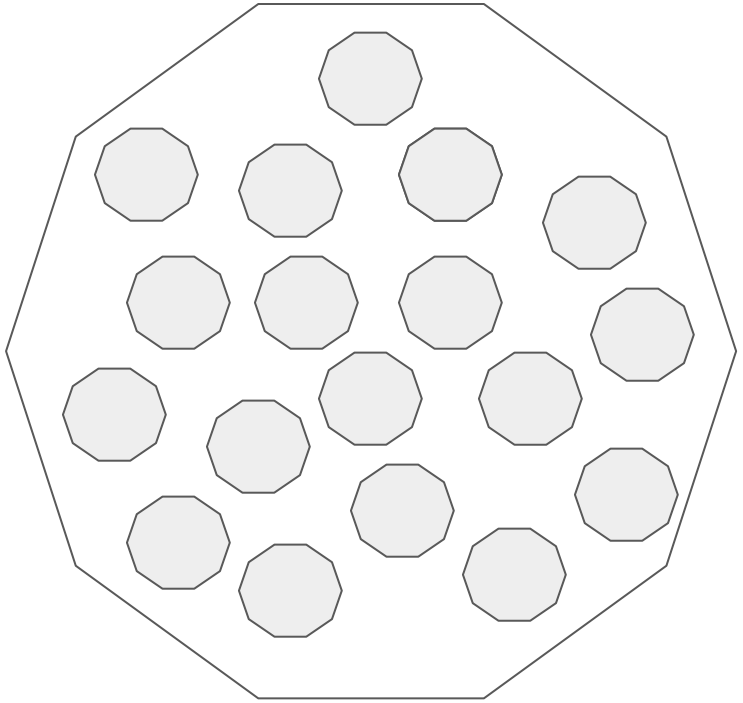


Red Hat

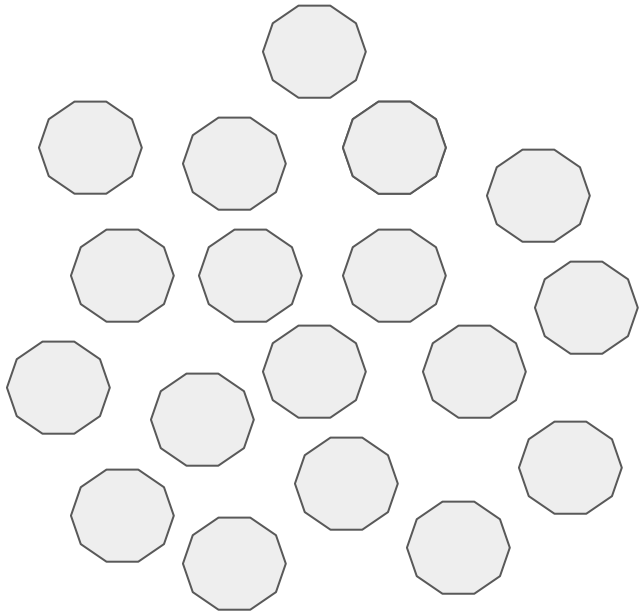
The Application



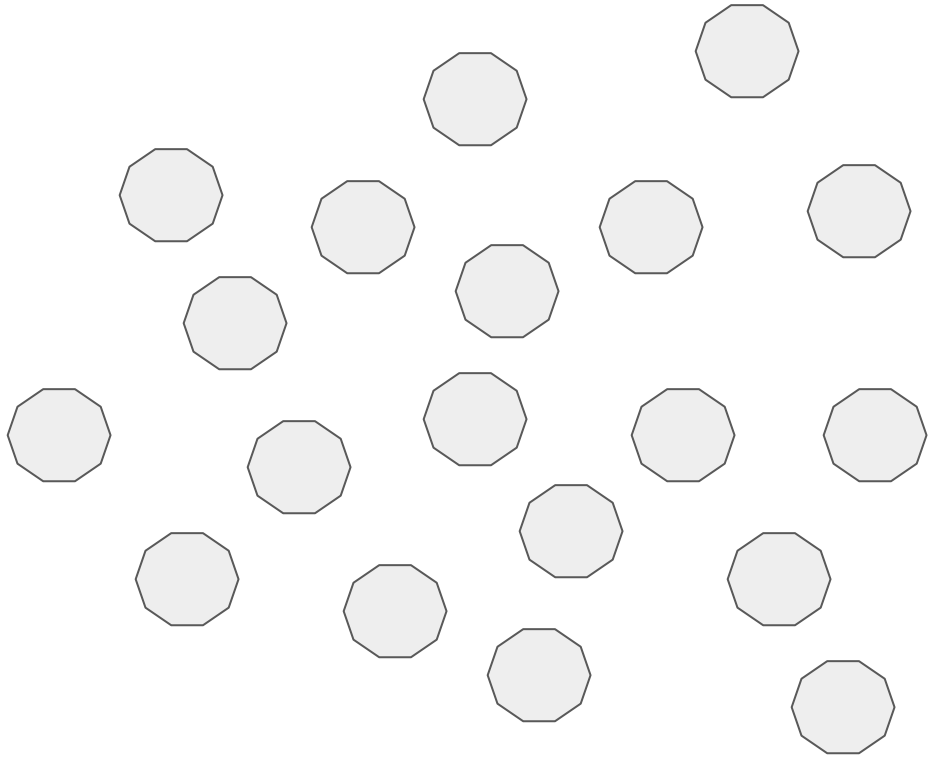
Modules



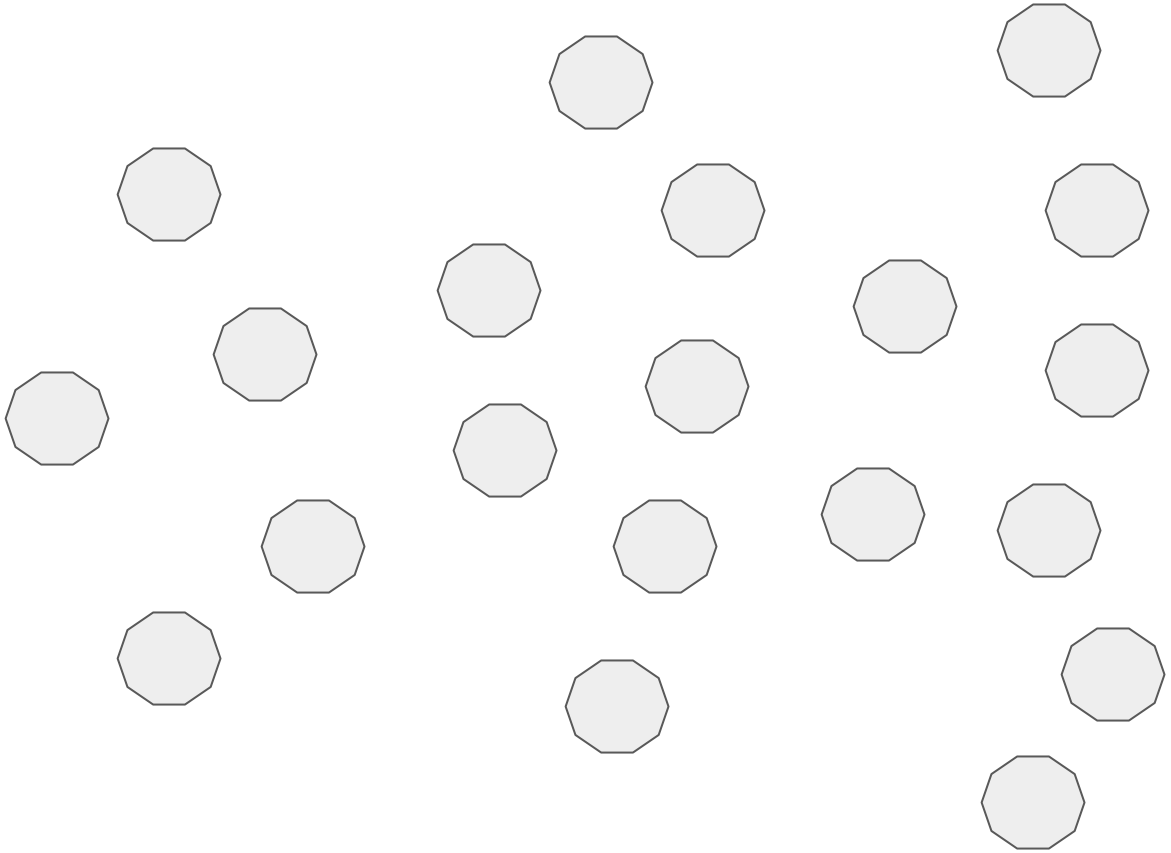
Microservices



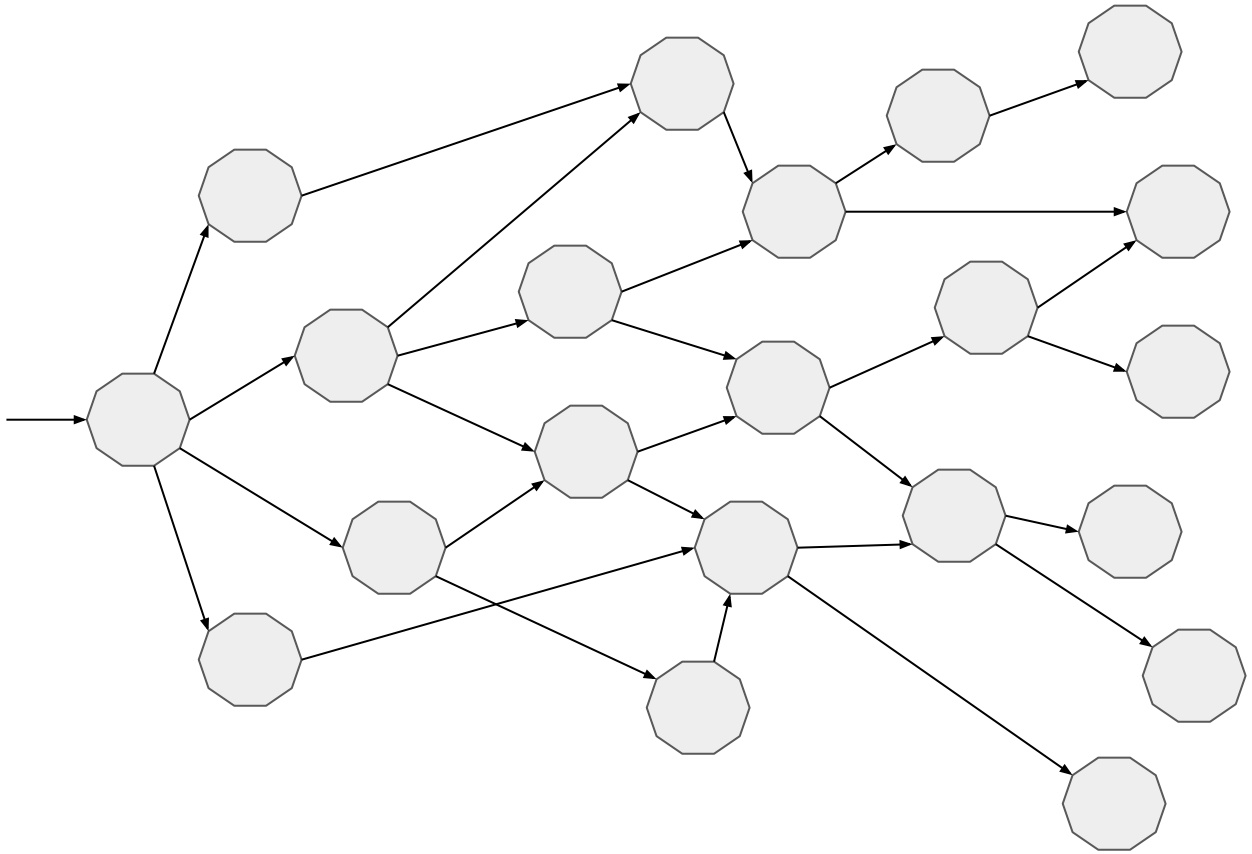
Microservices



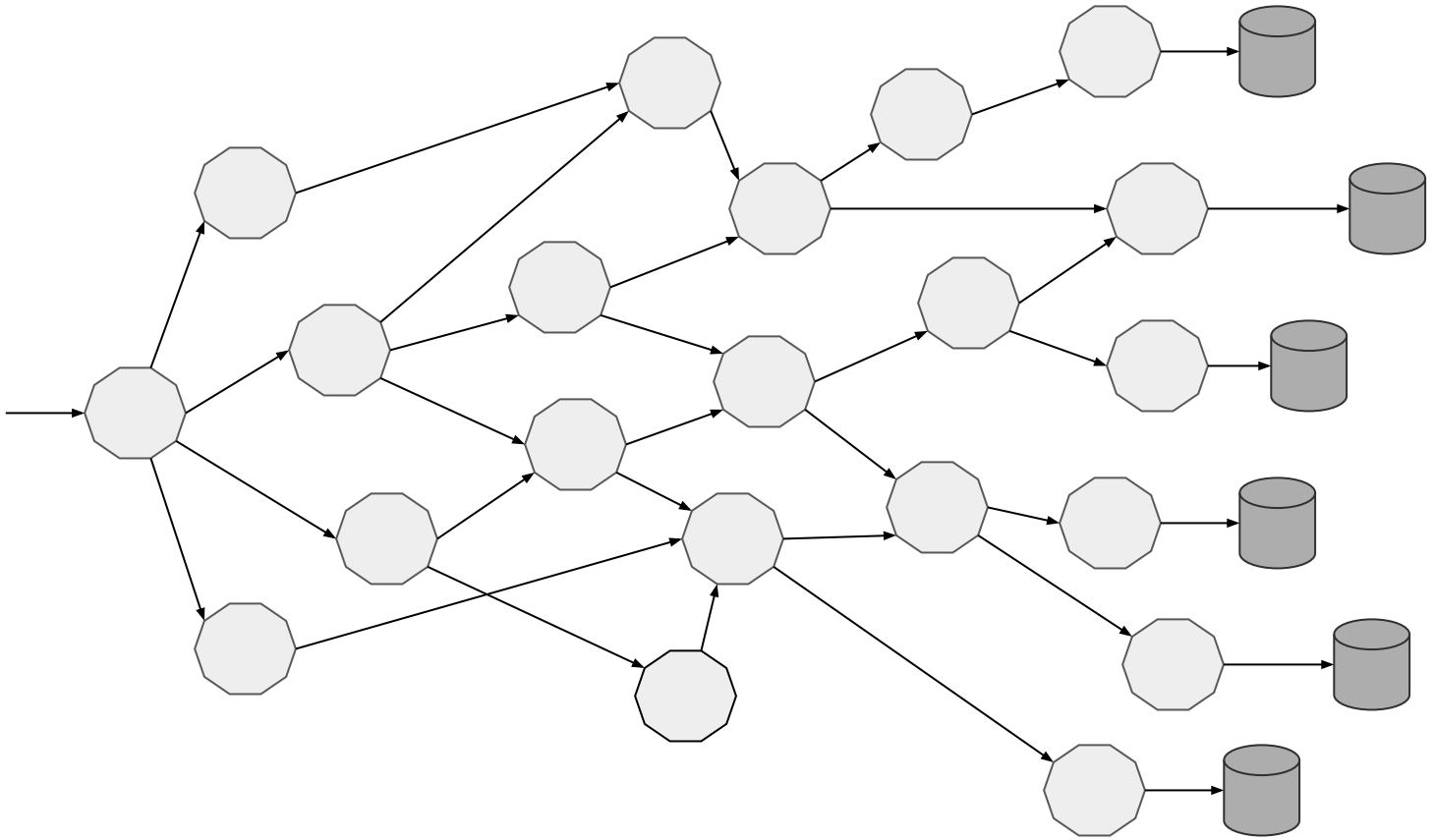
Microservices



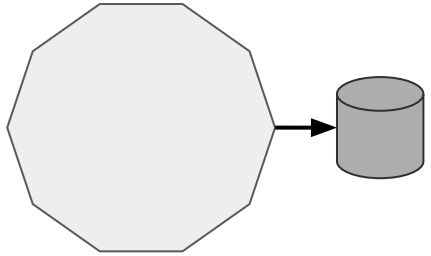
Network of Services



Microservices own their Data

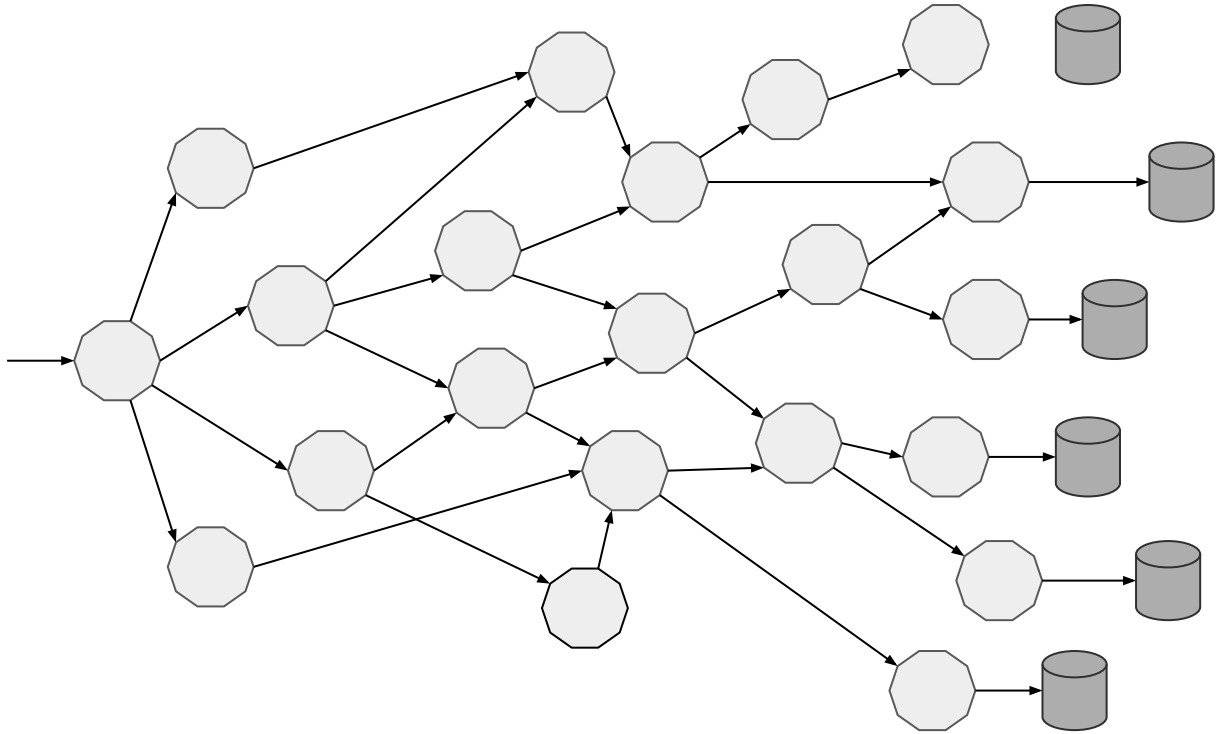


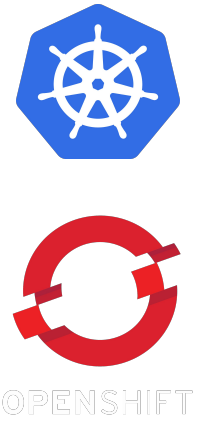
Old School



Love Thy Mono

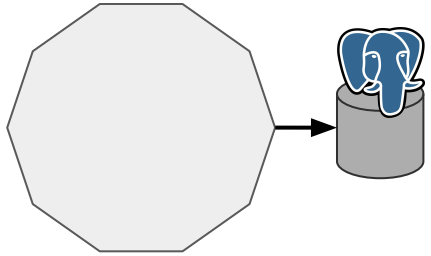
New School





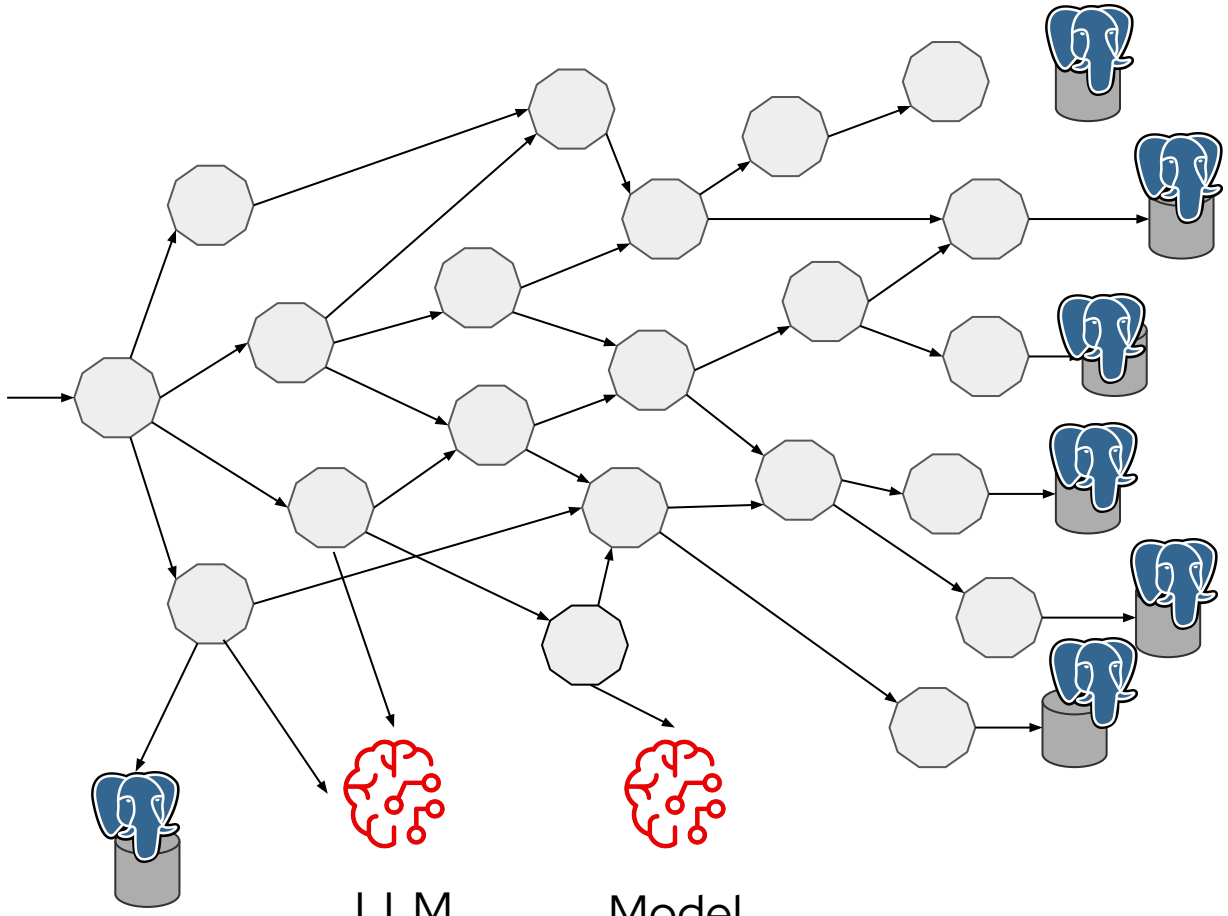
Red Hat

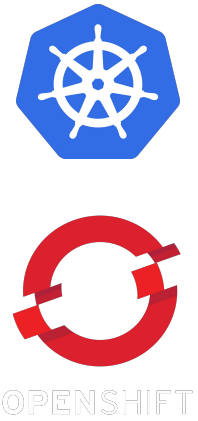
Old School



Love Thy Mono

New School



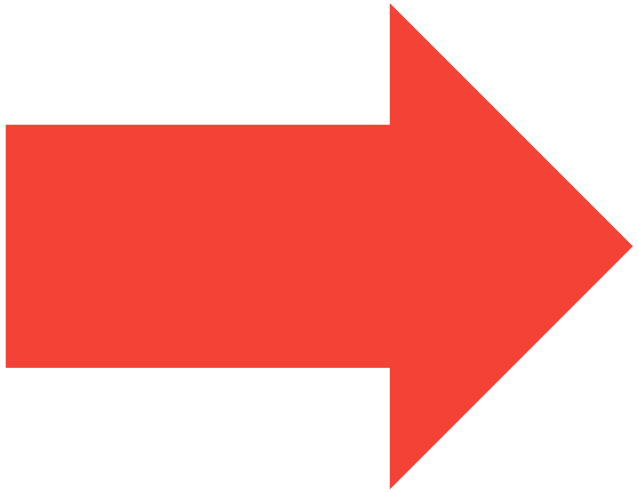


Red Hat

What is Kubernetes?

An open source orchestration system for managing containerized workloads across a cluster of nodes.

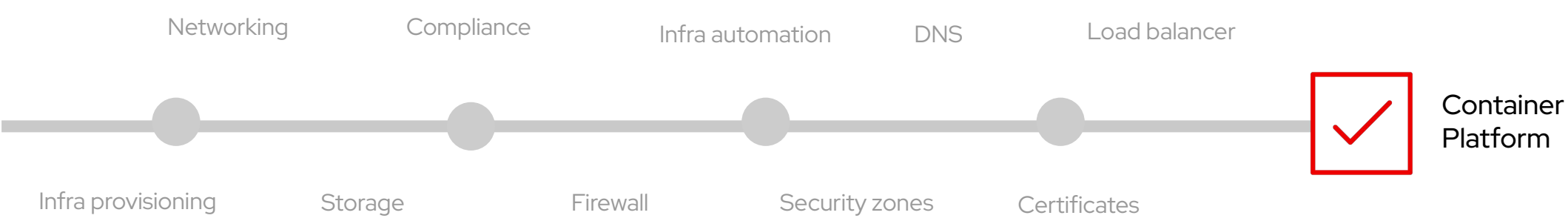
Understanding Kubernetes Objects



Kubernetes objects are persistent entities that represent the desired state of your cluster that you can manage

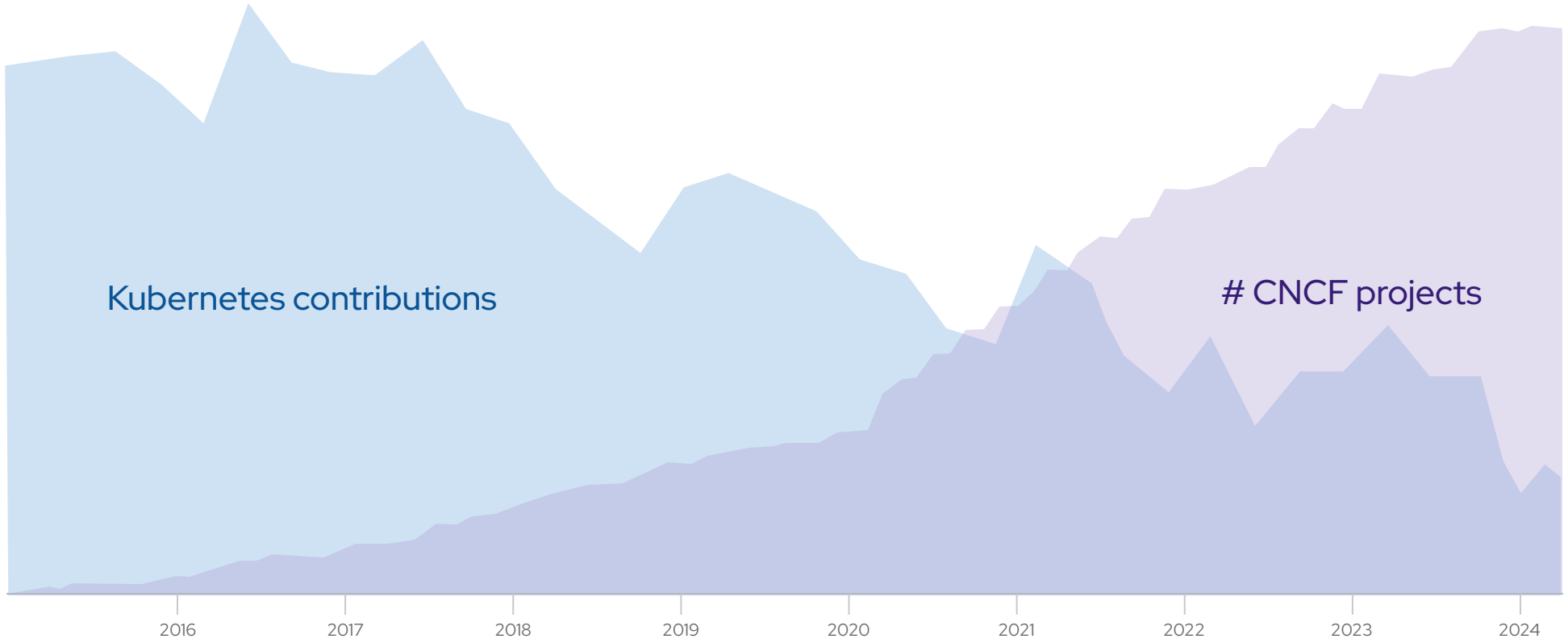
Kubernetes is a **declarative** platform with the K8s API

Kubernetes adoption journey



Red Hat

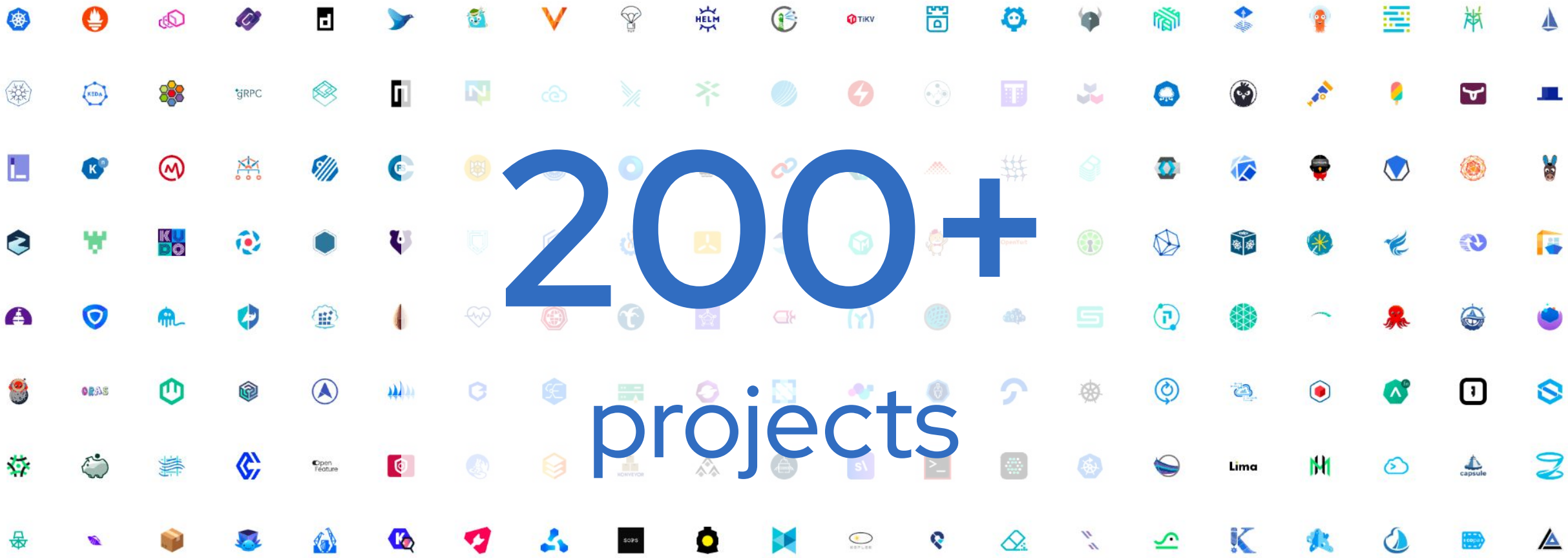
Innovation has shifted to applications and CNCF



Source:
https://www.cncf.io/project-metrics
https://k8s.devstats.cncf.io/d/49/github-stats-by-repository

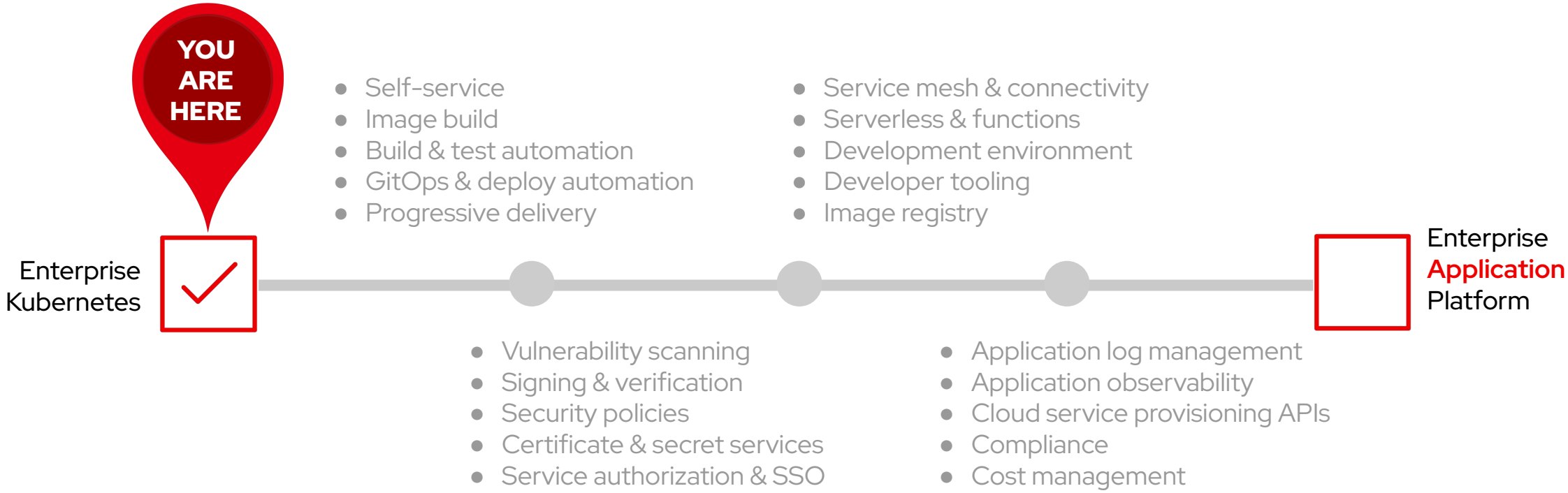
Red Hat

The cloud-native landscape



Red Hat

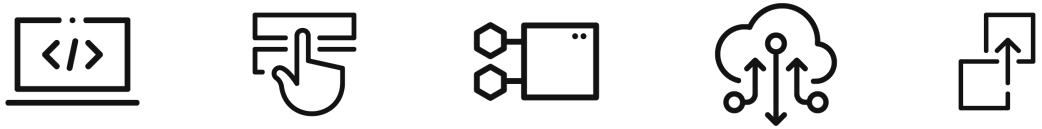
Kubernetes is only as valuable as the applications running on it



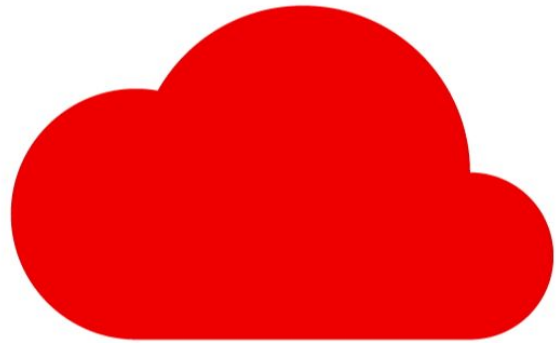
Source: CNCF Platforms White Paper

Red Hat

Hybrid Cloud



Red Hat



Red Hat

Hybrid for the next decade

Clear trends in hybrid cloud architecture have emerged

Applications

A small percentage of applications are cloud-native or on cloud-native architectures

Datacenter

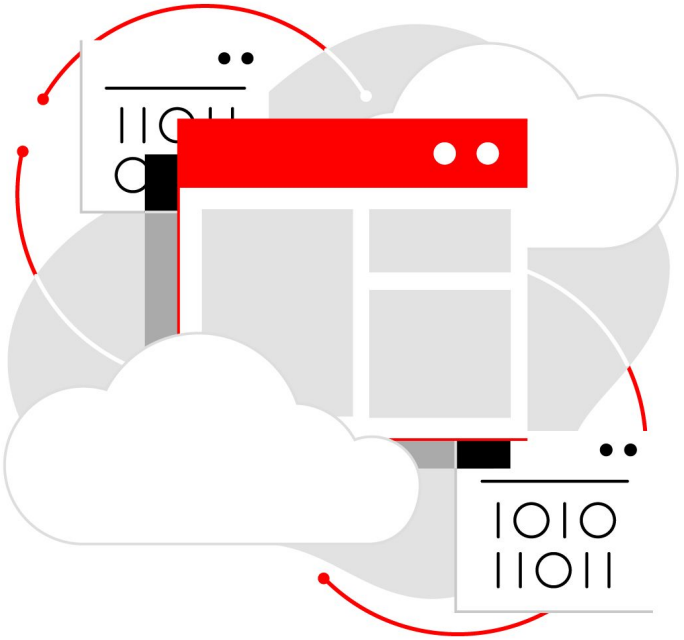
The majority of their existing applications still remain in the datacenter

Edge

A growing number of organizations are planning to run applications at the edge

Red Hat

Manage an application portfolio that spans technologies



Application portfolio

Runs across a mix of datacenter, cloud, and edge environments

Application architectures

Evolve from monolithic and n-tier applications to cloud-native

Data and insights

Provide greater intelligence to applications

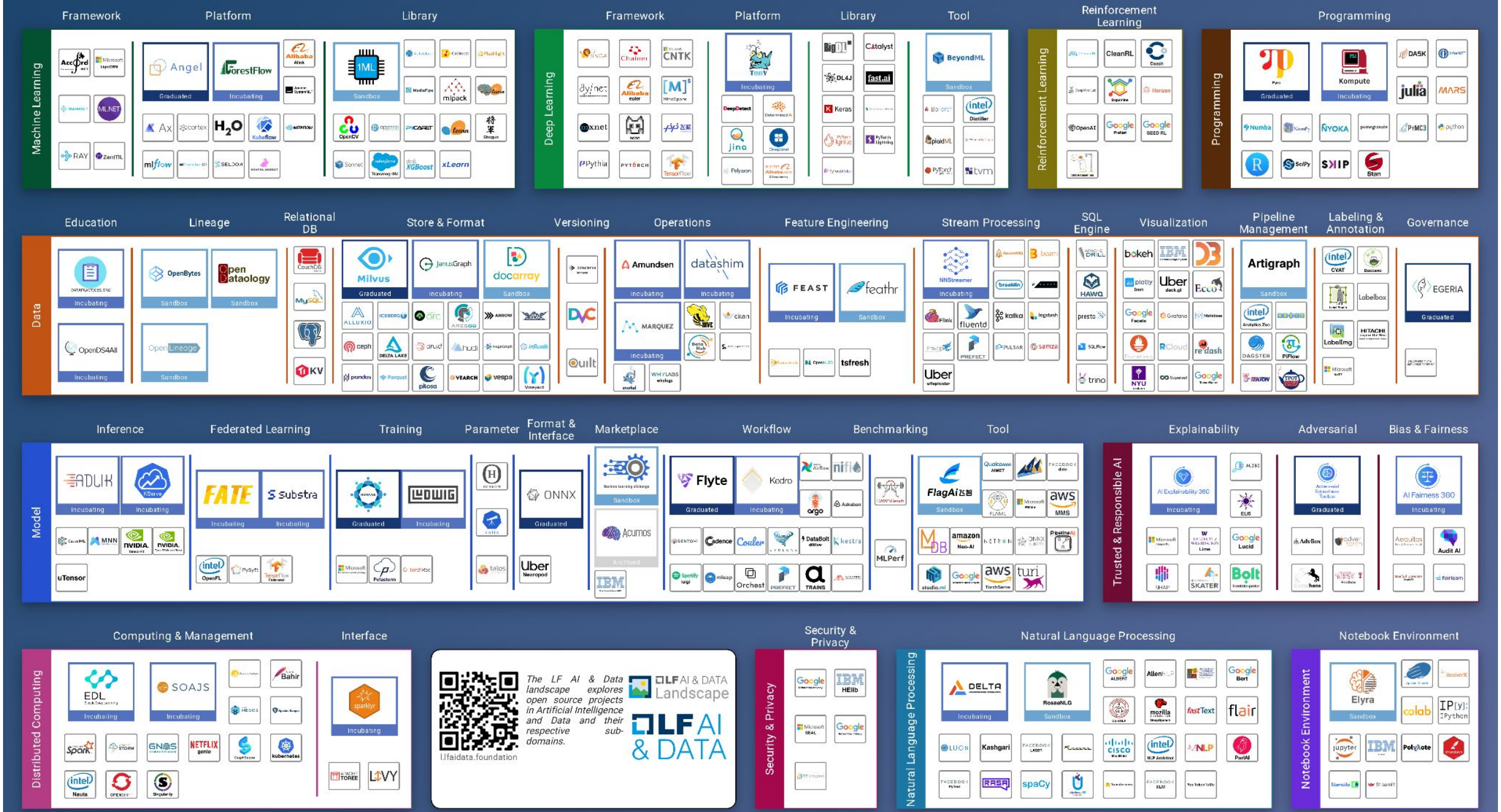

AI
Hi!
How can I help you?

Linux Foundation AI & Data Landscape

1.0

See the interactive landscape at l.lfaidata.foundation

Greyed logos are not open source

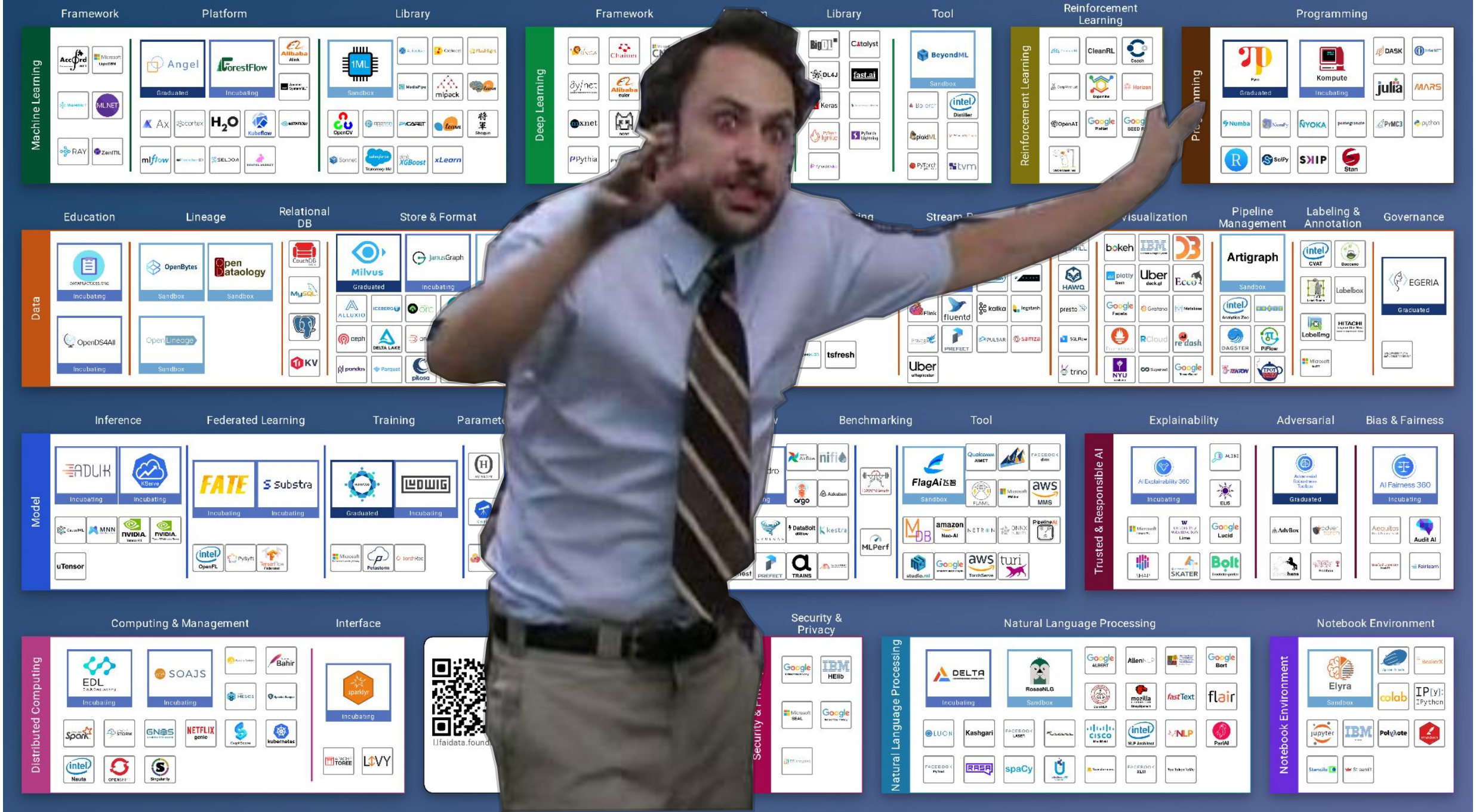


Linux Foundation AI & Data Landscape

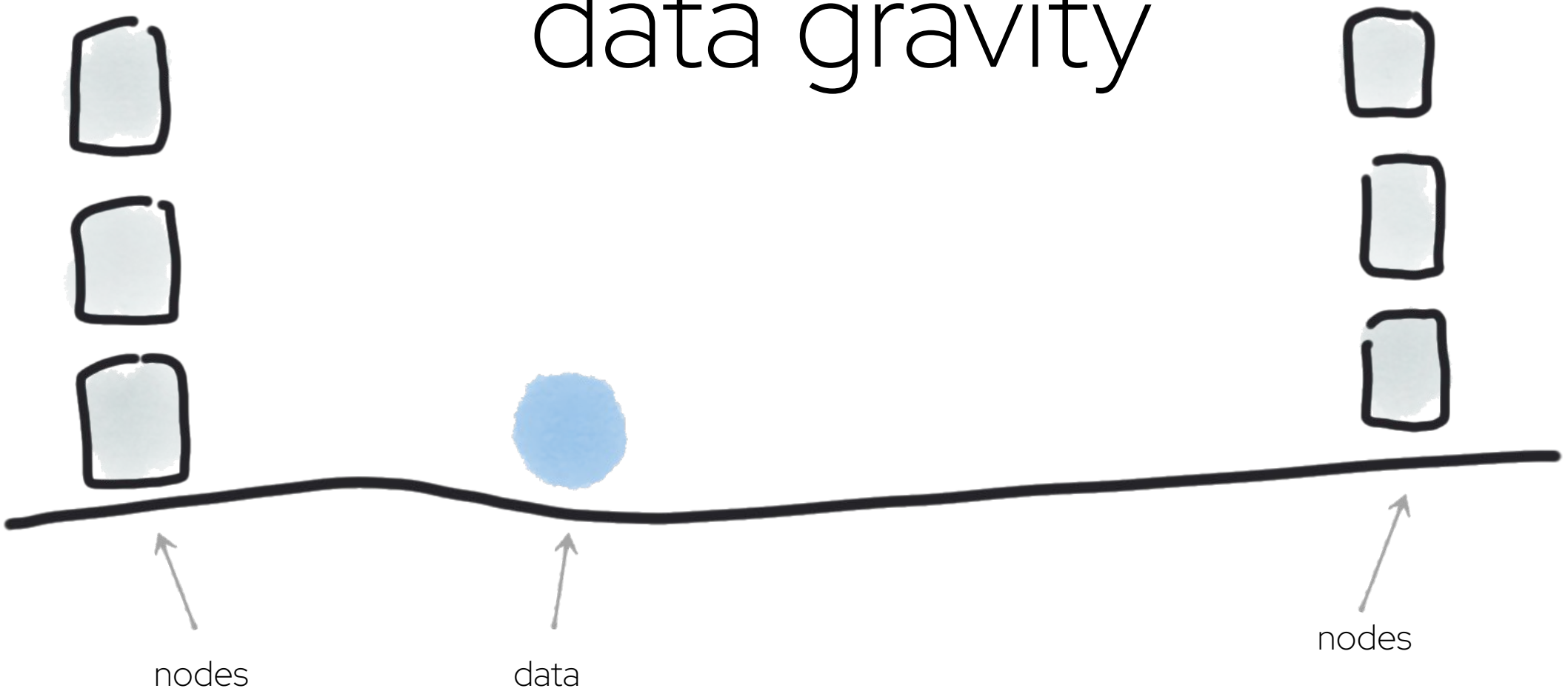
1.0

See the interactive landscape at l.lfaidata.foundation

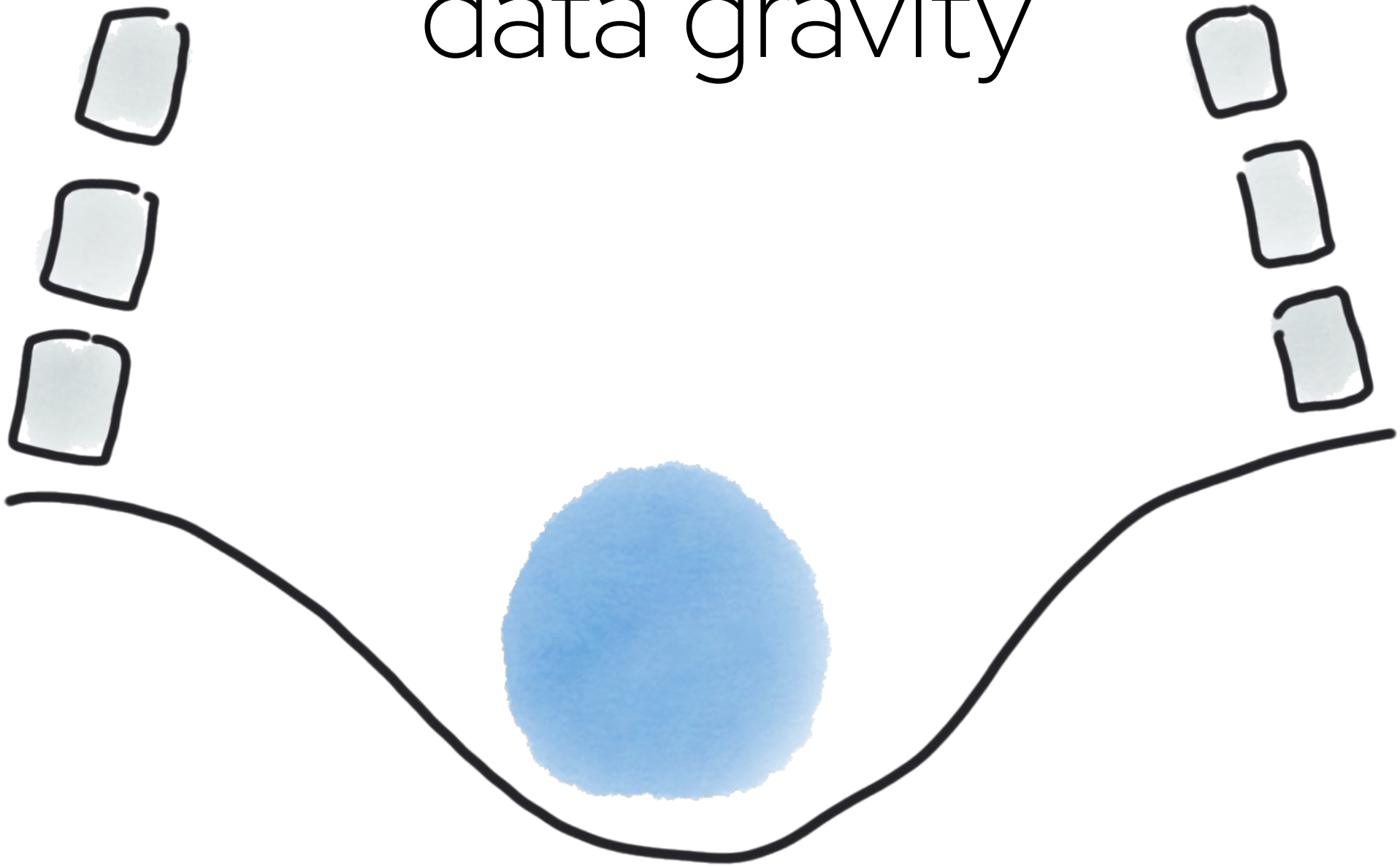
Greyed logos are not open source



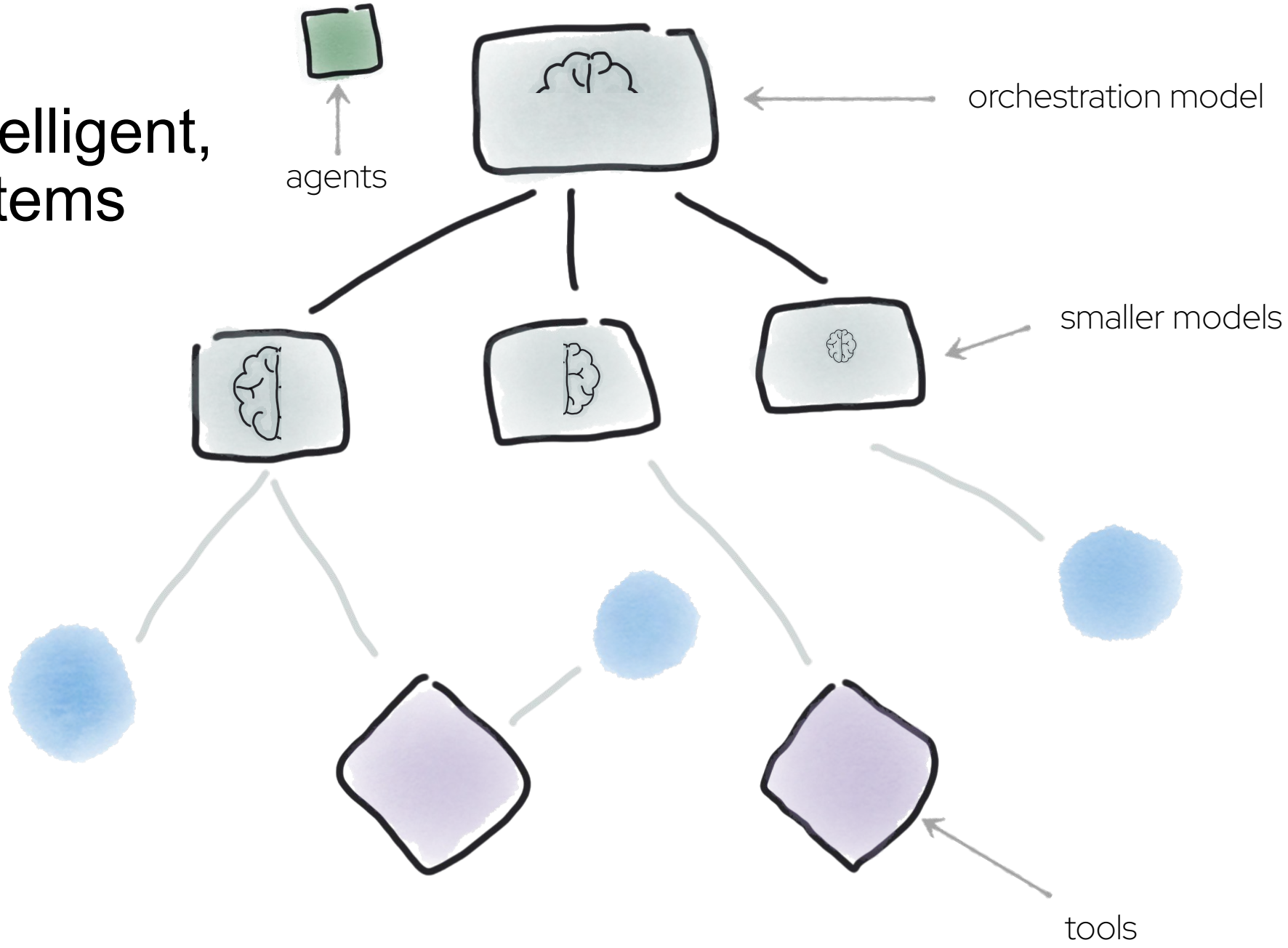
data gravity



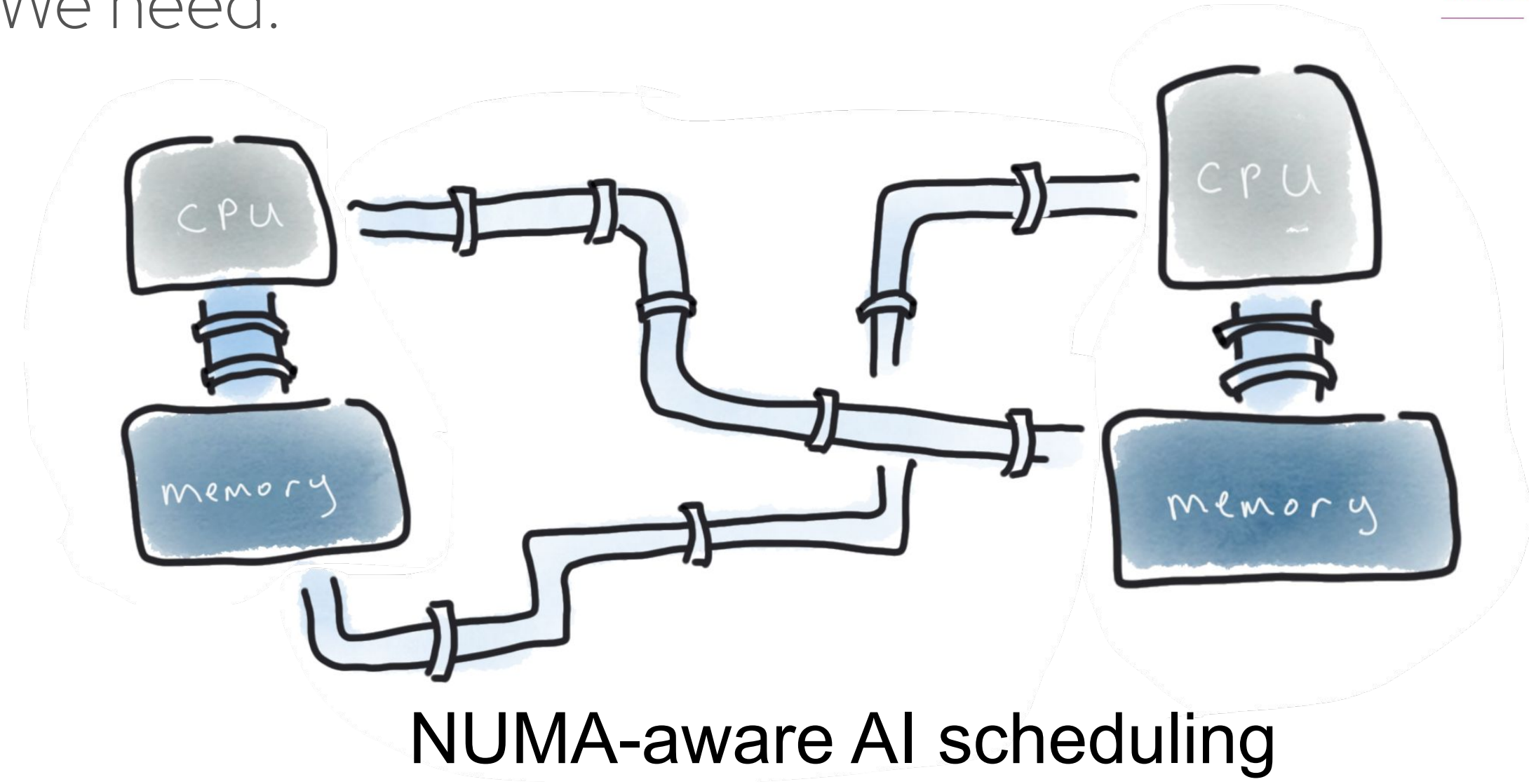
data gravity



A new era of intelligent, data-driven systems

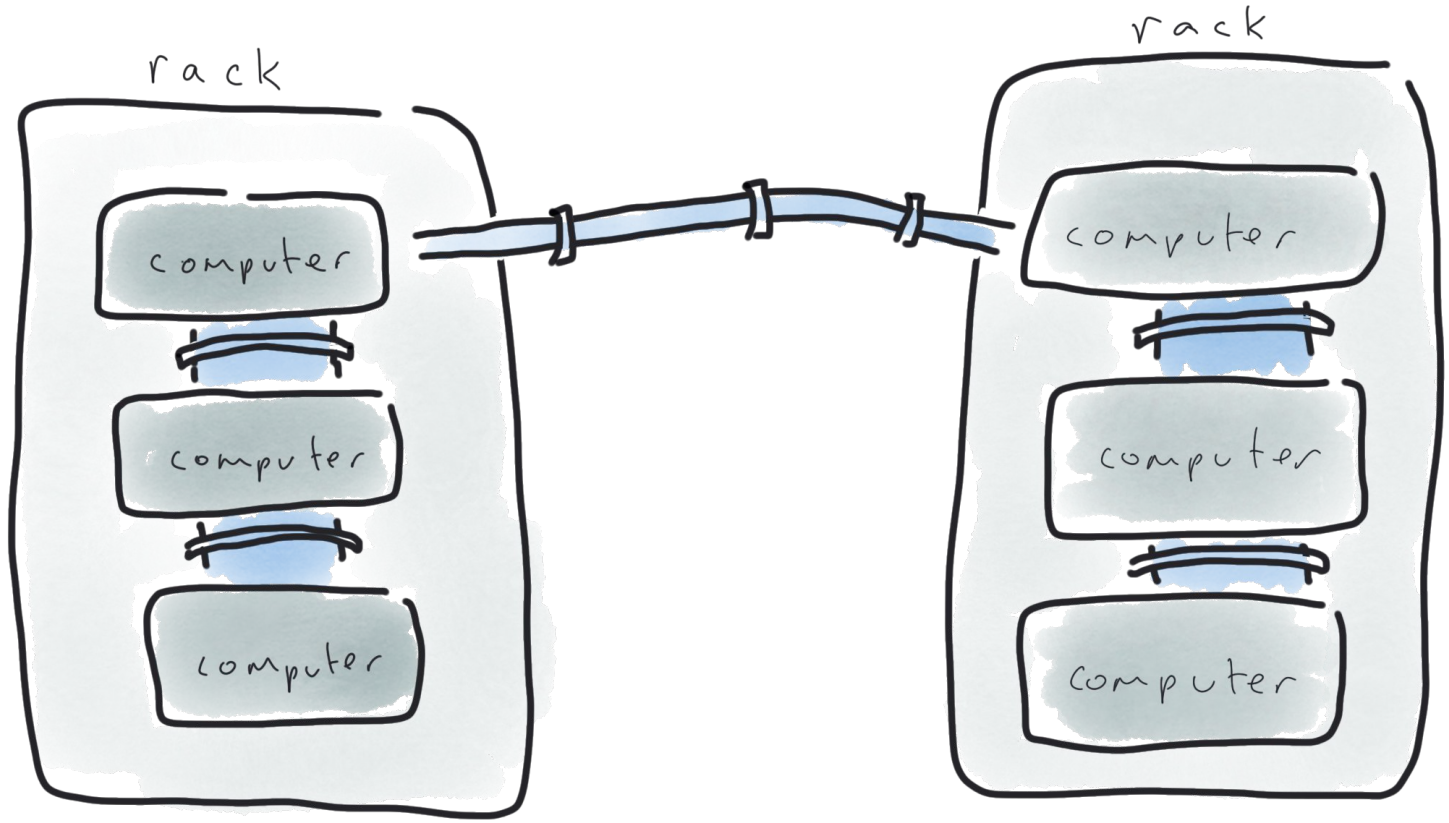


We need:

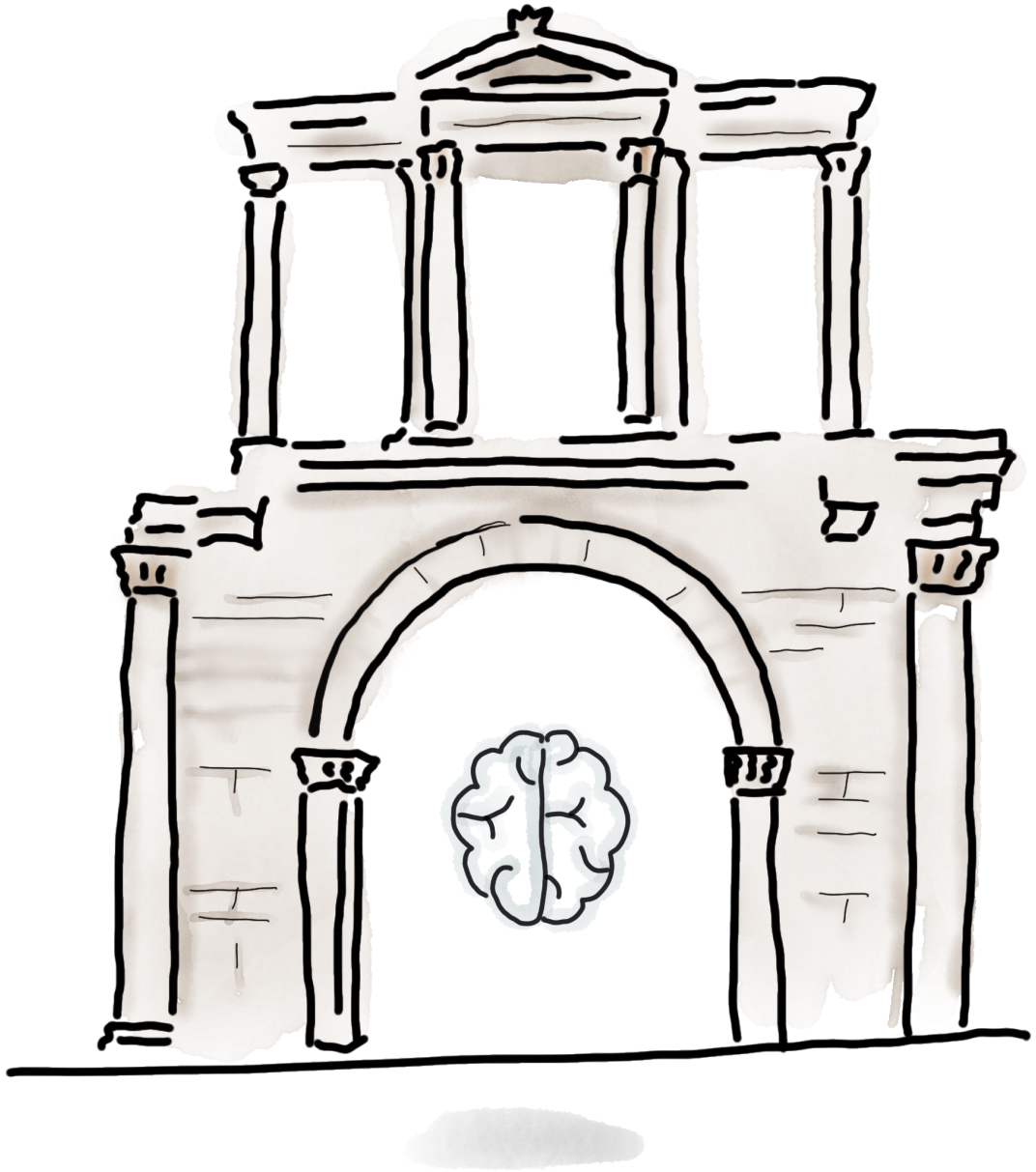


NUMA-aware AI scheduling

We need:

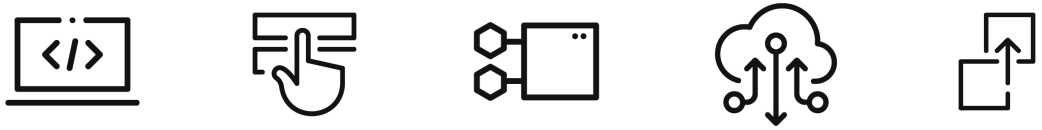


GPU and topology-aware AI scheduling



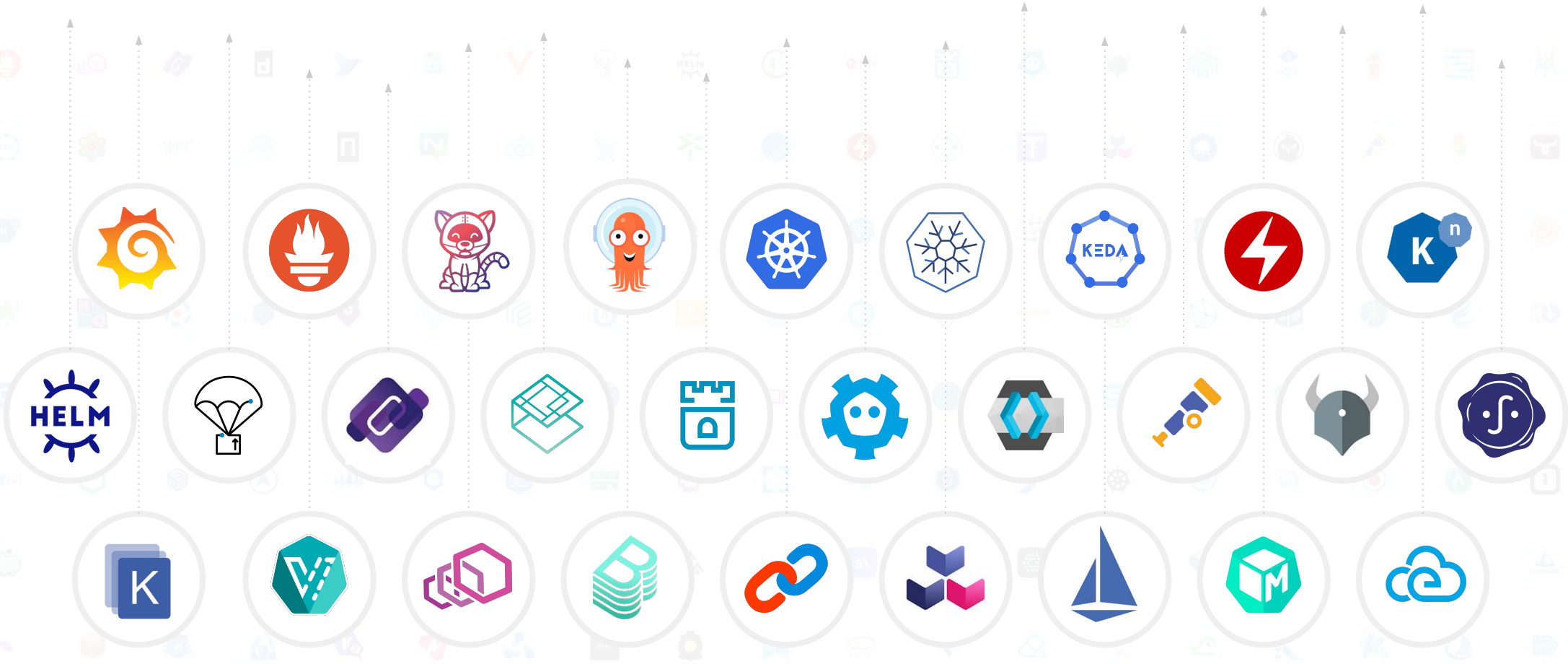
LLM Gateway

Red Hat OpenShift

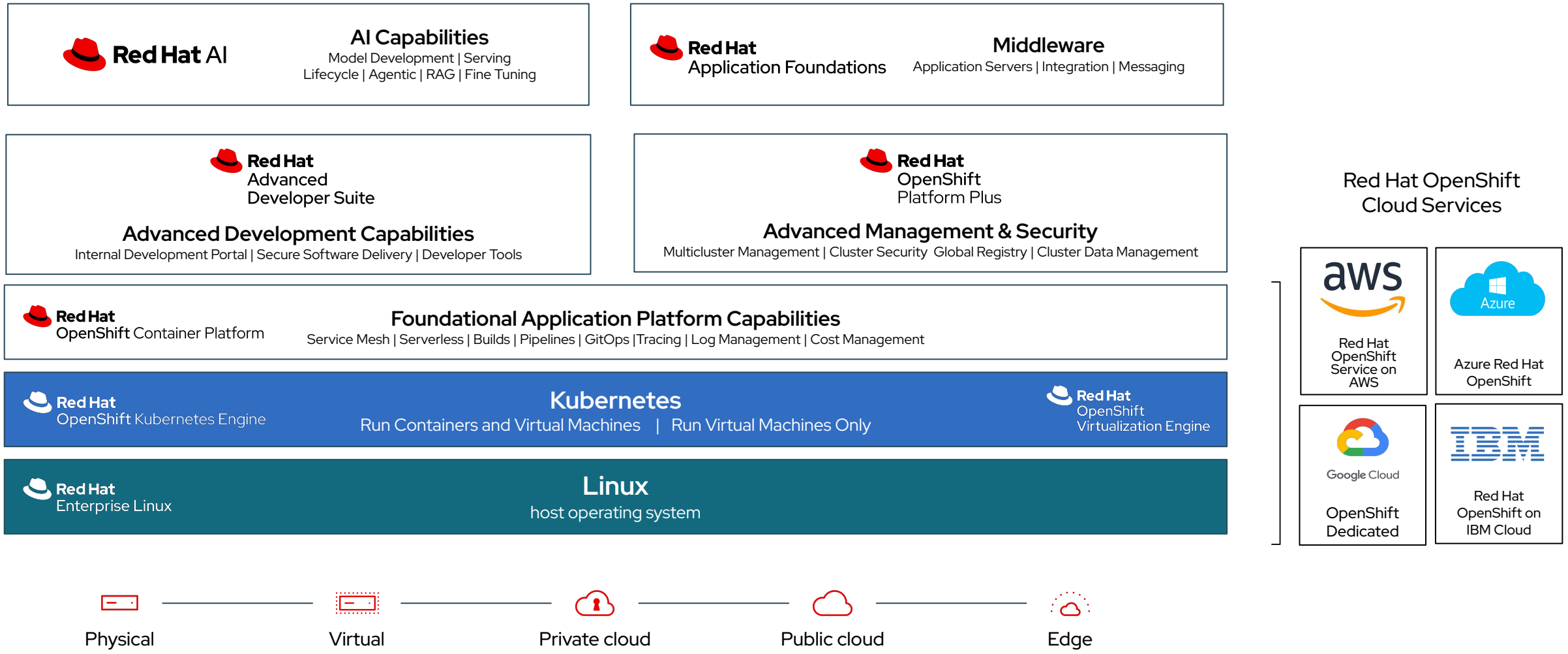


Red Hat

Red Hat OpenShift



Red Hat OpenShift and Open Hybrid Cloud

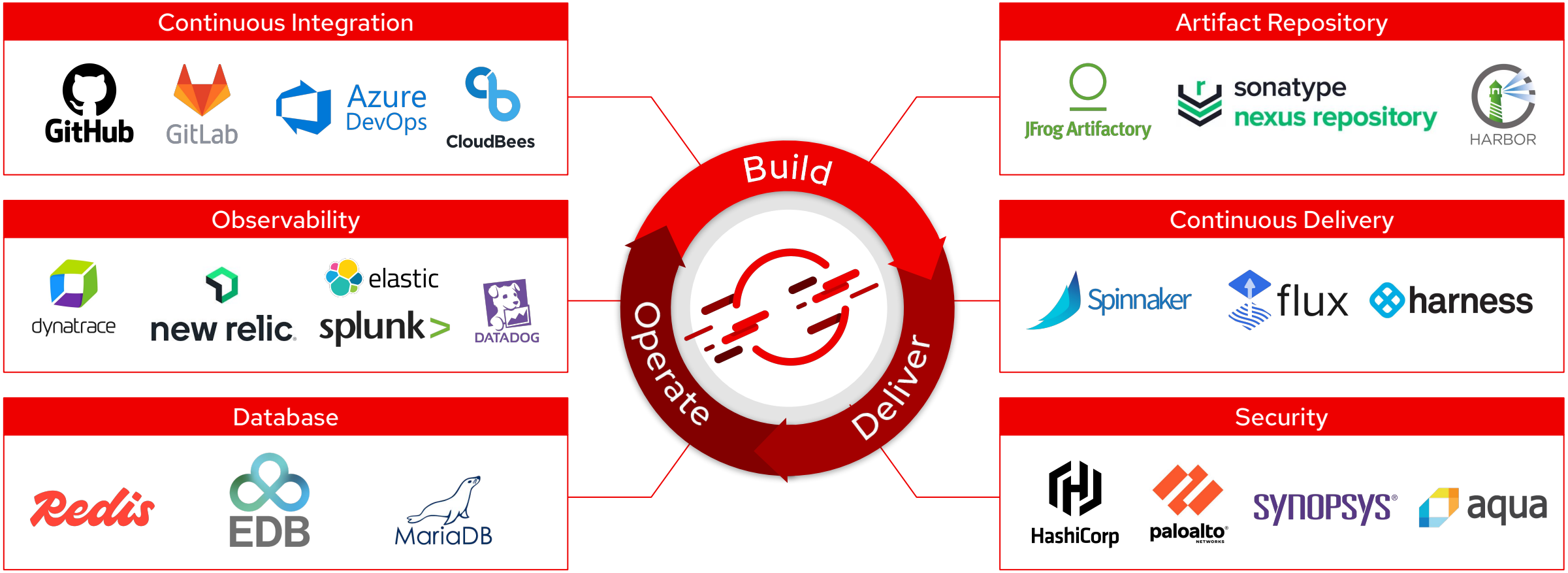


Red Hat OpenShift Cloud Services

Red Hat

Extend with Your Broader Toolchain

Extend Platform Capabilities with Integrations



Red Hat

Why OpenShift?

Trusted
Container engine

Reduce Risk

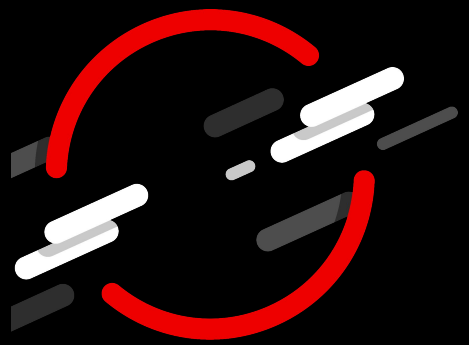
Comprehensive
Application platform

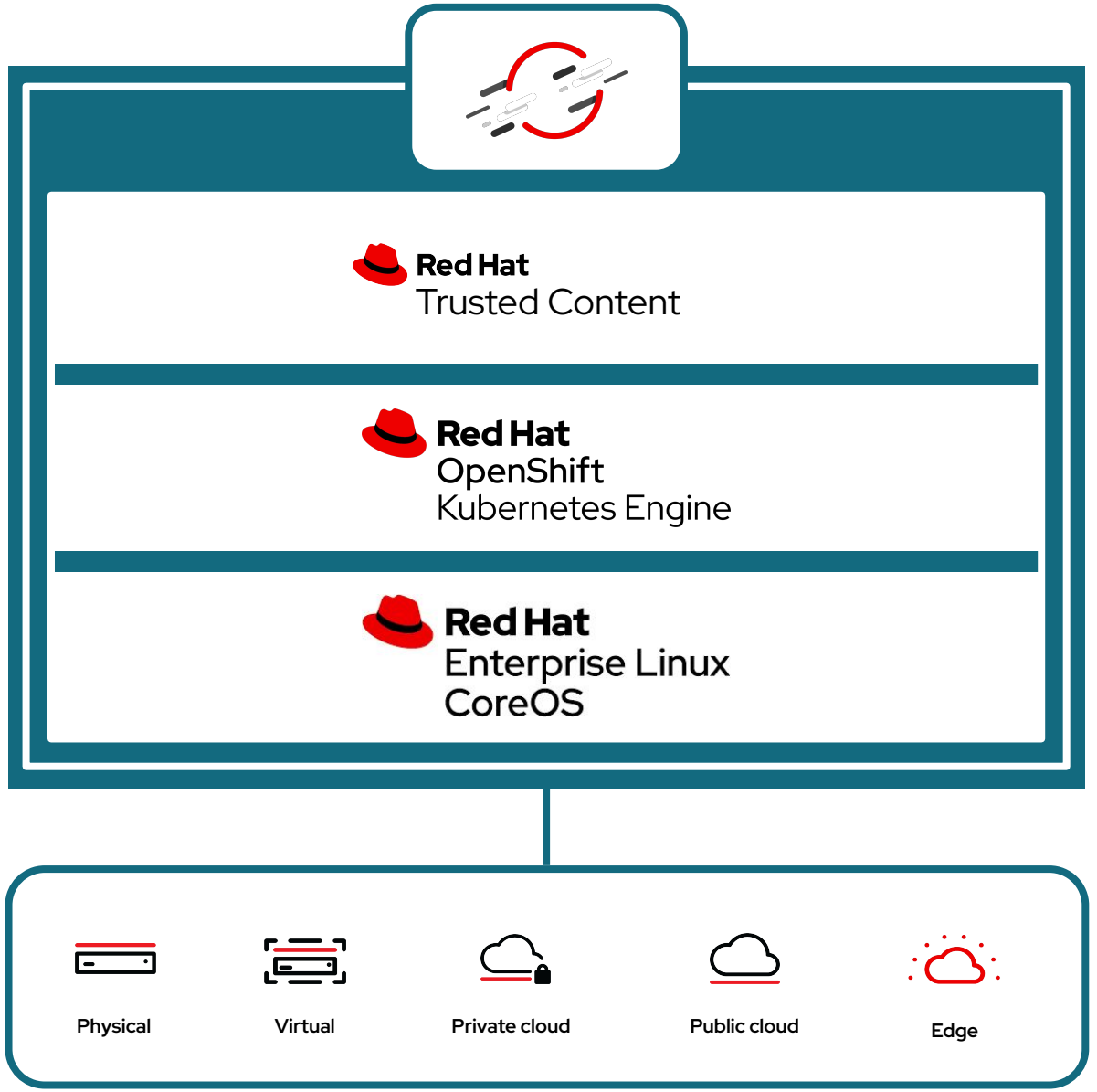
Improve Productivity

Consistent
Across hybrid cloud

Increase Flexibility

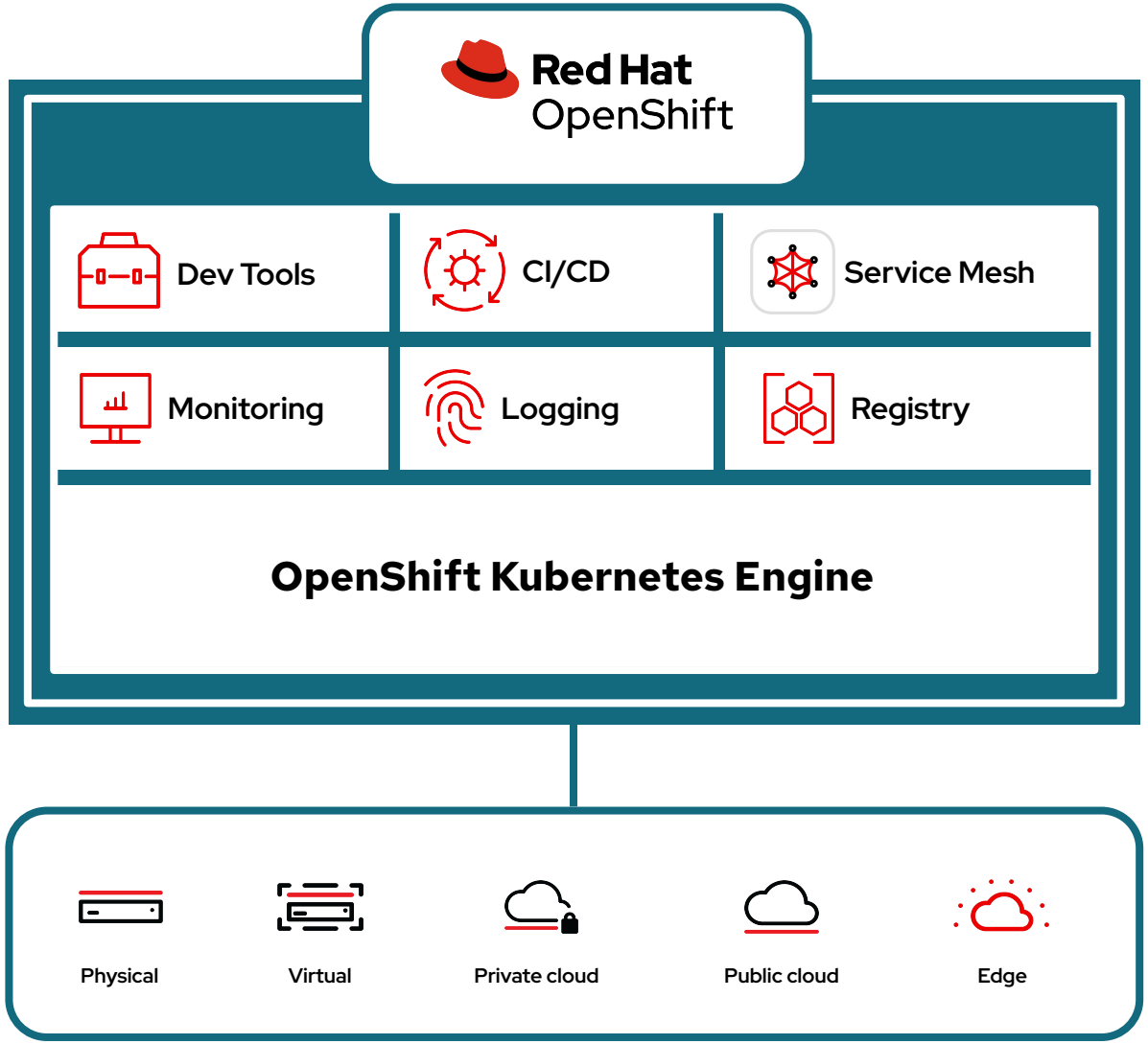
OpenShift is built on a Trusted Container Engine





Red Hat

OpenShift Delivers a Comprehensive Application Platform

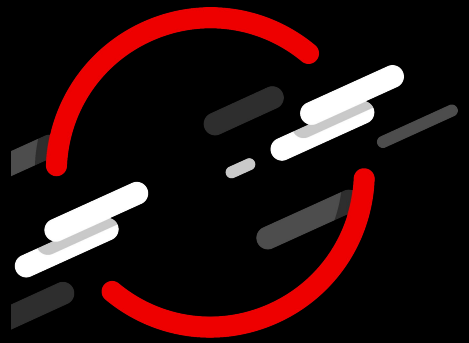


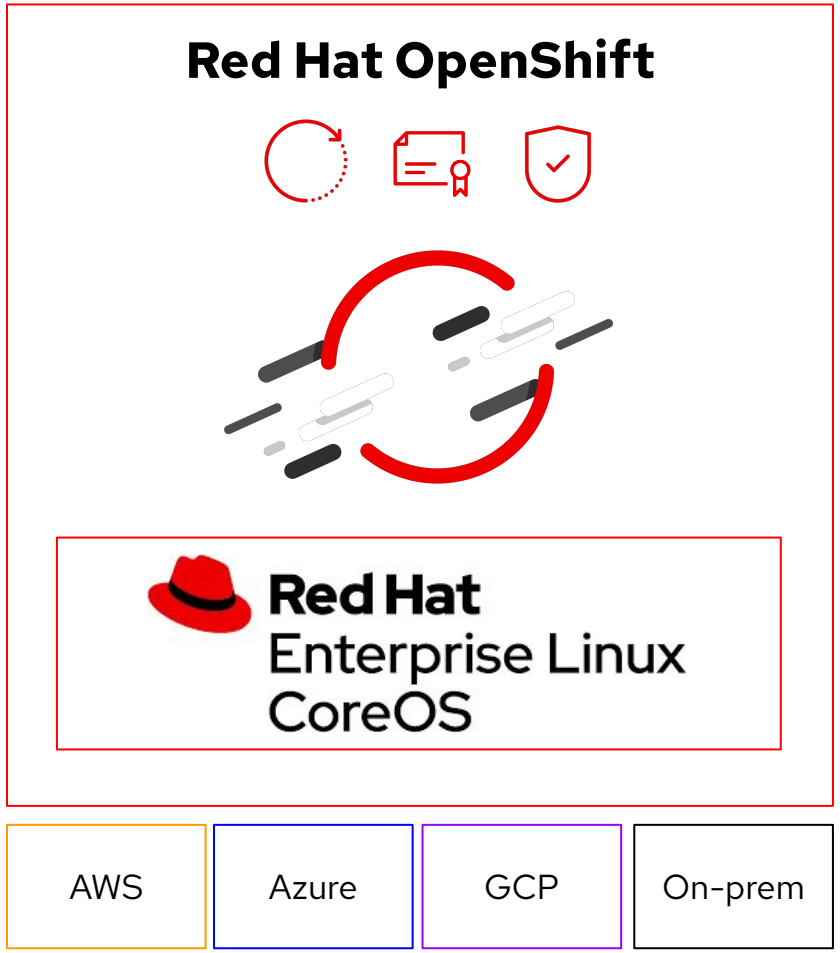
Red Hat

OpenShift Delivers a

Consistent

Experience across the hybrid cloud



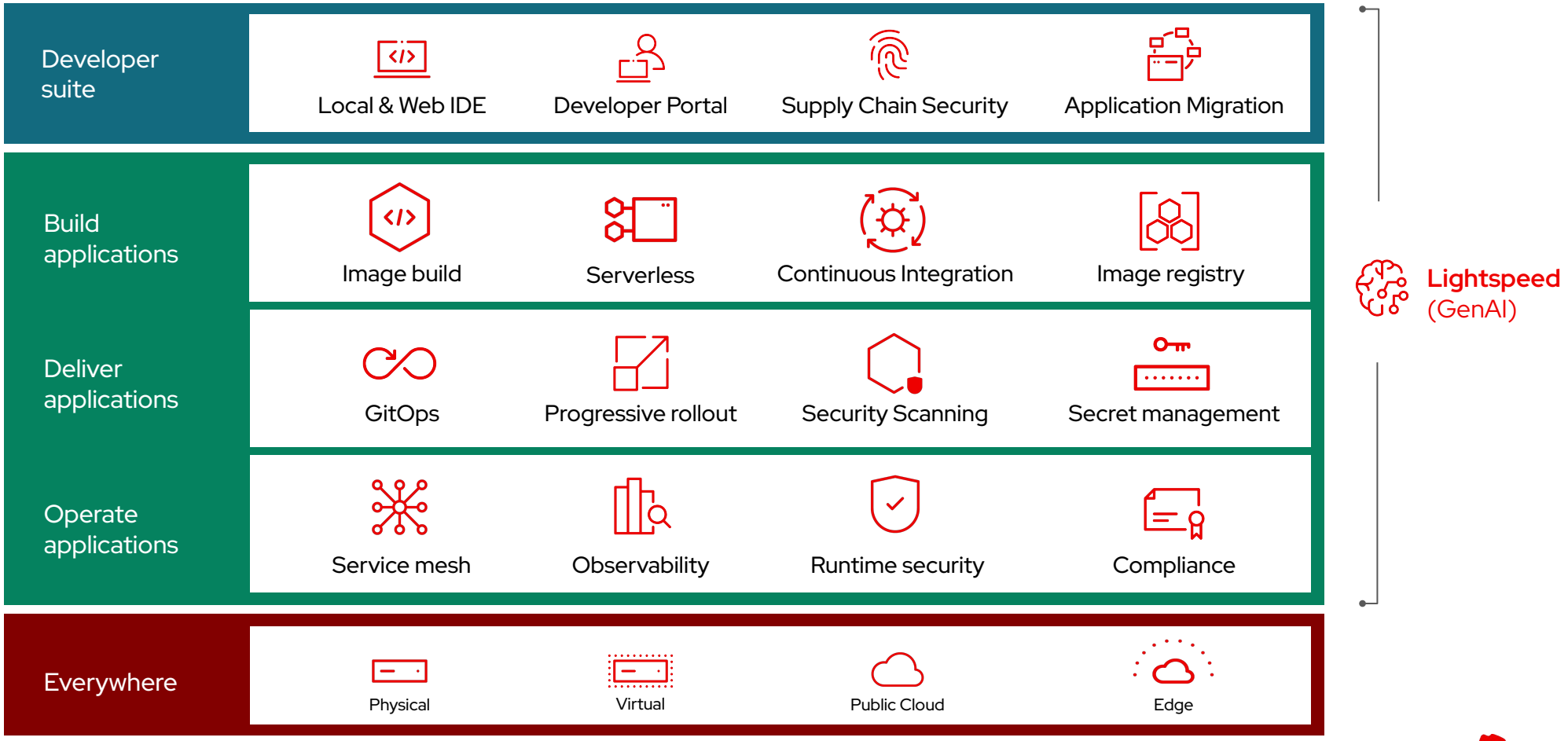


One platform across all environments

Red Hat

OpenShift Application Platform

The cloud-native stack for the enterprise



Red Hat

Don’t just take our word for it…

Red Hat

Red Hat is *a Leader* in the 2024 Gartner® Magic Quadrant™: Cloud Application Platforms

Cloud **Application** Platforms are intended to be more than just a platform for running applications; they are essential for businesses aiming to achieve excellence in software engineering, productivity and market responsiveness." *Gartner*

Source: Gartner, "Magic Quadrant for Cloud Application Platforms," By Tigran Egiazarov, Mukul Saha, Anne Thomas, Steve Schwent, 4 November 2024

Red Hat

Red Hat
OpenShift

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/OpenShift


facebook.com/redhatinc


twitter.com/Openshift

Red Hat

PostgreSQL as an AI Data Management Platform for OpenShift AI.

Franck Sidi, Davide Tamaro EDB
Natale Vinto - Red Hat

Get more value from your data
with an enterprise-ready
Omni-Data Platform for
transactional, analytical and AI
workloads,
with high-availability, scalability,
and compliance - out of the box

The promise of **AI and data.**

PG.AI

Database

Database Servers

- Enterprise Postgres
- Oracle Compatible
- Community PostgreSQL

- Multi-Model Extensions
- Management & Observability (Basic)
- Kubernetes Operators
- Supply Chain Security
- Migration Toolkit
- High Availability

Analytics Accelerator

- Analytics
  Columnar Query Engine
  MPP Warehouse
- Lakehouse Storage
  Delta Tables, Iceberg
- Support for Greenplum Workloads

AI Factory

- Vector Engine
- AI Pipeline
- GenAI Builder
- Agent Orchestrator
- Model Serving

Hybrid Management

- Hybrid Observability
- Hybrid DBaaS
- Distributed HA (99.999%)
- Migration Center

Deploy Anywhere

ENGINEERED SYSTEM | HYBRID SOFTWARE | MANAGED CLOUD

Deployment Partners: AWS, GCP, Azure, Red Hat, IBM, Supermicro

Integration Services (CX)

Solution Architecture | Pilot to Production SLAs | Managed Services

PG.AI

Database

Database Servers

- Enterprise Postgres
- Oracle Compatible
- Community PostgreSQL

- Multi-Model Extensions
- Management & Observability (Basic)
- Kubernetes Operators
- Supply Chain Security
- Migration Toolkit
- High Availability

Analytics Accelerator

- Analytics
  Columnar Query Engine
  MPP Warehouse
- Lakehouse Storage
  Delta Tables, Iceberg
- Support for Greenplum Workloads

AI Factory

- **Vector Engine**
- **AI Pipeline**
- **GenAI Builder**
- **Agent Orchestrator**
- **Model Serving**

Hybrid Management

- Hybrid Observability
- Hybrid DBaaS
- Distributed HA (99.999%)
- Migration Center

Deploy Anywhere

ENGINEERED SYSTEM | HYBRID SOFTWARE | MANAGED CLOUD

Deployment Partners: AWS, GCP, Azure, Red Hat, IBM, Supermicro

Integration Services (CX)

Solution Architecture | Pilot to Production SLAs | Managed Services

68%

Marriage of AI and data is mission critical for success

EDB May 2025

THE AI IMPLEMENTATION GAP

80%

6-24 months to production

Morning Consult for IBM 2024

30%

Abandoned after POC

Gartner July 2024

Key business challenges addressed by EDB AI Factory

Accelerate time to value with a complete agentic AI system for intelligent applications

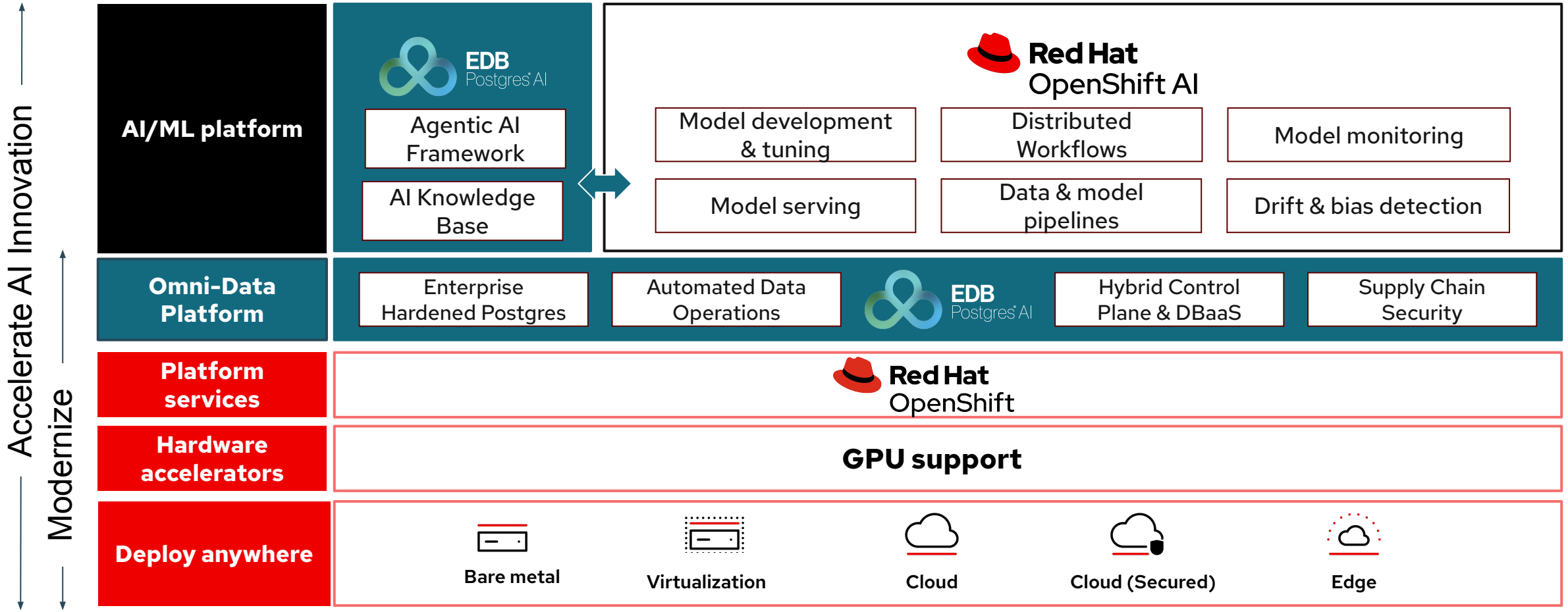
Empower everyone with a low barrier to agentic AI

Mitigate integration complexity

Safeguard critical data assets

SOVEREIGN AI AND DATA PLATFORM FOR PRODUCTION-READY AI

EDB Postgres AI and Red Hat OpenShift AI work seamlessly together to provide a comprehensive platform



EDB

Red Hat

Red Hat AI

Accelerate the development and delivery of AI solutions across hybrid-cloud environments

Increase efficiency with **fast, flexible and efficient inferencing**

Simplified and consistent experience for **connecting models to data**

Accelerate Agentic AI delivery and stay at the forefront of innovation

Flexibility and consistency when **scaling AI across the hybrid cloud**

Red Hat AI

Red Hat
AI Inference Server

Red Hat
Enterprise Linux AI

Red Hat
OpenShift AI

Trusted, Consistent and Comprehensive foundation

NVIDIA AMD intel Hardware Acceleration Google aws IBM

Physical

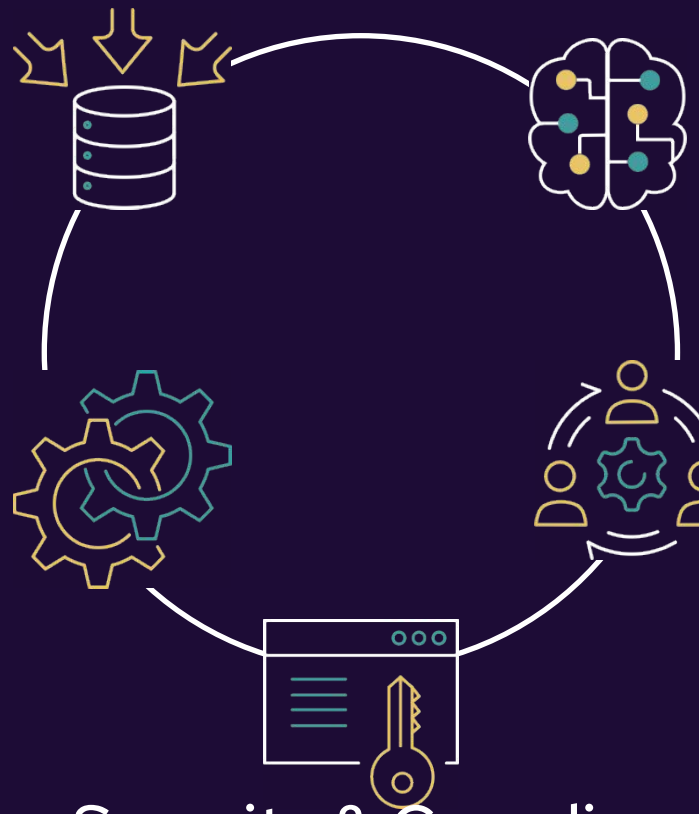
Virtual

Private Cloud

Public Cloud

Edge

THE CHALLENGES OF GEN AI IN PRODUCTION



Data Silos

Complex AI Production

Lack of deployment flexibility

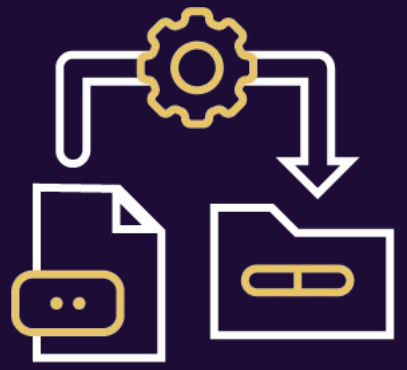
Collaboration is lacking

Security & Compliance

EDB

Red Hat

Deployment using EDB Postgres AI and Red Hat OpenShift AI



CREATE THE DATA PIPELINE

CREATE THE KNOWLEDGE BASE

CREATE THE VIRTUAL ASSISTANT

DEPLOY THE CHAT BOT

EDB

Red Hat

DELIVERING ON THE PROMISE OF AI

Simplified AI development

Faster deployment

Data sovereignty

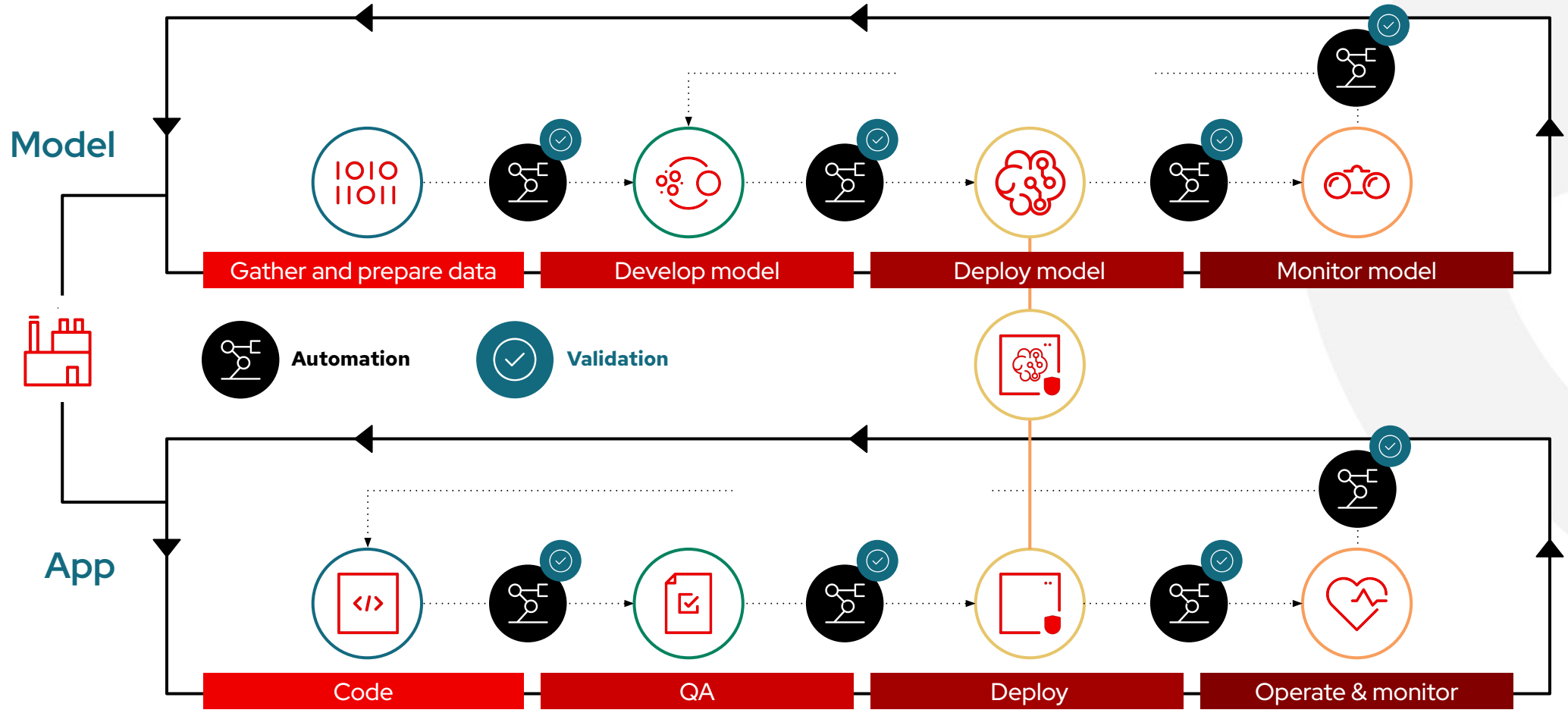
Hybrid deployments

Lower operational costs

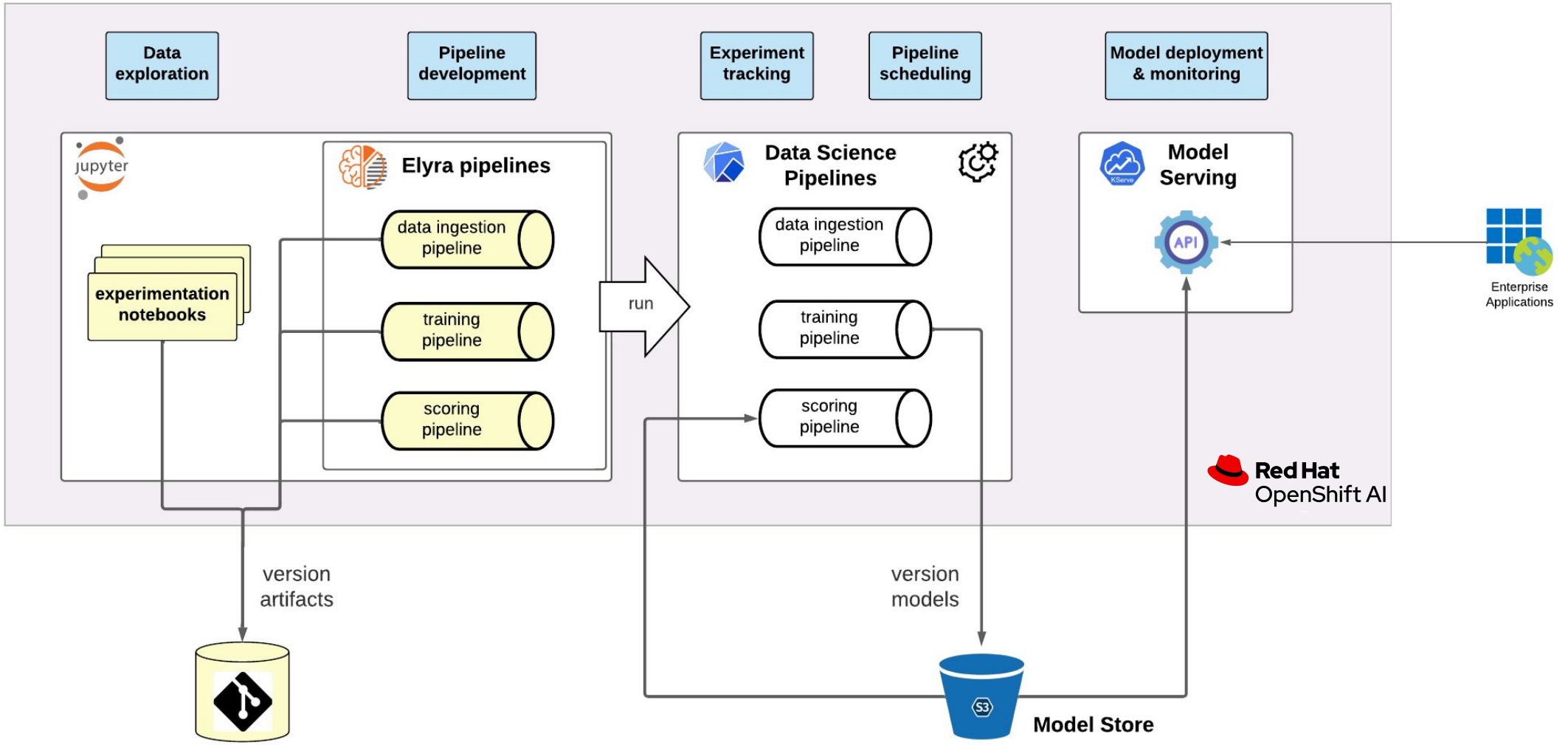
99.999% Availability

Observability

Lifecycle for operationalizing models



ML workflow with OpenShift AI



Gen AI Use cases…

Employee Assistant

- Credit Risk assessment/Loan Approval
- Policies and Regulations advisor
- Customer Support
- HR CVs recommendation and summarization
- RFP, RFP Responses, Contracts summarization

Customer Experience

- Gen AI enabled Chatbots
- Get answers intelligently from knowledge bases (FAQ, Policies, Manuals, …)
-

OCR automation

- KYC documents (Passports, ID Cards, …)
- Cheques

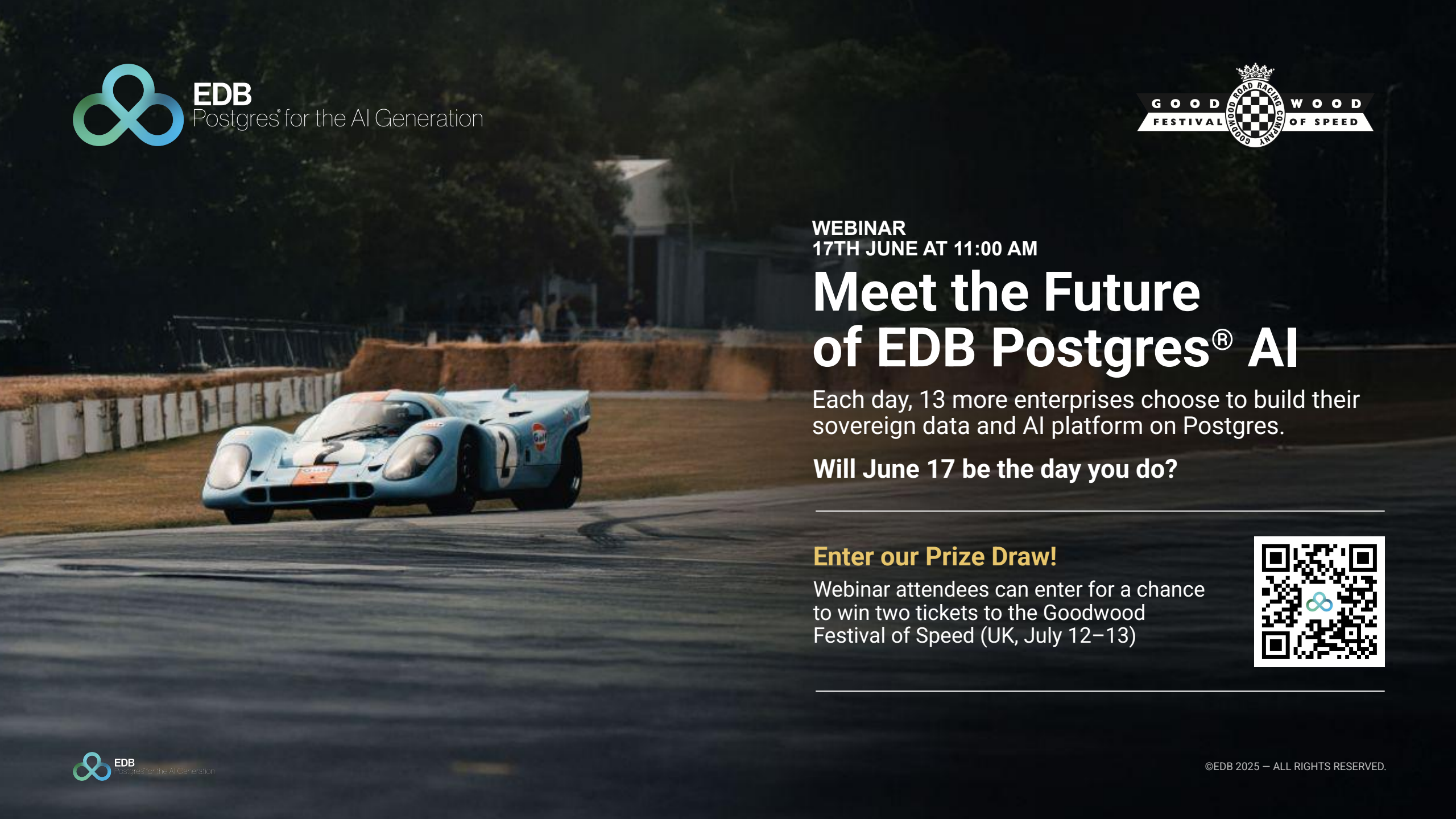
Semantic Search

- Documents
- Emails
- Multi-Lingual
- Search in images

Demonstration EDB Postgres AI on Red Hat OpenShift AI

EDB

Red Hat



EDB
Postgres® for the AI Generation

GOODWOOD FESTIVAL OF SPEED

WEBINAR
17TH JUNE AT 11:00 AM

Meet the Future of EDB Postgres® AI

Each day, 13 more enterprises choose to build their sovereign data and AI platform on Postgres.

Will June 17 be the day you do?

Enter our Prize Draw!

Webinar attendees can enter for a chance to win two tickets to the Goodwood Festival of Speed (UK, July 12–13)

Red Hat | intel. | GitLab

Developer Days

Milano e Roma

8 e 10 luglio, 2025