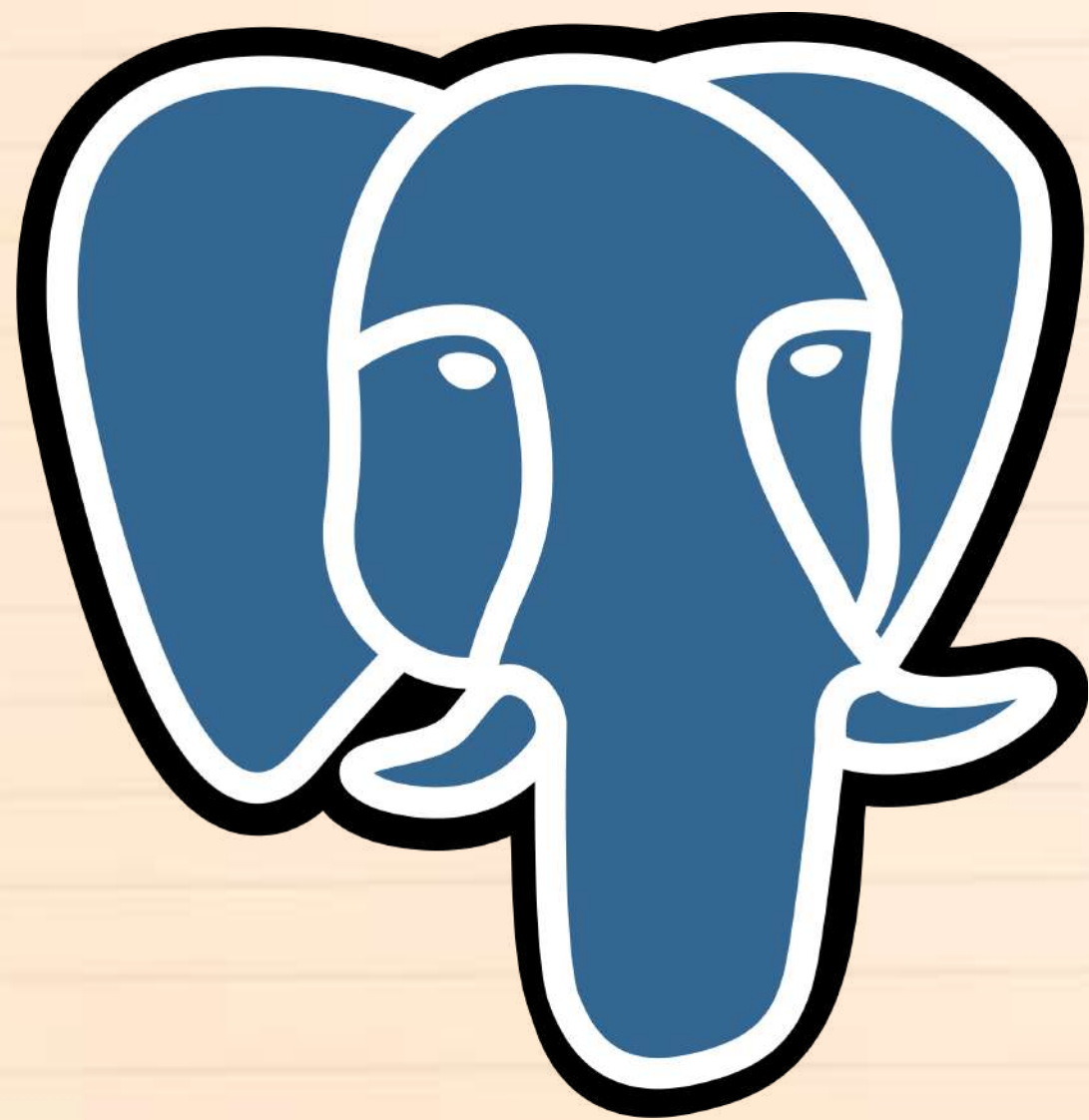


Do



Don't



PostgreSQL ❤️ Kubernetes

Chris Engelbert

Devrel @ simplyblock

Previous fun companies:

- Ubisoft / Blue Byte
- Hazelcast
- Instana
- clevabit
- Timescale

Interests:

- Developer Relations
- Anything Performance Engineering
- Backend Technologies
- Fairy Tales (AMD, Intel, Nvidia)

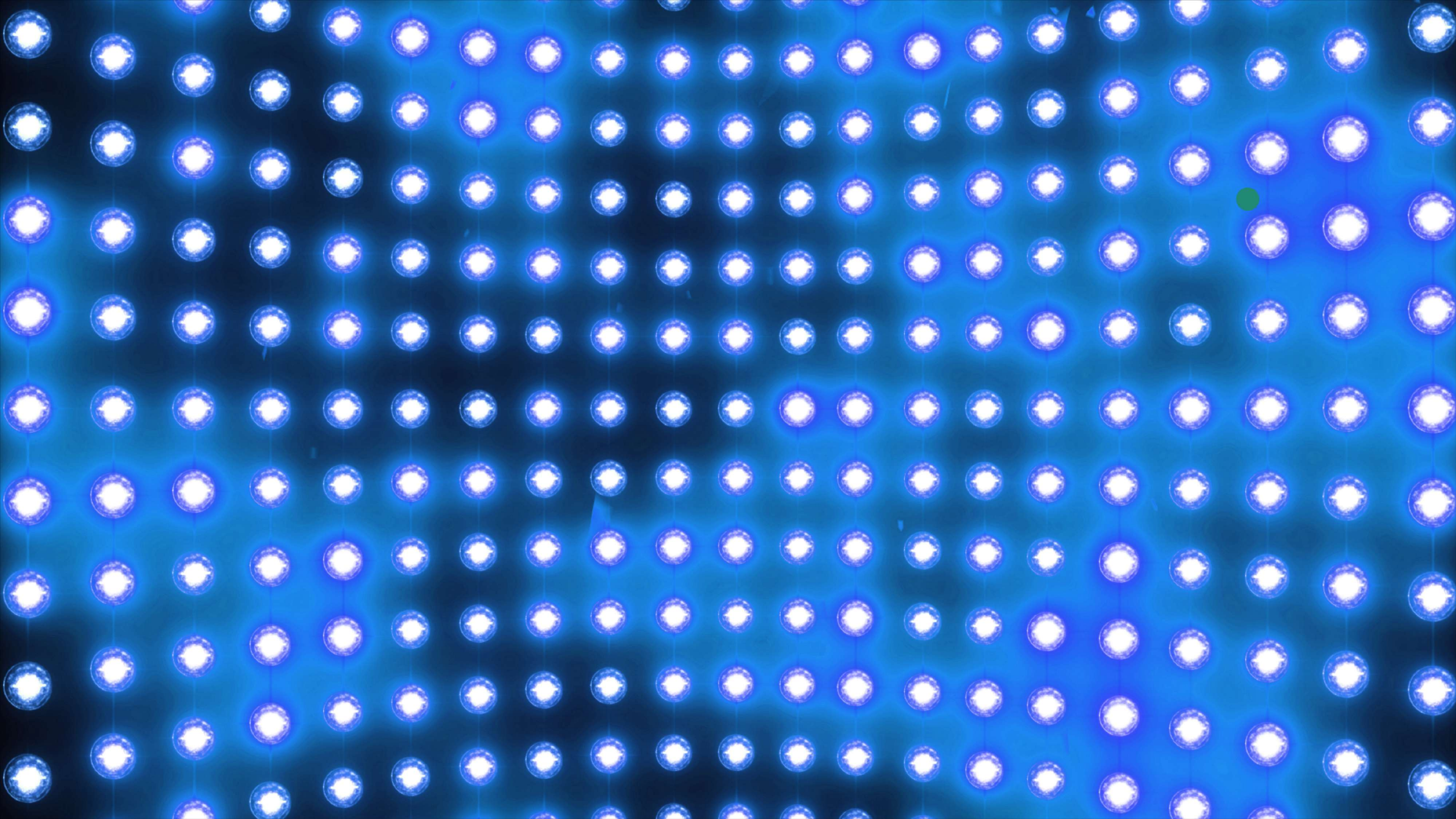
 @noctarius2k

 @noctarius2k@mastodon.online

 @noctarius.com







Question 01

**FAMILY
FEUD**

Question 01

Why you shouldn't run a database in Kubernetes?

**FAMILY
FEUD**

Why not to run a database in Kubernetes?



Christoph Engelbert / Noctarius ツ / エンゲルベルト クリス 
@noctarius2k

Why you ***SHOULD NOT*** run a database in Kubernetes?
What do you think? Please help me, I need you! 🙏❤️

PS: Asking for a friend! 😅

via: [#postgresql](#) [#mysql](#) [#mariadb](#) [#kafka](#) [#kubernetes](#)

Why not to run a database in Kubernetes?

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

Persistent Data will kill you! Too slow!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

Persistent Data will kill you! Too slow!

Nobody understands Kubernetes!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

Persistent Data will kill you! Too slow!

Nobody understands Kubernetes!

What's the benefit; databases don't need autoscaling!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

Persistent Data will kill you! Too slow!

Nobody understands Kubernetes!

What's the benefit; databases don't need autoscaling!

Databases and applications should be separated!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

Persistent Data will kill you! Too slow!

Nobody understands Kubernetes!

What's the benefit; databases don't need autoscaling!

Databases and applications should be separated!

Not another layer of indirection / abstraction!

Why not to run a database in Kubernetes?

K8s is not designed with Databases in mind!

Never run Stateful Workloads in k8s!

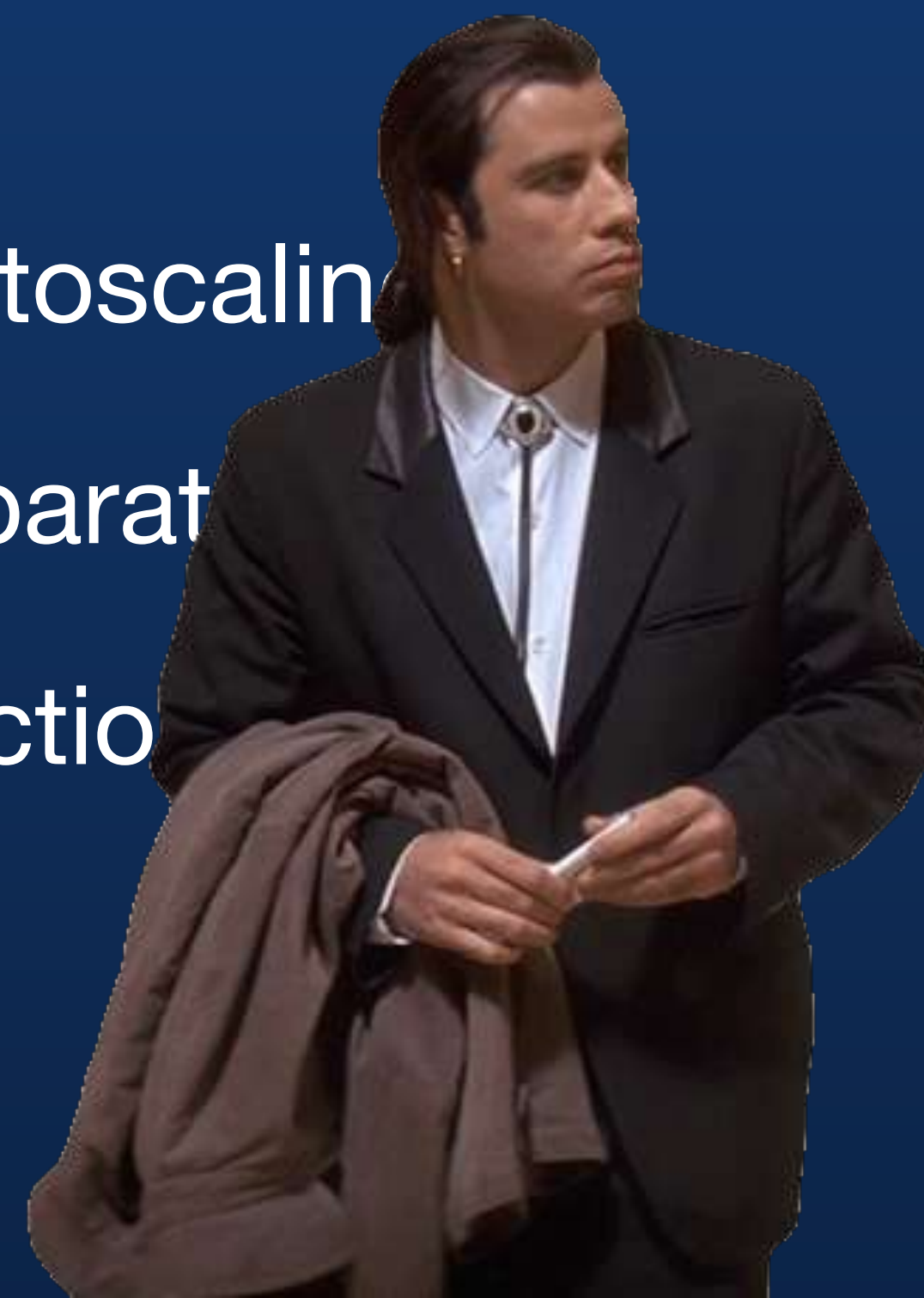
Persistent Data will kill you! Too slow!

Nobody understands Kubernetes!

What's the benefit; databases don't need autoscaling

Databases and applications should be separated

Not another layer of indirection / abstraction



BURN IN HELL!





The Happy Place

 Where are my gamers at?

So we need to cheat!?



B **A**

Why?



Why?

No Cloud-Vendor Lock-In



Why?

No Cloud-Vendor Lock-In

Faster Time To Market



Why?

No Cloud-Vendor Lock-In

Faster Time To Market

Decreasing cost



Why?

No Cloud-Vendor Lock-In

Faster Time To Market

Decreasing cost

Automation



Why?

No Cloud-Vendor Lock-In

Faster Time To Market

Decreasing cost

Automation

Unified deployment architecture



Why?

No Cloud-Vendor Lock-In

Faster Time To Market

Decreasing cost

Automation

Unified deployment architecture

Need read-only replicas



Let's get something
out of the way first!

 Call the Police!



Call the Police!



Enable TLS

 Call the Police!



Enable TLS
Use Kubernetes Secrets

Call the Police!



Enable TLS

Use Kubernetes Secrets

Use Cert-Manager

Call the Police!



Enable TLS

Use Kubernetes Secrets

Use Cert-Manager

Encrypt Data-At-Rest



Call the Police!



Enable TLS

Use Kubernetes Secrets

Use Cert-Manager

Encrypt Data-At-Rest

Backup and Recovery



Backup and Recovery

You want Continuous Backup and PITR



Backup and Recovery

You want Continuous Backup and PITR

Roll your own pg_basebackup or pg_dump (don't!)



Backup and Recovery

You want Continuous Backup and PITR

Roll your own `pg_basebackup` or `pg_dump` (don't!)

Use tools like `pgbackrest`, `barman`, `PGHoard`, ...



Backup and Recovery

You want Continuous Backup and PITR

Roll your own `pg_basebackup` or `pg_dump` (don't!)

Use tools like `pgbackrest`, `barman`, `PGHoard`, ...

Upload backups to S3? Cost!



Backup and Recovery

You want Continuous Backup and PITR
Roll your own pg_basebackup or pg_dump (don't!)
Use tools like pgbackrest, barman, PGHoard, ...
Upload backups to S3? Cost!

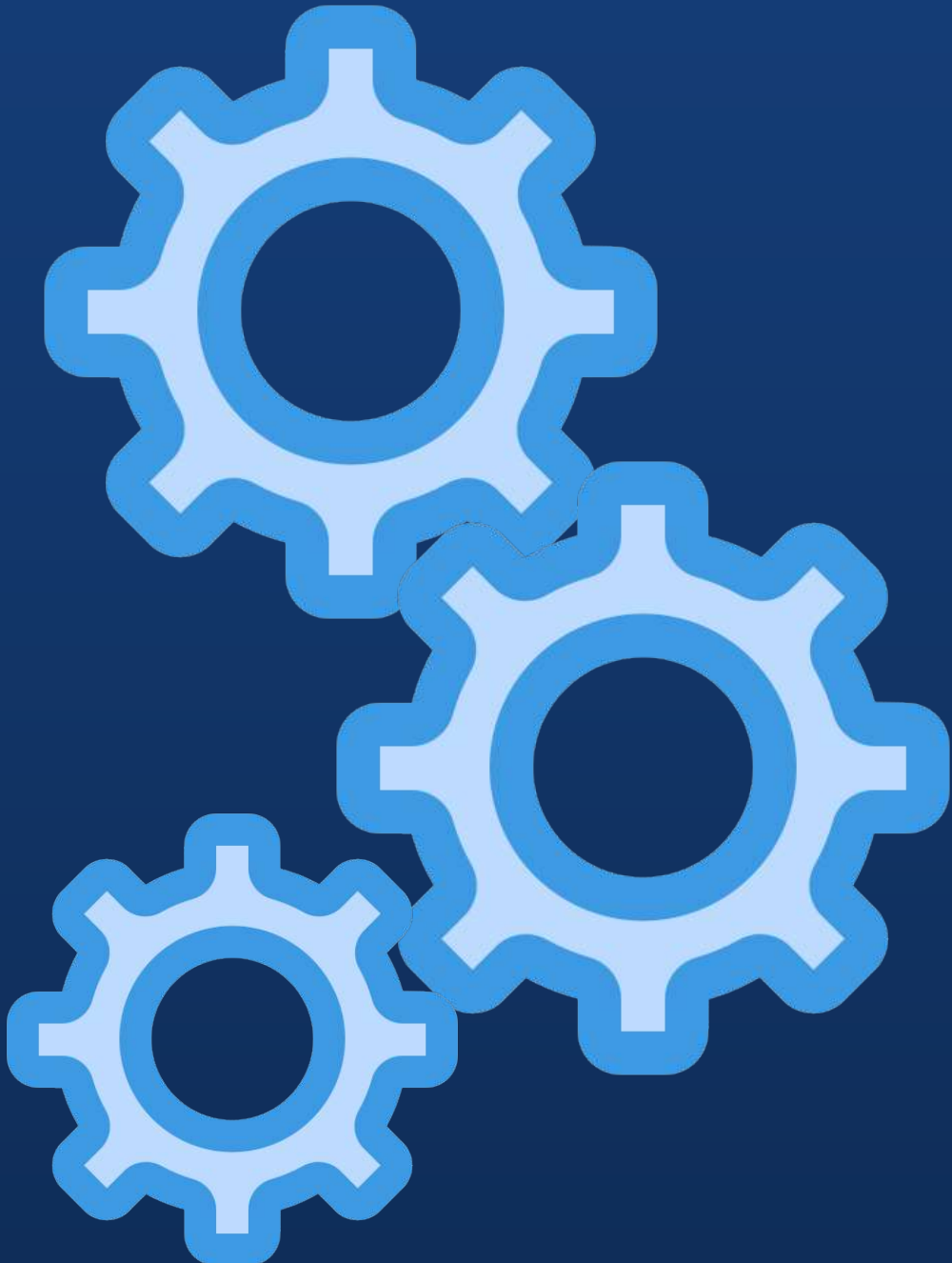
😅 Test Your Backups 😅

PostgreSQL Configuration

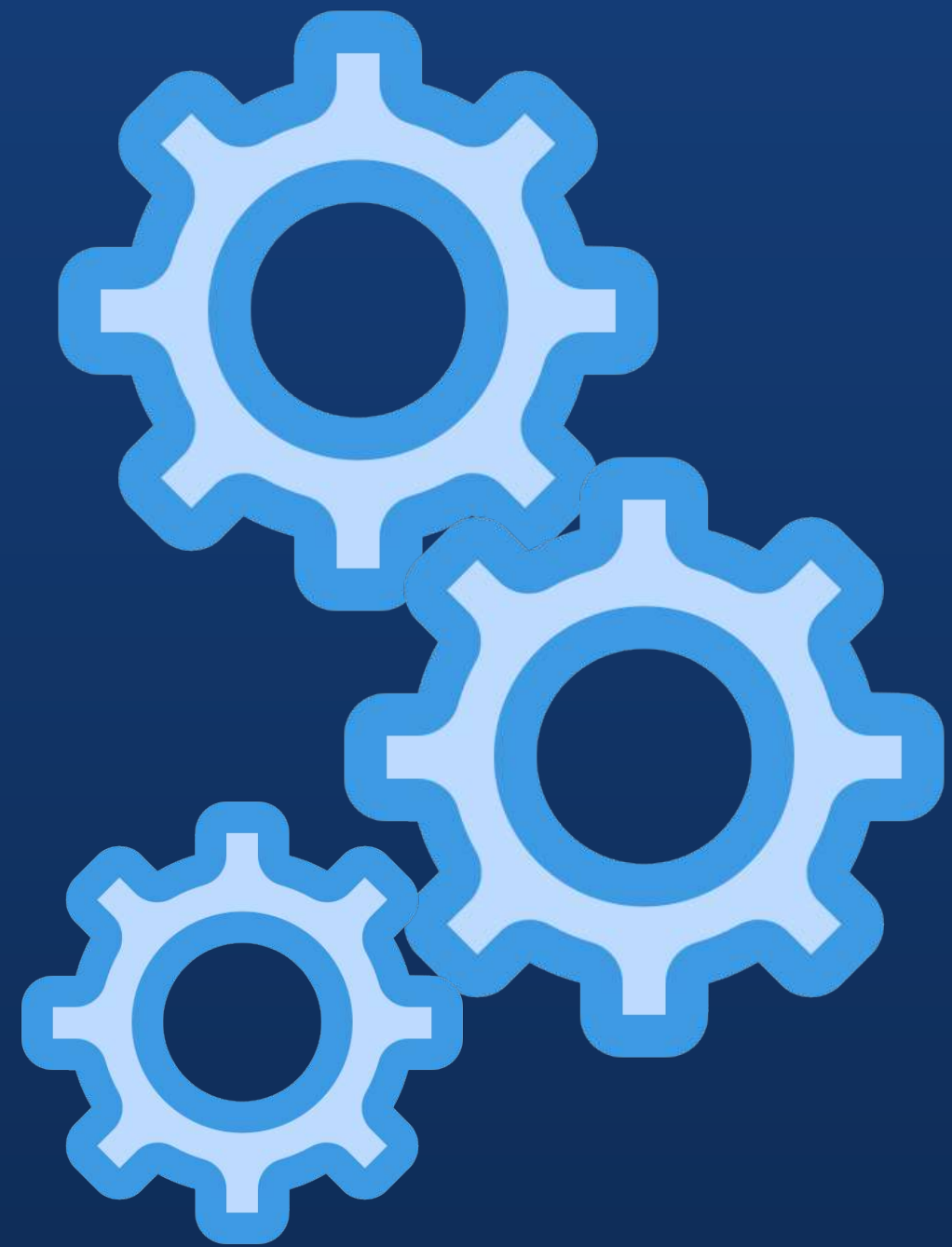


PostgreSQL Configuration

The PostgreSQL Configuration isn't too much influenced



PostgreSQL Configuration



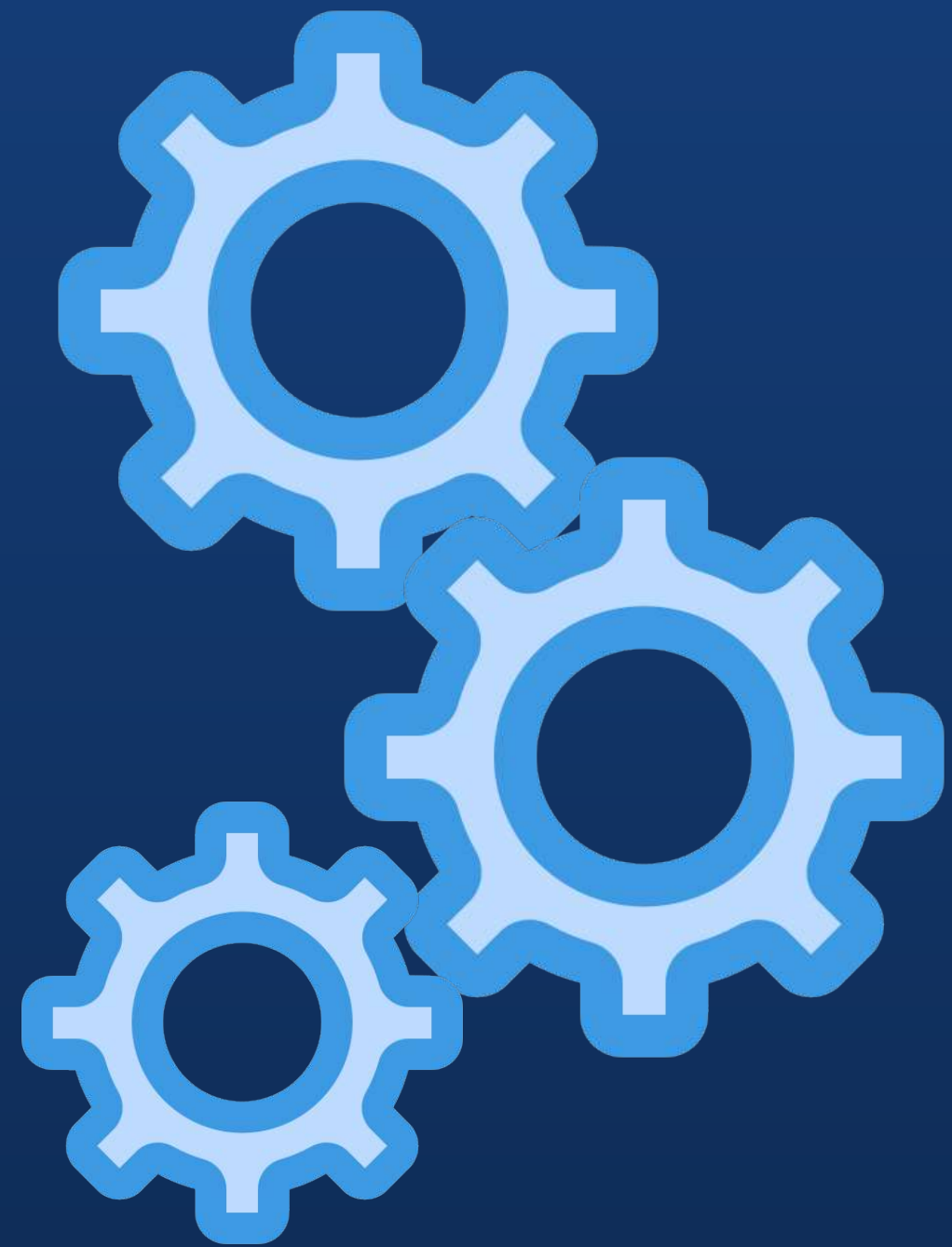
The PostgreSQL Configuration isn't too much influenced

`shared_buffers`

`(maintenance_)work_mem`

`effective_cache_size`

PostgreSQL Configuration



The PostgreSQL Configuration isn't too much influenced

`shared_buffers`

`(maintenance_)work_mem`

`effective_cache_size`

Use Huge Pages!

PostgreSQL Configuration

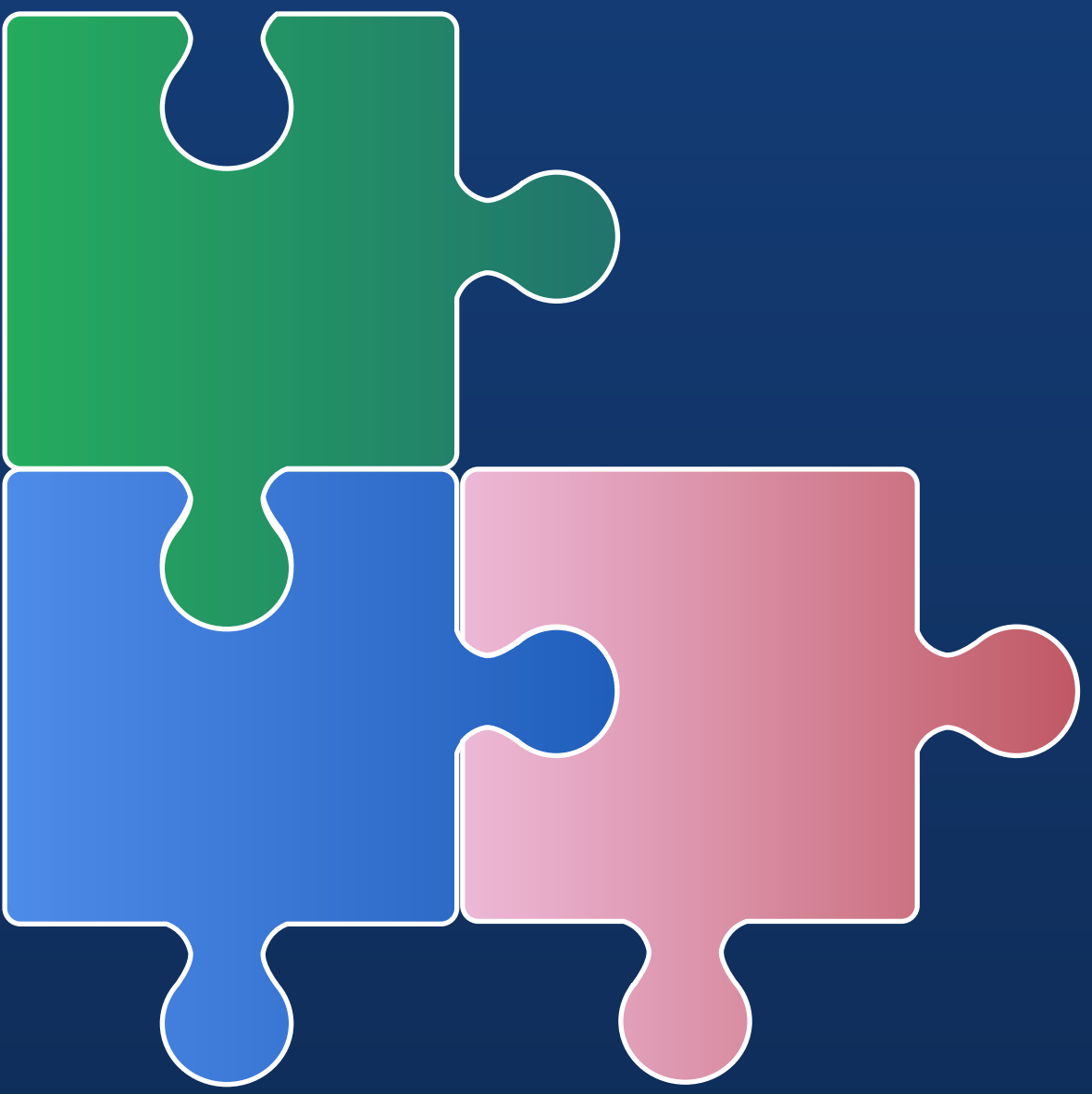


<https://www.youtube.com/watch?v=S0LEDGbAnn8>

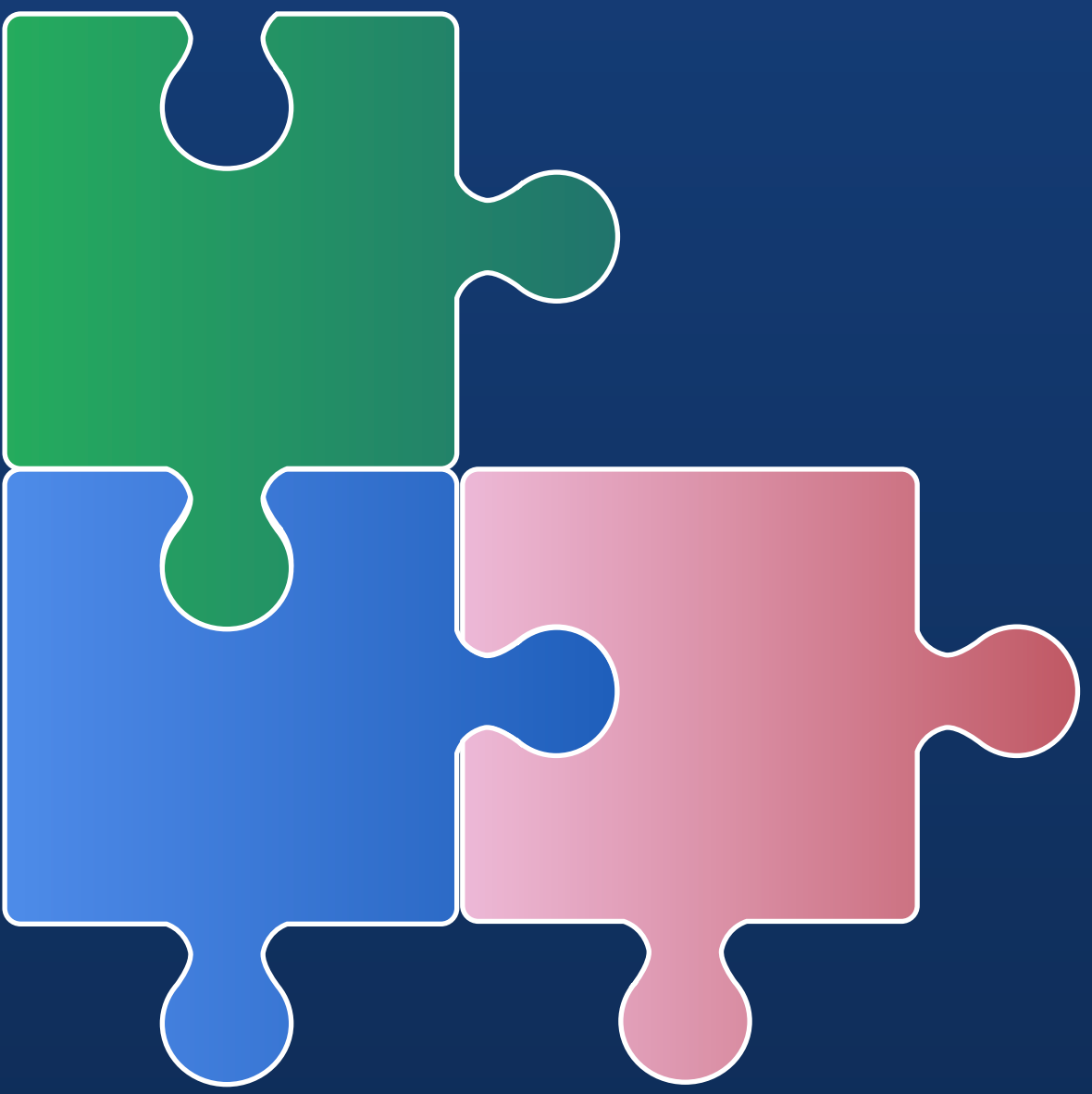
<https://www.crunchydata.com/blog/optimize-postgresql-server-performance>

<https://www.percona.com/blog/using-huge-pages-with-postgresql-running-inside-kubernetes/>

Do you need more? Extensions!

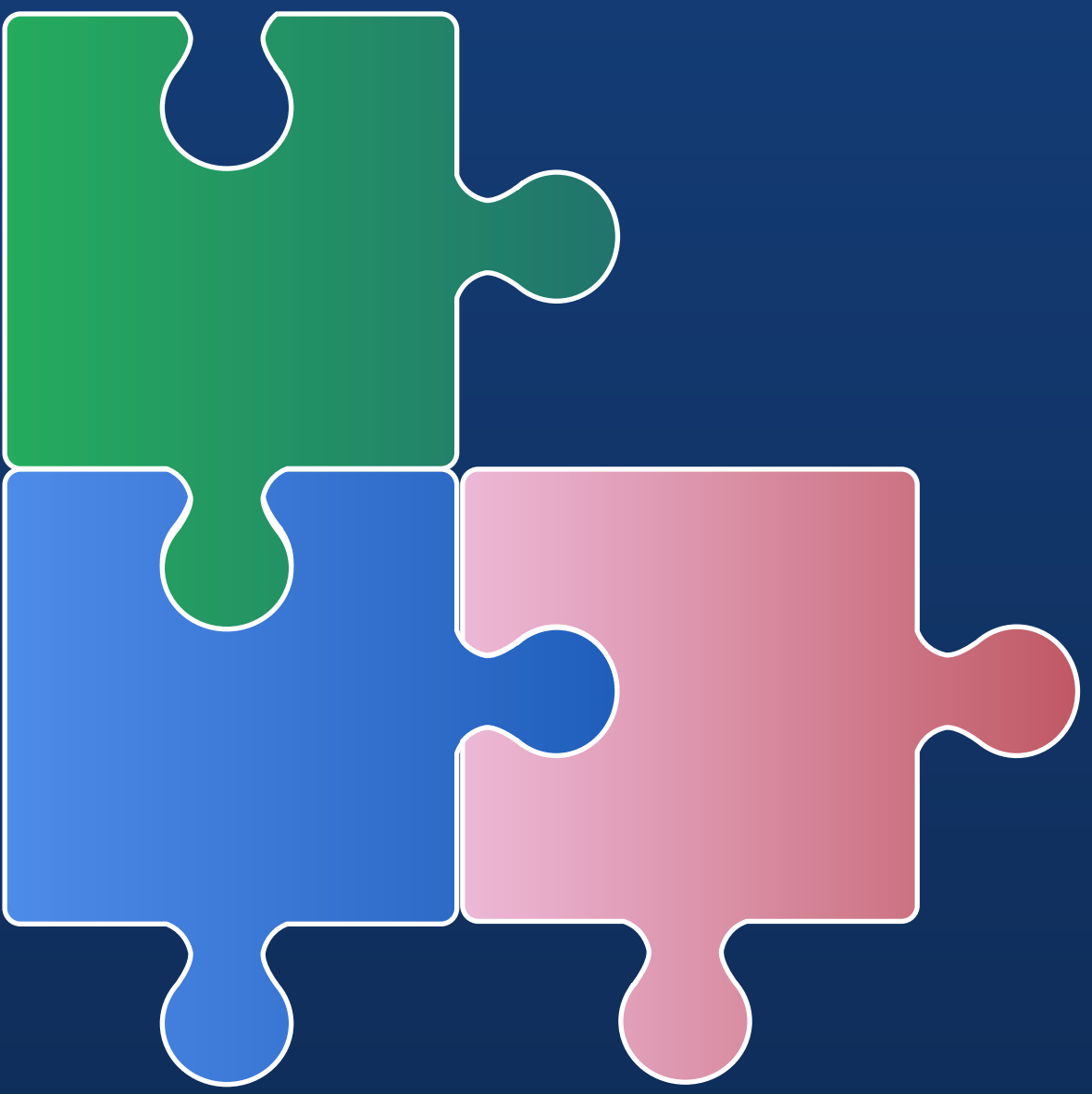


Do you need more? Extensions!



Do you need PG Extensions?

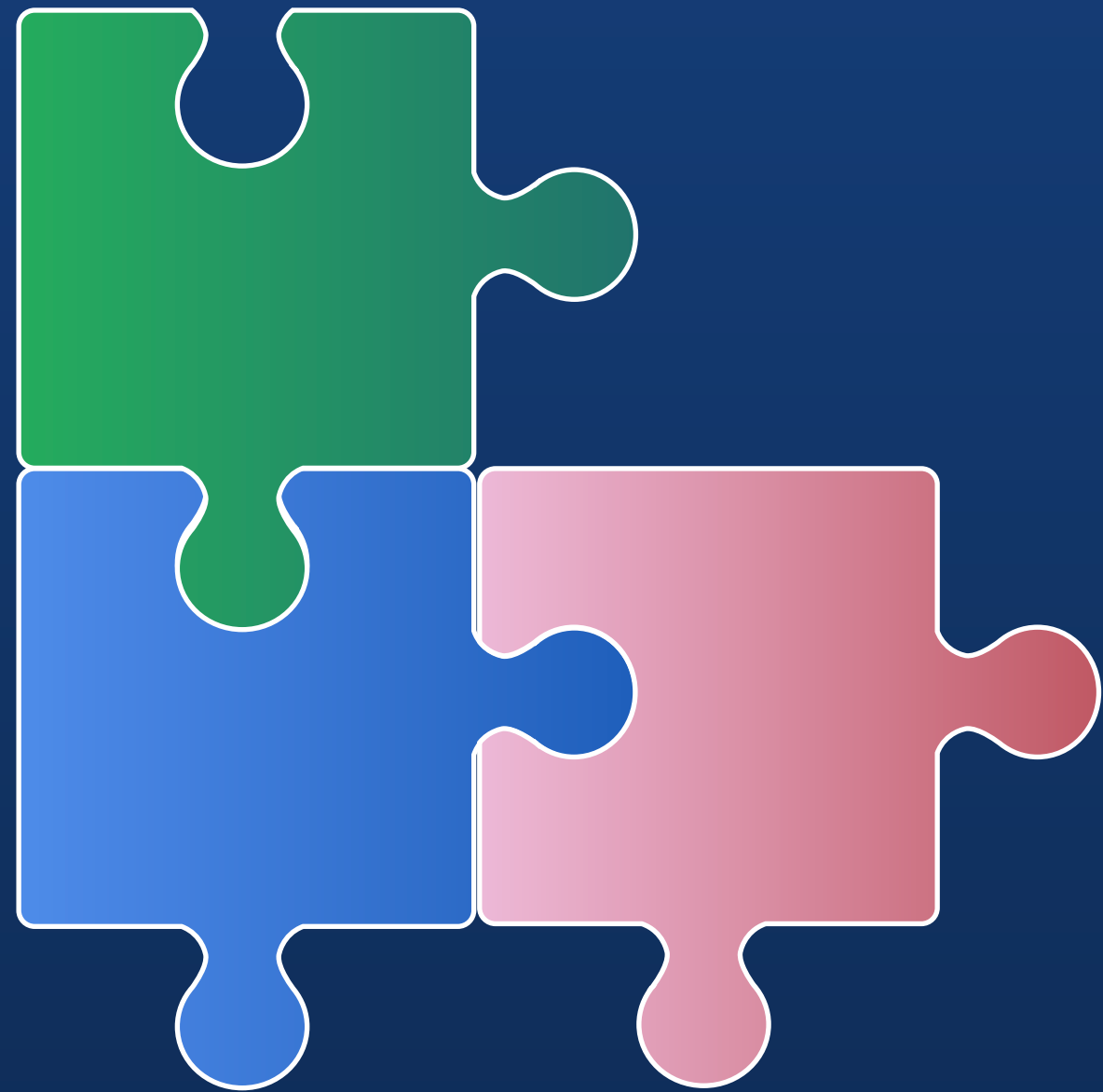
Do you need more? Extensions!



Do you need PG Extensions?

Is the extension part of the container image?

Do you need more? Extensions!

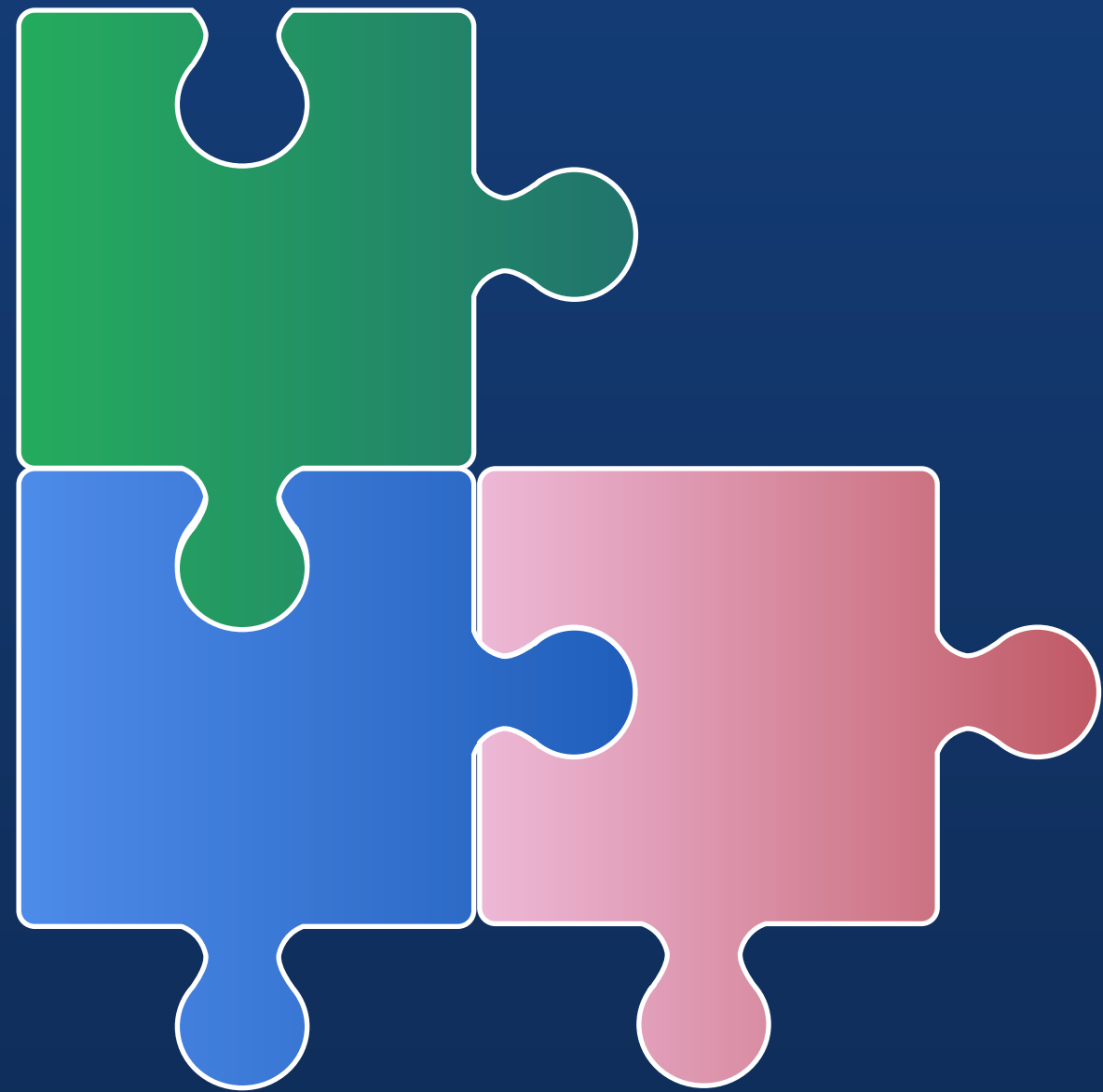


Do you need PG Extensions?

Is the extension part of the container image?

If not, you need to build your own layer...

Do you need more? Extensions!



Do you need PG Extensions?

Is the extension part of the container image?

If not, you need to build your own layer...

or use some magic (more on this later).

Keep an Eye on PG and Kubernetes Versions

So *What* is important or different?



Use Persistent Volumes



Use Persistent Volumes
(local volumes are a bad idea)





Use Persistent Volumes
(local volumes are a bad idea)

Should be dynamically provisioned



Use Persistent Volumes
(local volumes are a bad idea)

Should be dynamically provisioned

CSI provider enables encryption at rest



Use Persistent Volumes
(local volumes are a bad idea)

Should be dynamically provisioned

CSI provider enables encryption at rest

High IOPS (SSD or NVMe)



Use Persistent Volumes
(local volumes are a bad idea)

Should be dynamically provisioned

CSI provider enables encryption at rest

High IOPS (SSD or NVMe)

Low Latency



Use Persistent Volumes
(local volumes are a bad idea)

Should be dynamically provisioned

CSI provider enables encryption at rest

High IOPS (SSD or NVMe)

Low Latency

Database performance is as fast as your storage



Use Persistent Volumes
(local volumes are a bad idea)

Should be dynamically provisioned

CSI provider enables encryption at rest

High IOPS (SSD or NVMe)

Low Latency

Database performance is as fast as your storage


I'd recommend a disaggregated storage!

CSI Driver Listing

Searchable listing of Kubernetes volume providers (Kubernetes CSI). Find your perfect implementation based on features, access modes, and more.

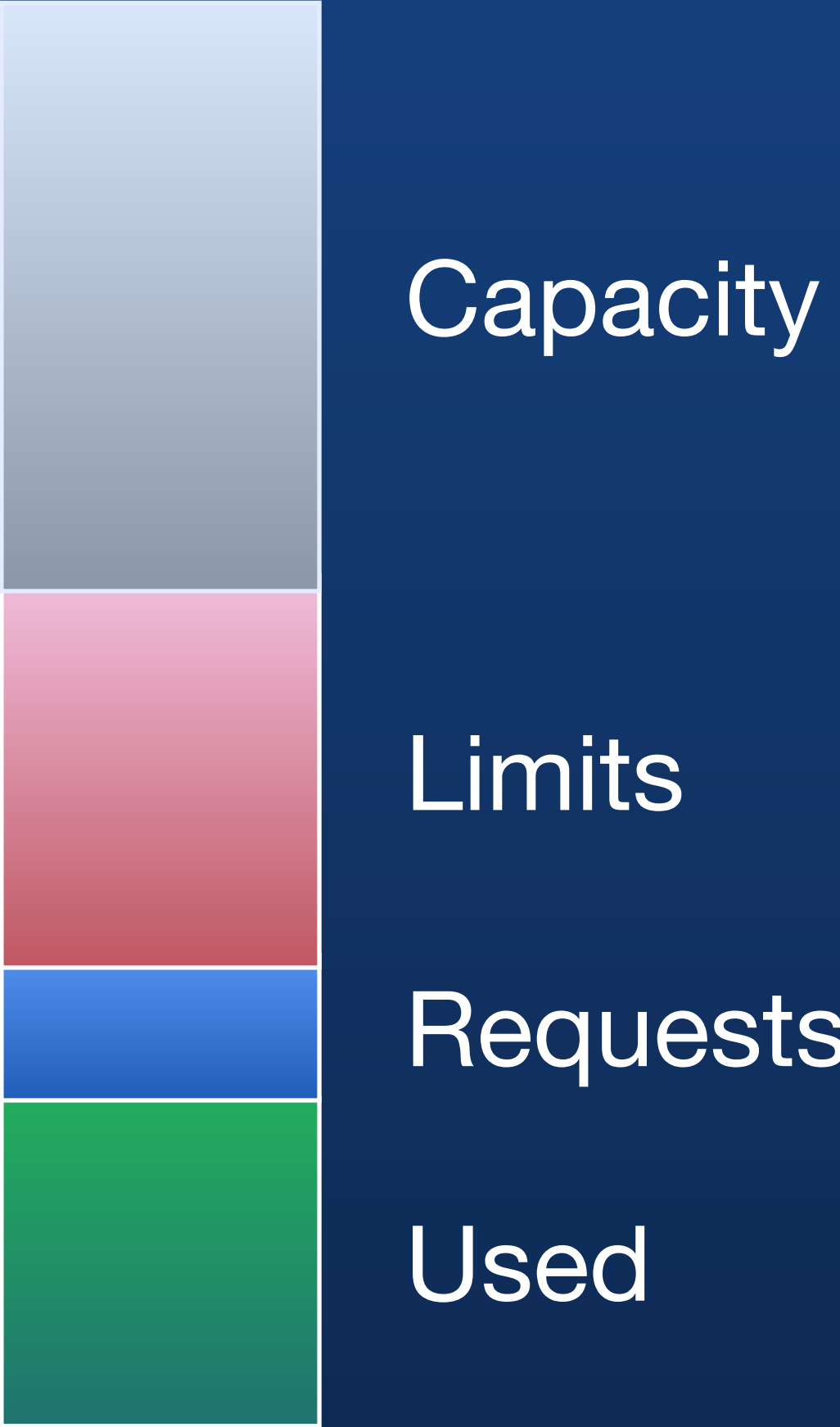
The following table contains a list of available Kubernetes container storage interface (CSI) implementations. All CSI drivers are listed with their respective capabilities. The list is updated bit by bit according to the documentation on a best effort basis. The list may not be complete and properties may be incorrect or incomplete. Vendors of the CSI drivers are welcome to validate and update their respective properties.

SearchTerm: Lifecycle Modes: Capabilities: Access Modes: Filter Deprecated:
Filter result: 136/150

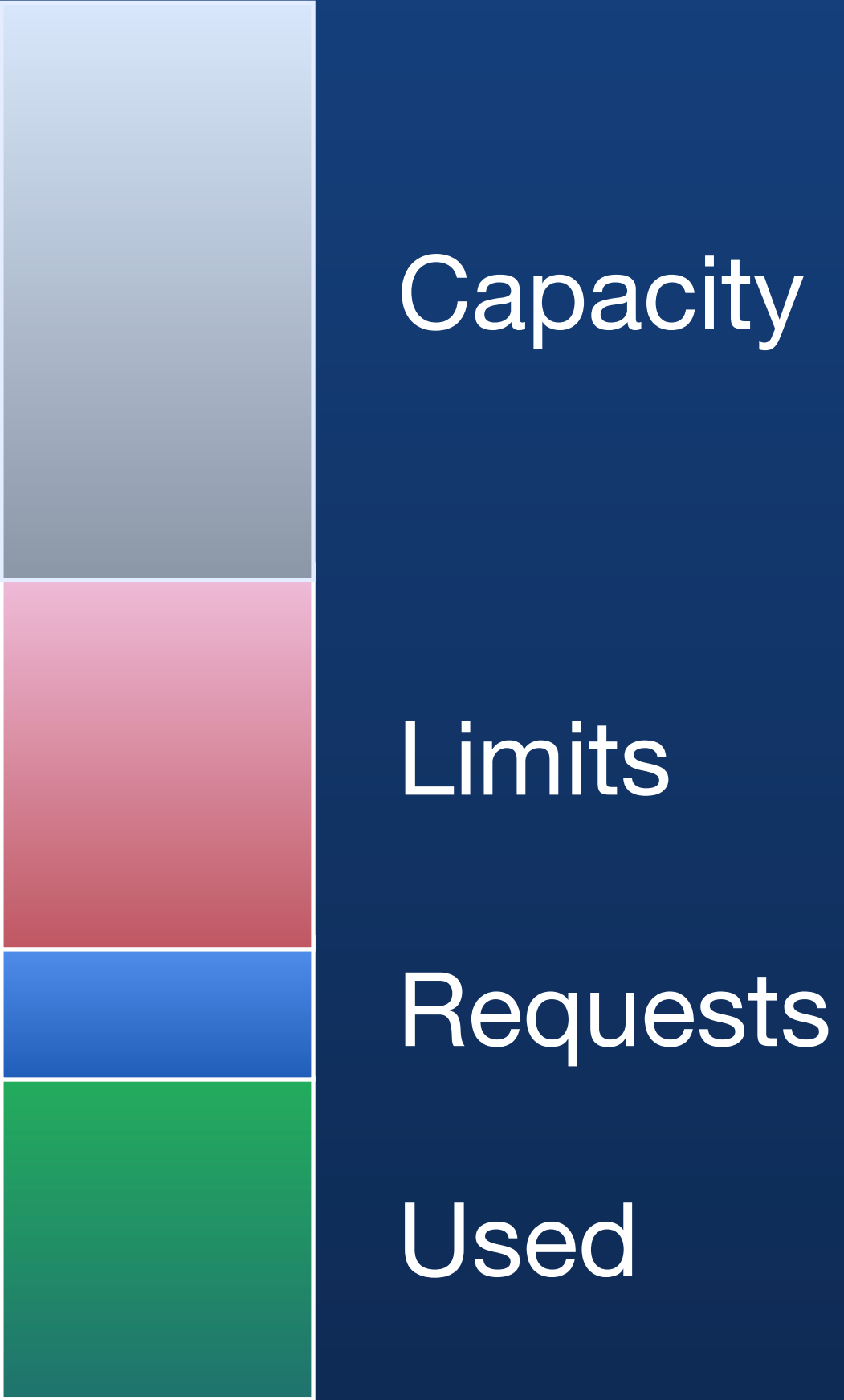
Driver Name	Driver Class	Description	Lifecycle Modes	Supported Capabilities	Access Modes
 AlibabaCloud Disk	diskplugin.csi.alibabacloud.com	CSI Driver for an AlibabaCloud Disk	<ul style="list-style-type: none">• Persistent: ✓• Ephemeral: ✗	<ul style="list-style-type: none">• Block Storage: ✓• File Storage: ✗• Object Storage: ✗• Dynamic: ✓• Snapshot: ✓• Clones: ✗• Expansion: ✓• Topology: ✓• Tracking: ✗	<ul style="list-style-type: none">• Read Only Many: ✓• Read Write Once: ✓• Read Write Many: ✗• Read Write Once Pod: ✓

www.storageclass.info/csdrivers

Requests, Limits, and Quotas

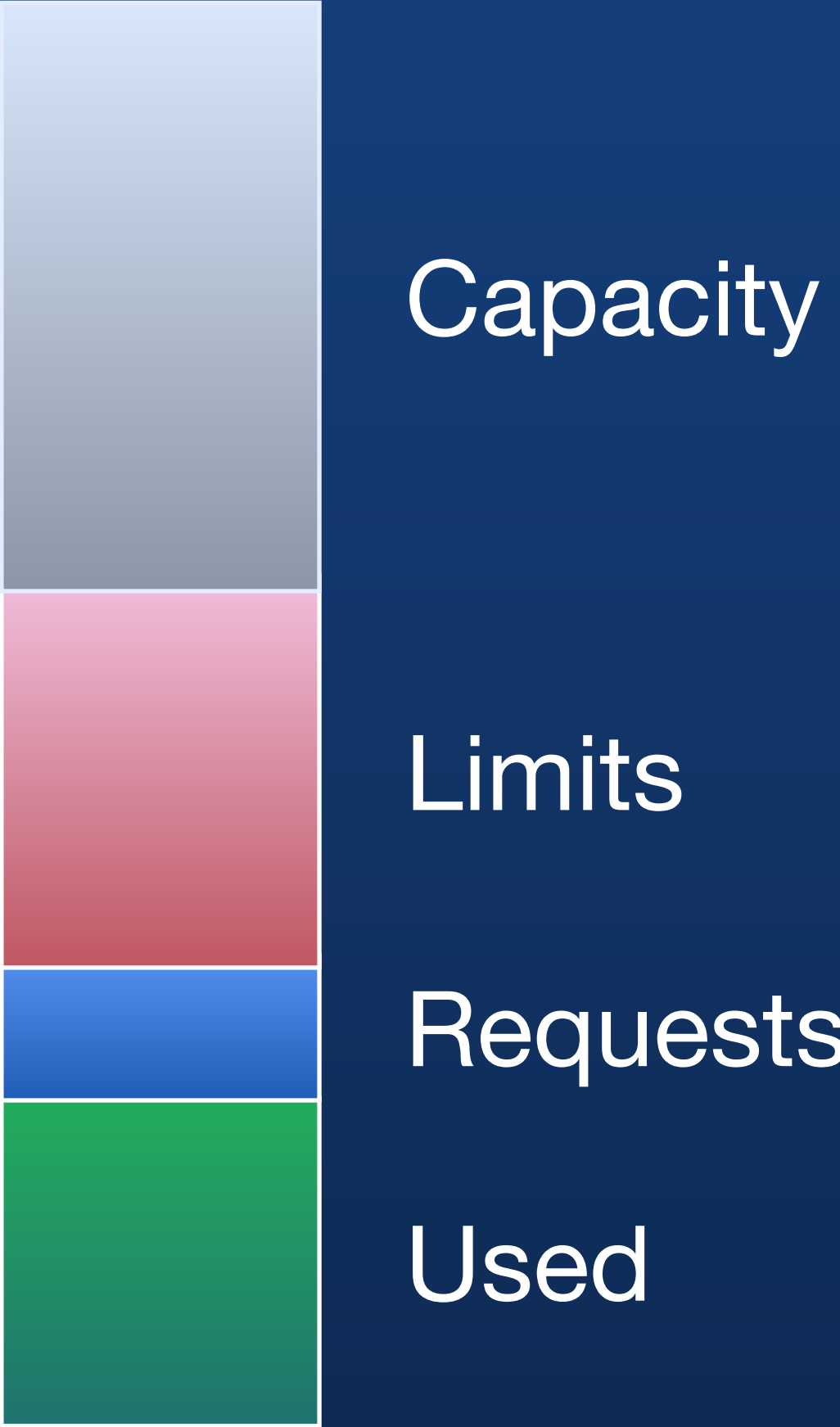


Requests, Limits, and Quotas



Use Resource Requests, Limits, Quotas

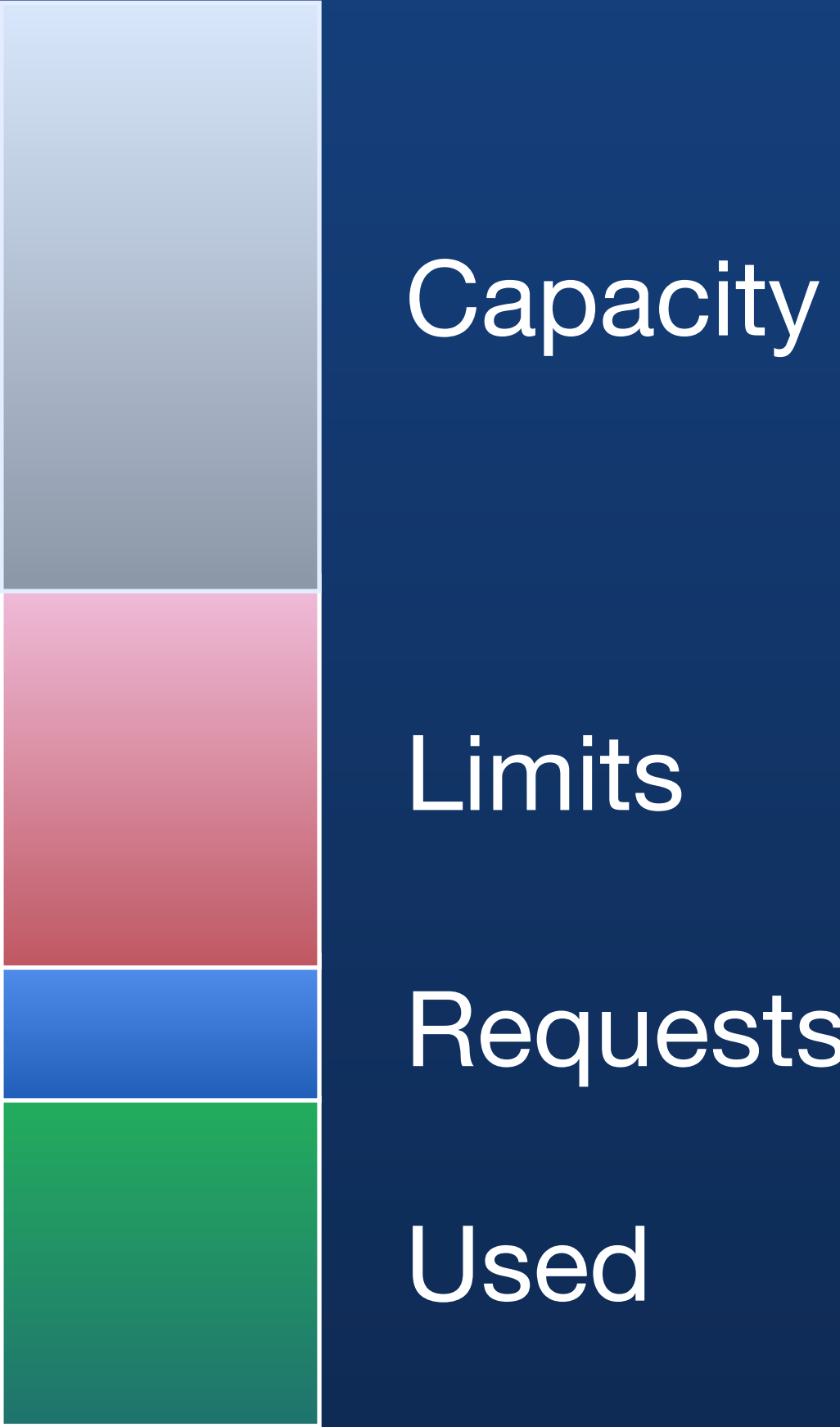
Requests, Limits, and Quotas



Use Resource Requests, Limits, Quotas

CPU and memory requests need to be accurate to prevent contention and ensure predictable performance

Requests, Limits, and Quotas



Use Resource Requests, Limits, Quotas

CPU and memory requests need to be accurate to prevent contention and ensure predictable performance

<https://codimite.ai/blog/kubernetes-resources-and-scaling-a-beginners-guide/>

 **Make it big!**

Enable Huge Pages!

 **Make it big!**

Enable Huge Pages!

In your OS and the Resource Descriptor.

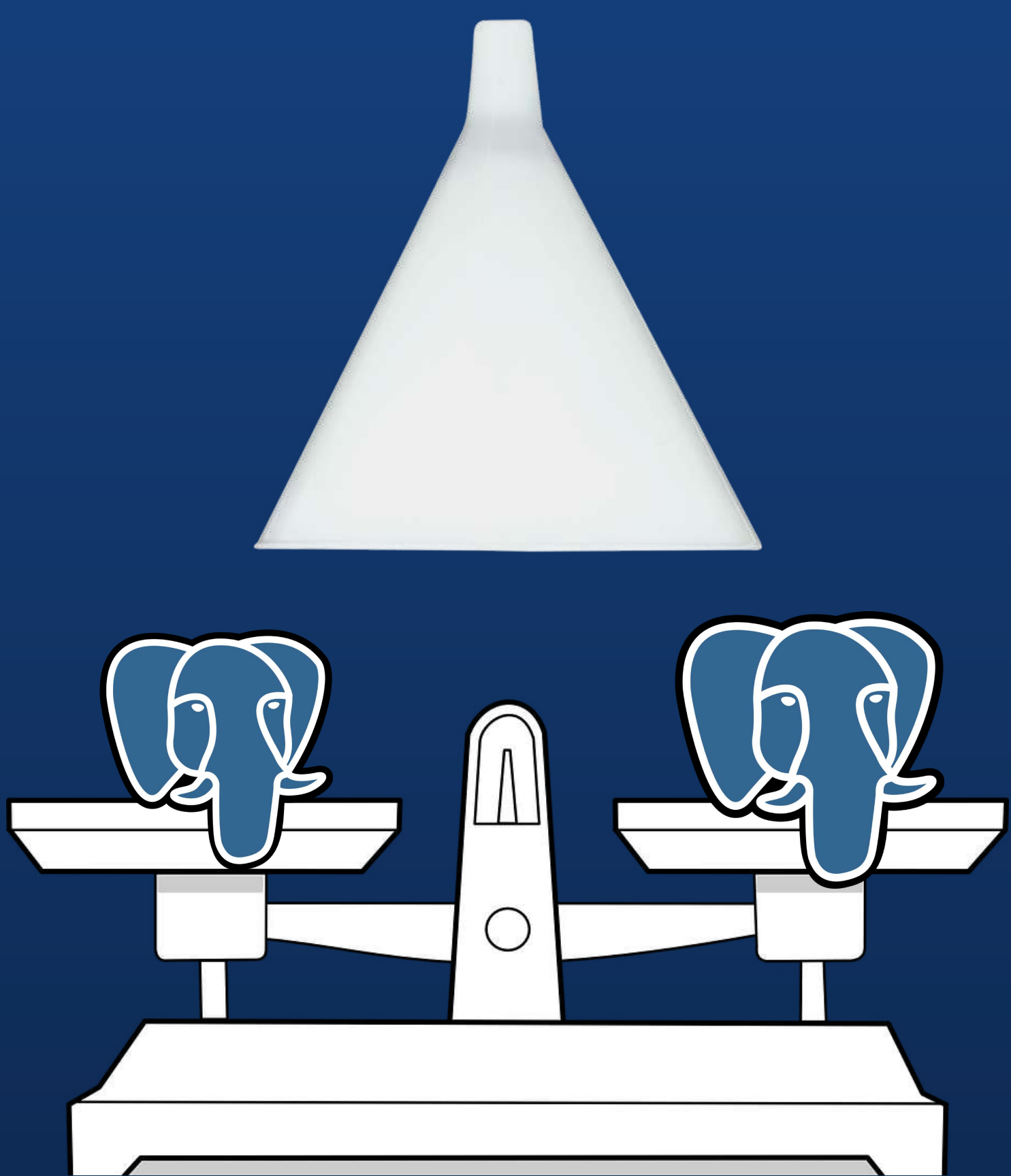
 **Make it big!**

Enable Huge Pages!

In your OS and the Resource Descriptor.

<https://www.percona.com/blog/using-huge-pages-with-postgresql-running-inside-kubernetes/>

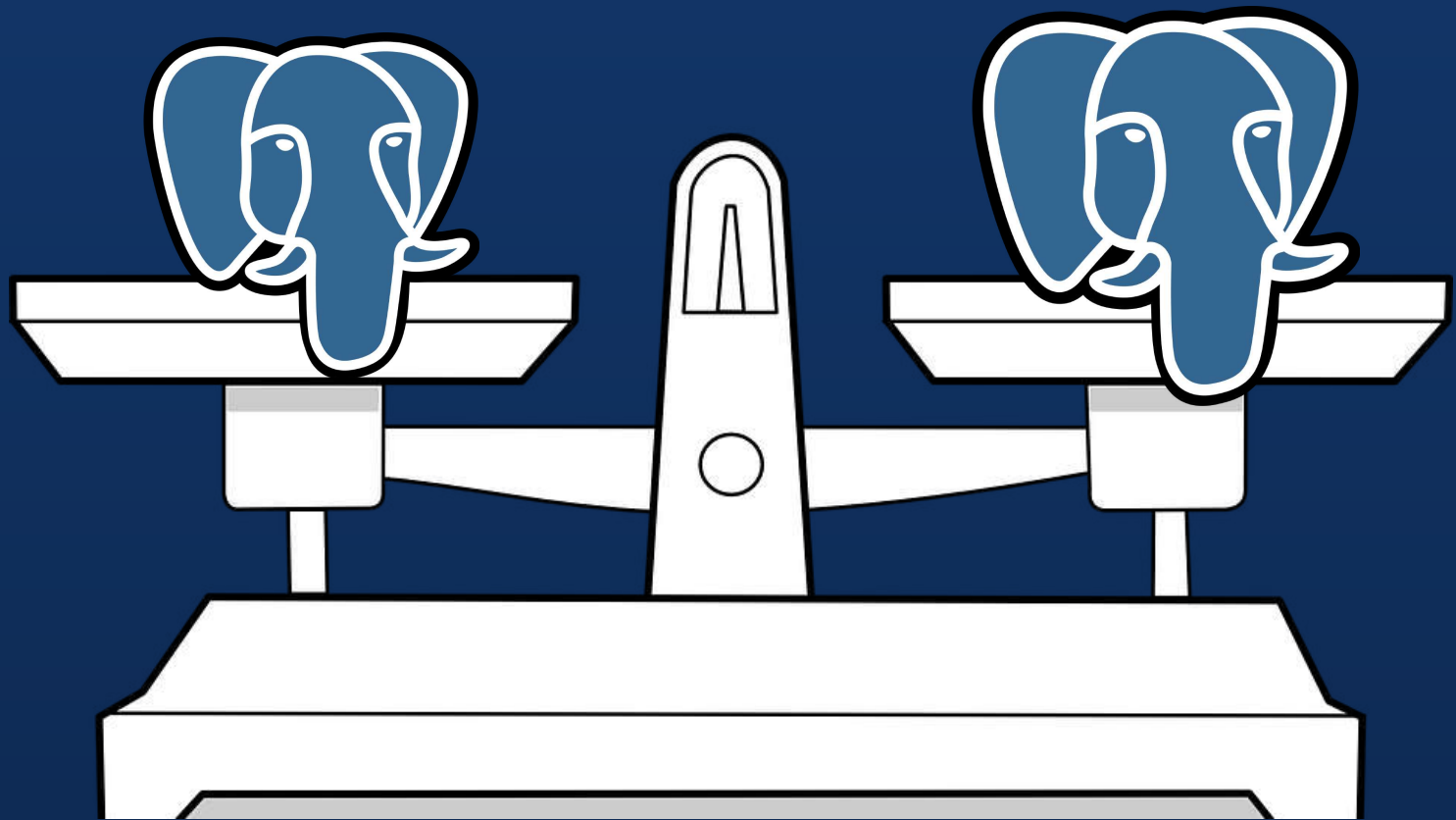
Resiliency and Overhead



Resiliency and Overhead



High Availability

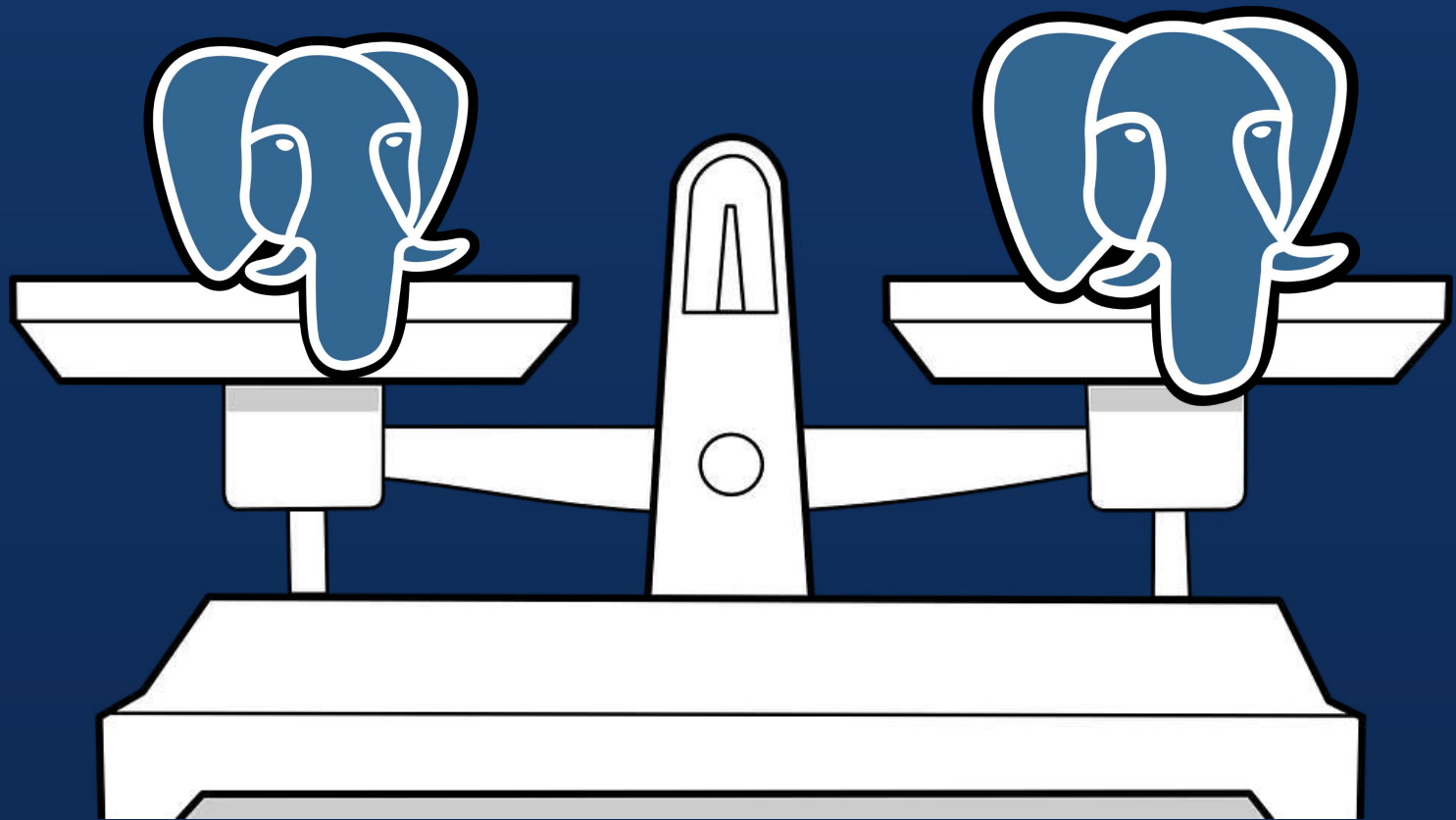


Resiliency and Overhead

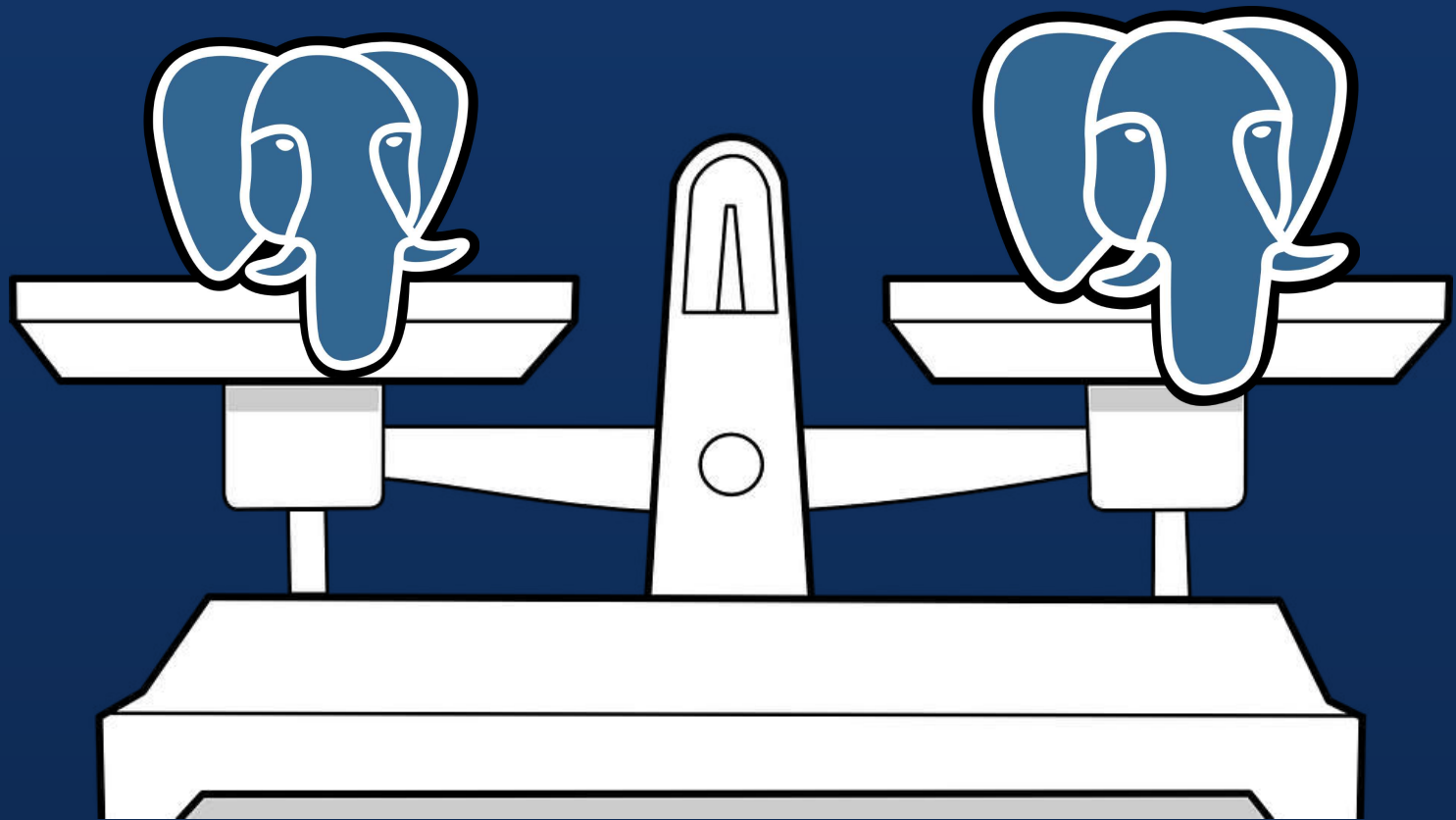


High Availability

Patroni, repmgr, pg_auto_failover, ...



Resiliency and Overhead

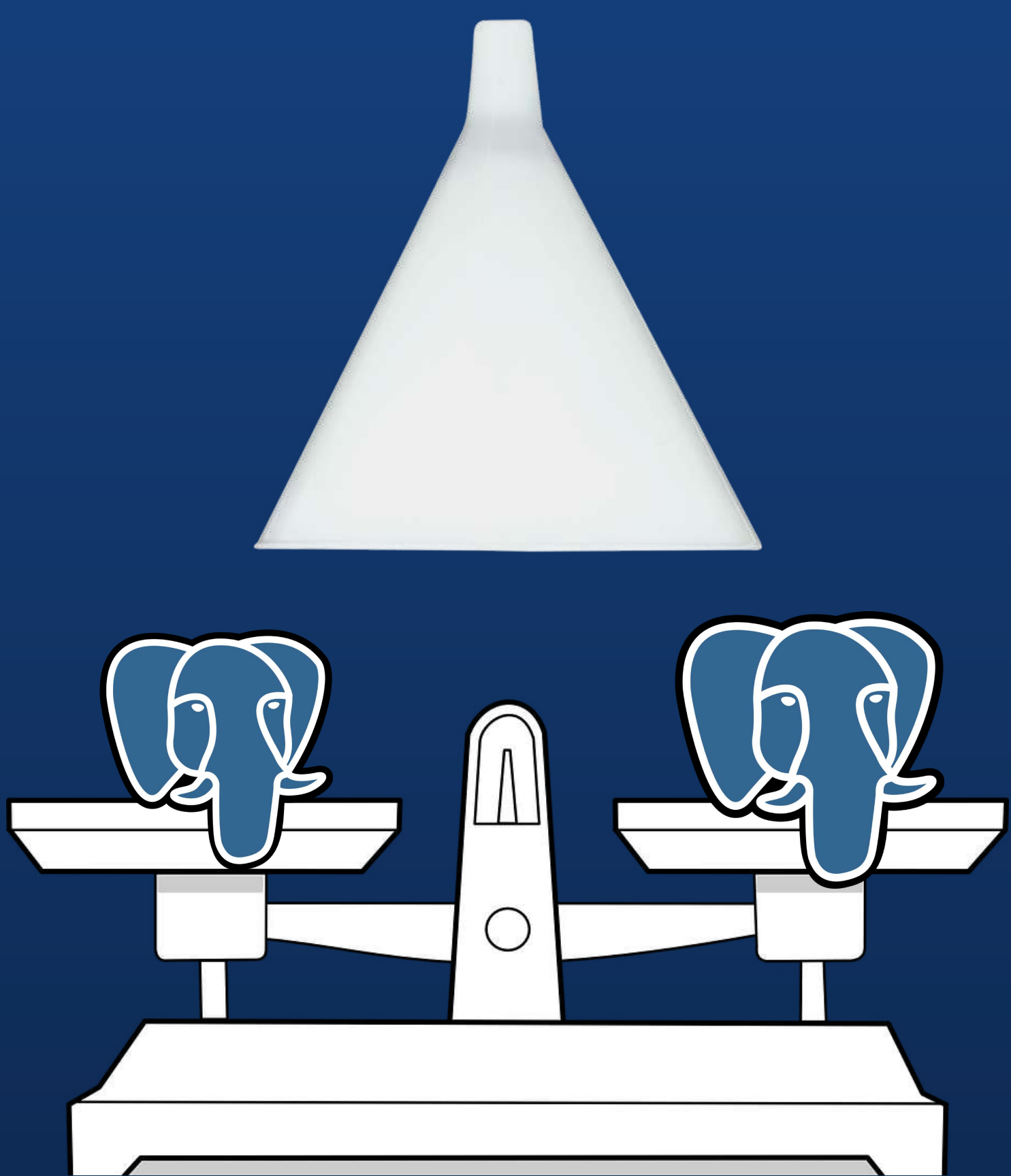


High Availability

Patroni, repmgr, pg_auto_failover, ...

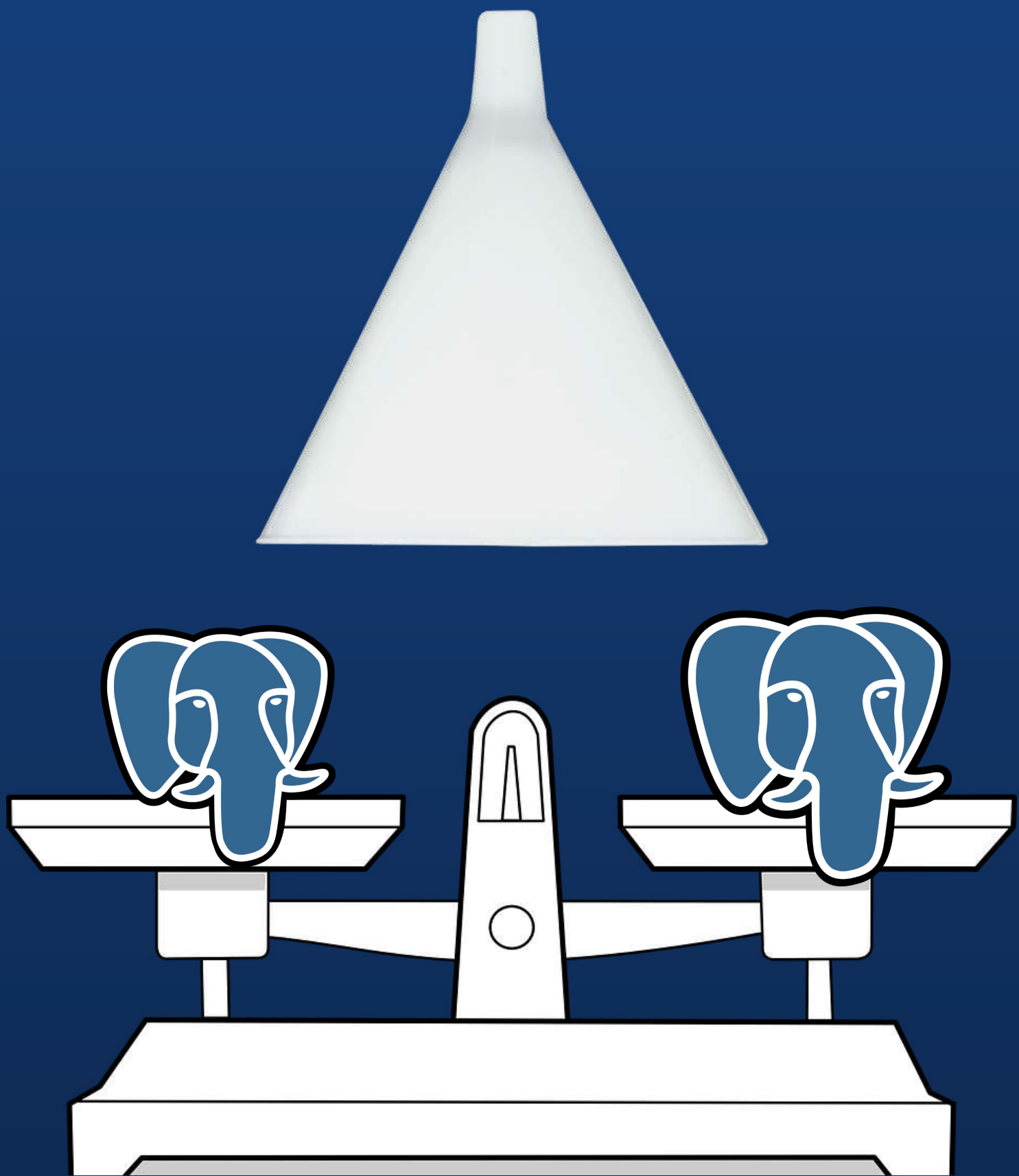
<https://medium.com/@kristi.anderson/whats-the-best-postgresql-high-availability-framework...>

Resiliency and Overhead



Resiliency and Overhead

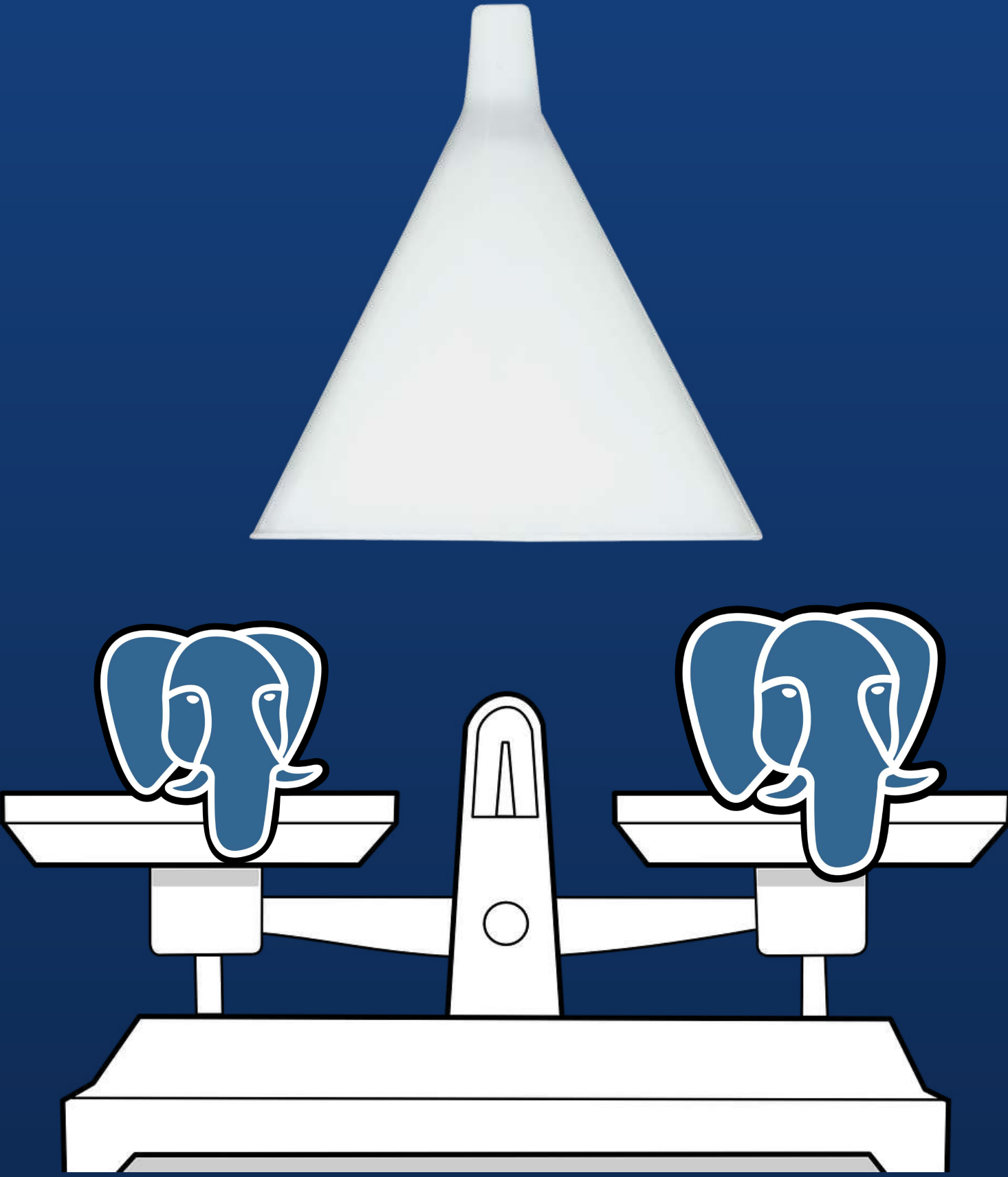
Connection Pooling



Resiliency and Overhead

Connection Pooling

Never use PostgreSQL without Connection Pooling!

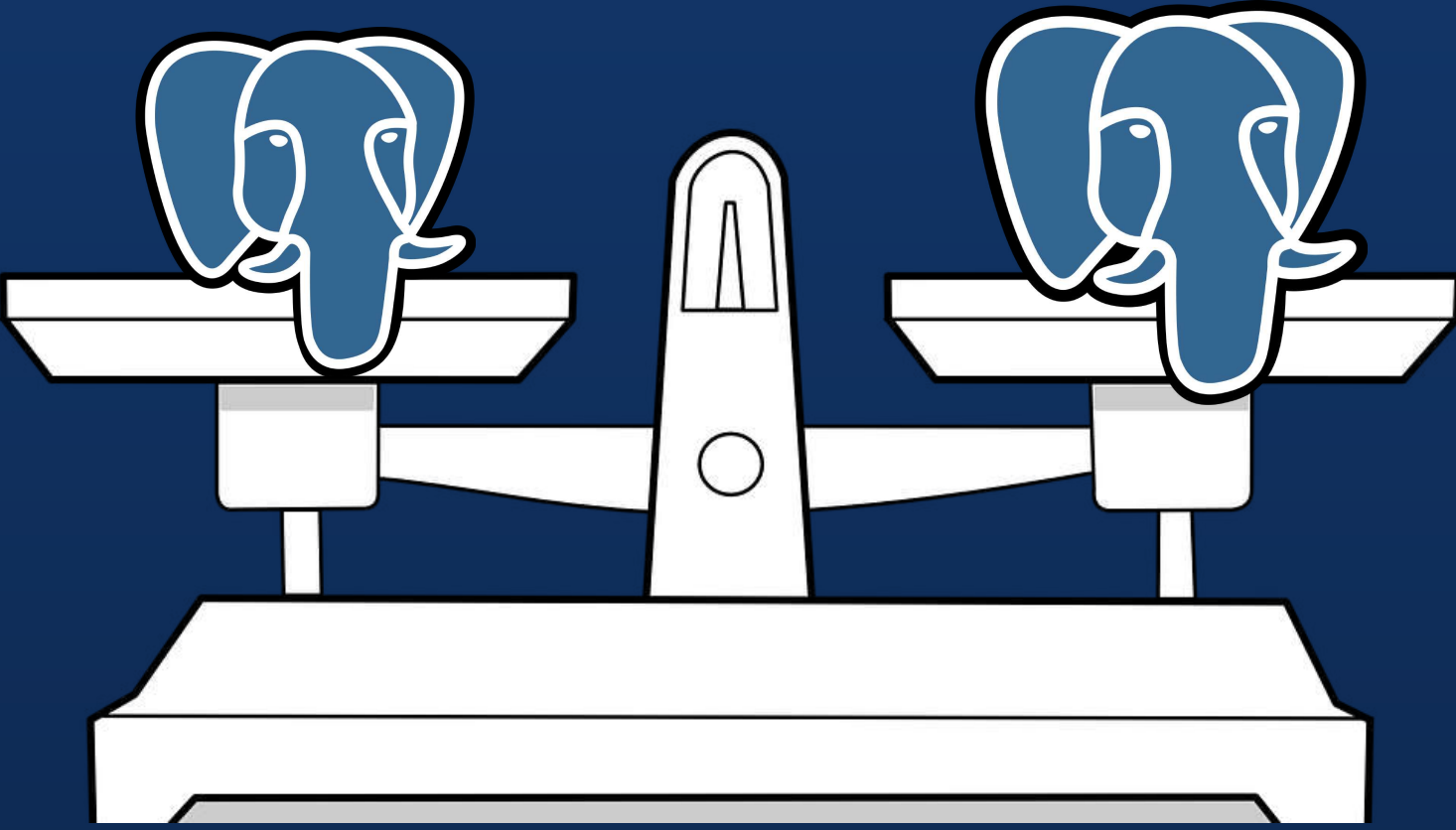


Resiliency and Overhead

Connection Pooling

Never use PostgreSQL without Connection Pooling!

Optimizes Overhead and Resource Utilization



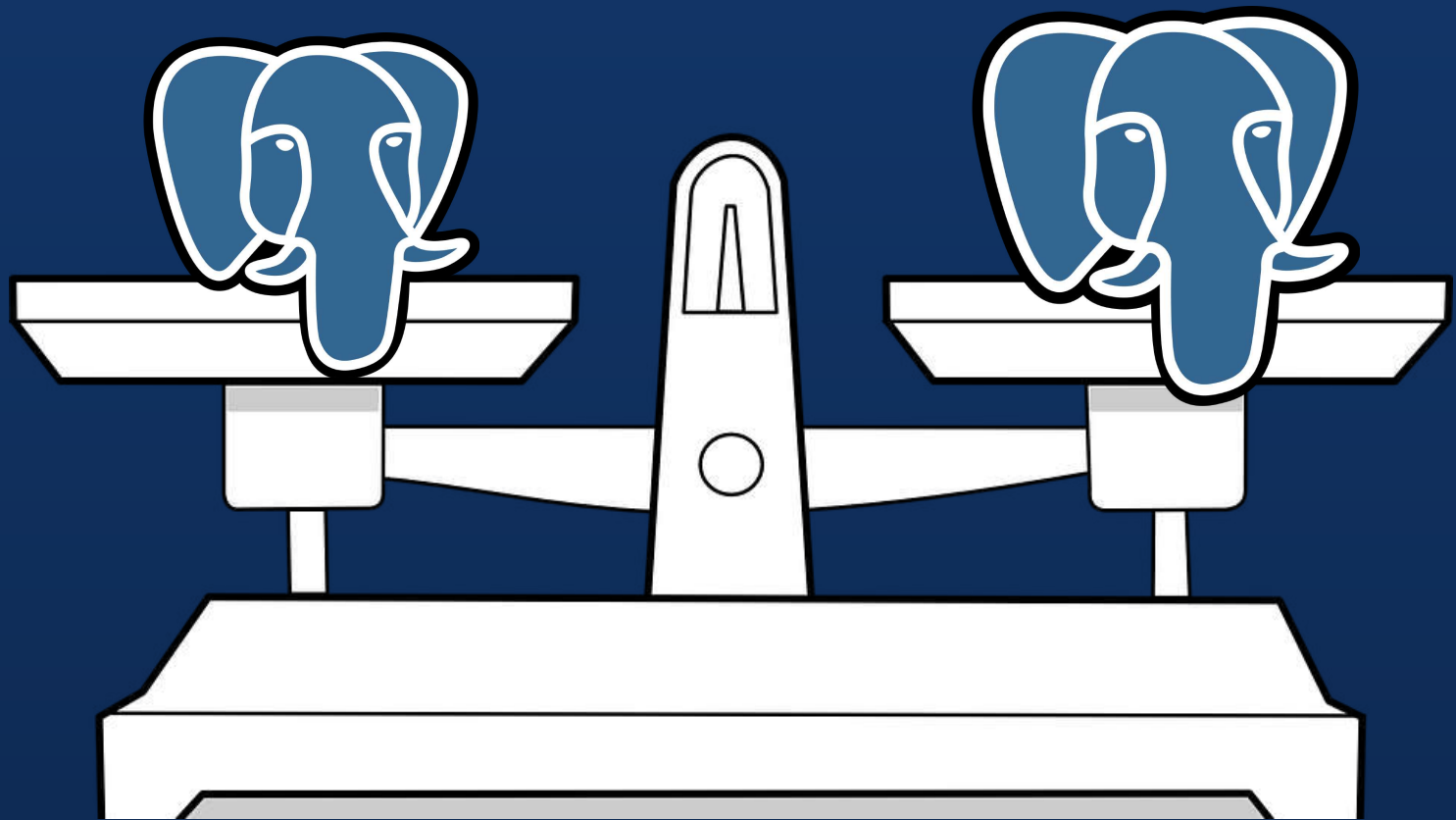
Resiliency and Overhead

Connection Pooling

Never use PostgreSQL without Connection Pooling!

Optimizes Overhead and Resource Utilization

Handles failovers, central switching of Primary



Resiliency and Overhead

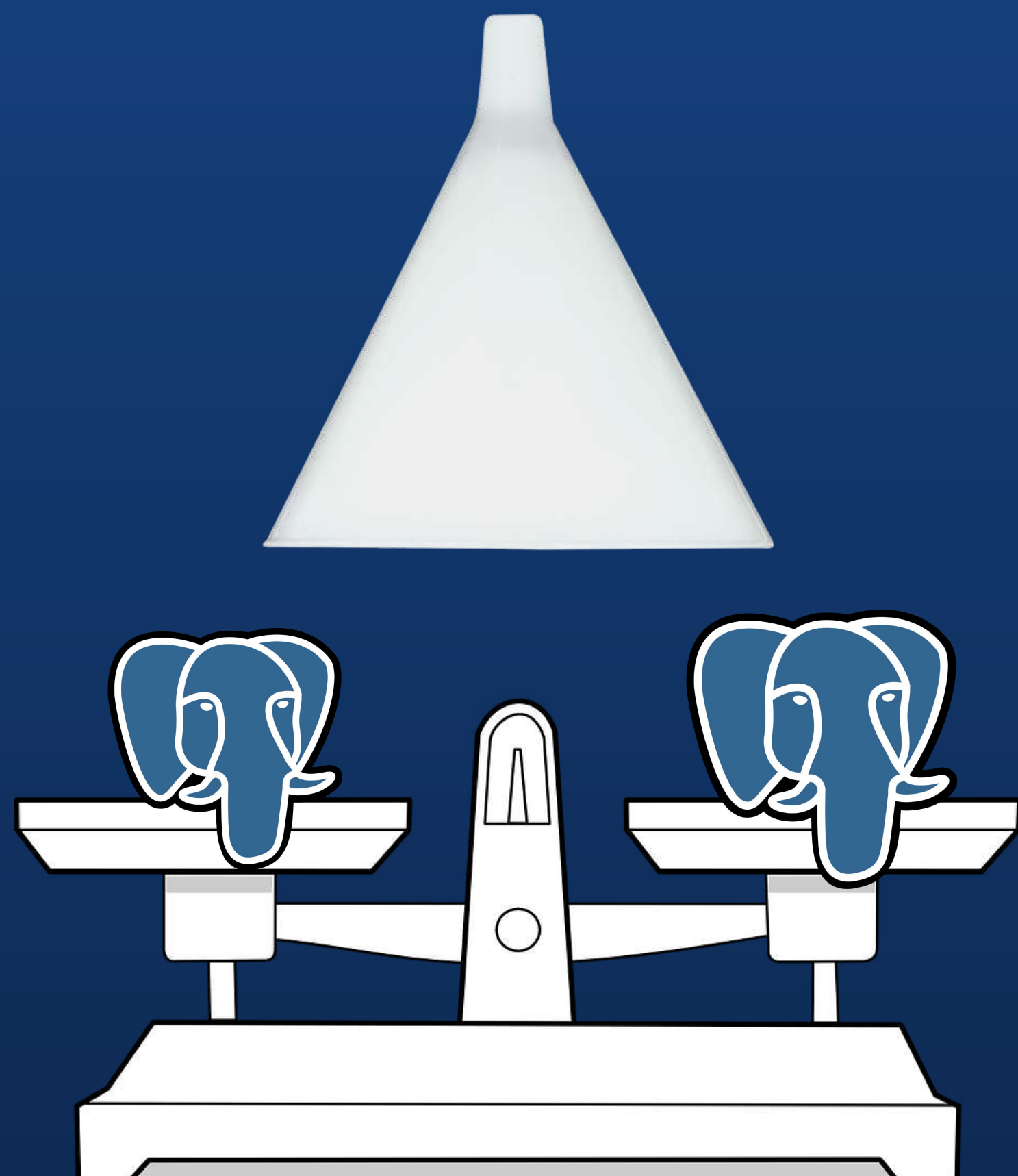
Connection Pooling

Never use PostgreSQL without Connection Pooling!

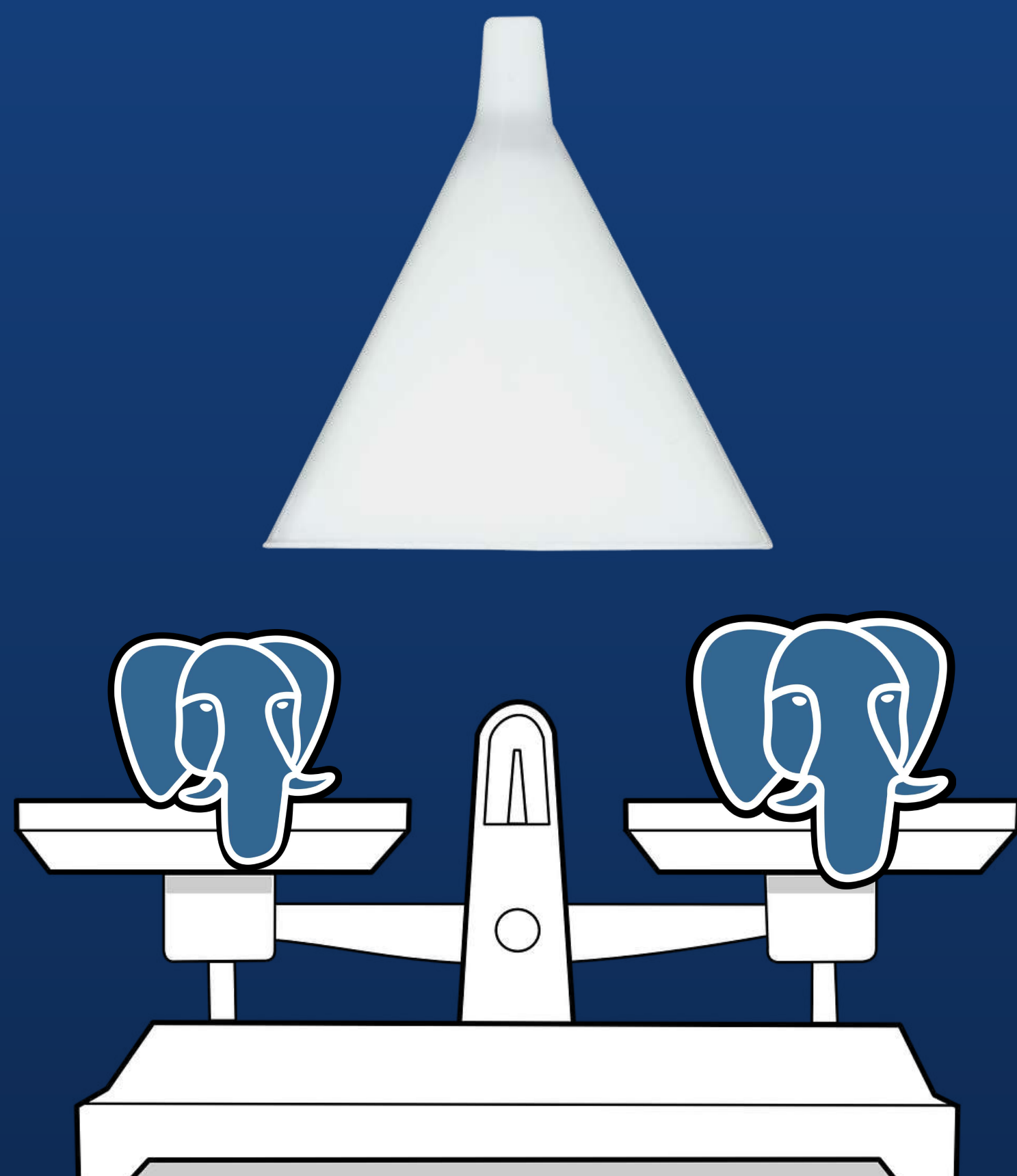
Optimizes Overhead and Resource Utilization

Handles failovers, central switching of Primary

Enables easy use of Read-Replicas



Resiliency and Overhead



Connection Pooling

Never use PostgreSQL without Connection Pooling!

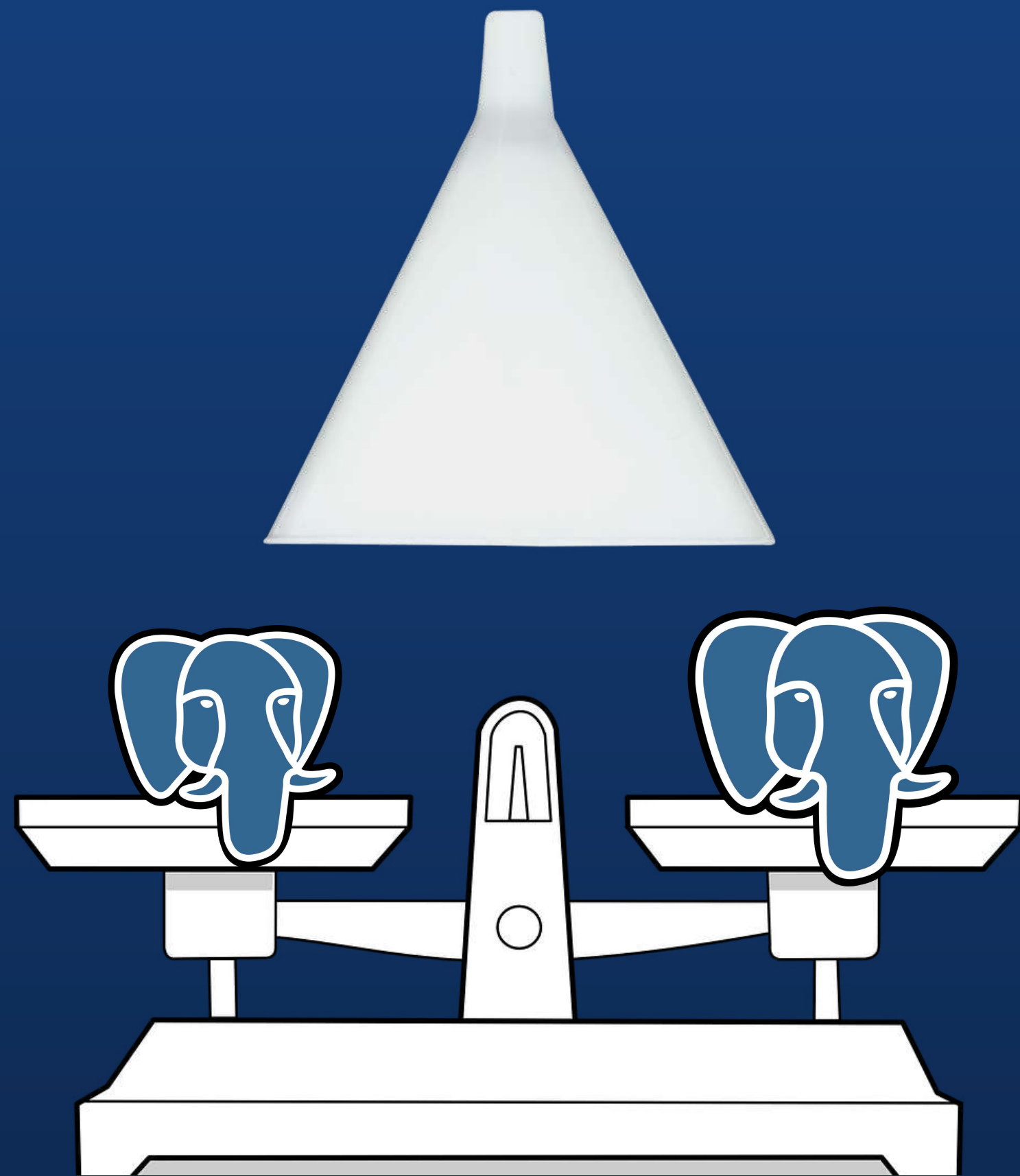
Optimizes Overhead and Resource Utilization

Handles failovers, central switching of Primary

Enables easy use of Read-Replicas

PgBouncer, PgPool-II, pgagroal, PgCat, Odyssey, ...

Resiliency and Overhead



Connection Pooling

Never use PostgreSQL without Connection Pooling!

Optimizes Overhead and Resource Utilization

Handles failovers, central switching of Primary

Enables easy use of Read-Replicas

PgBouncer, PgPool-II, pgagroal, PgCat, Odyssey, ...

<https://tembo.io/blog/postgres-connection-poolers>

Where's my Replicant?

Where's my Replicant?

Use available Kubernetes features

Where's my Replicant?

Use available Kubernetes features

StatefulSet

Where's my Replicant?

Use available Kubernetes features

StatefulSet



Networking and Access Control



Networking and Access Control

Use Network Policies



Networking and Access Control

Use Network Policies

Enable TLS (you remember?!)



Networking and Access Control



Use Network Policies
Enable TLS (you remember?!)
Setup Security Policies

Networking and Access Control



Use Network Policies

Enable TLS (you remember?!)

Setup Security Policies

Configure RBAC (Role-Based Access Control)

Networking and Access Control



Use Network Policies

Enable TLS (you remember?!)

Setup Security Policies

Configure RBAC (Role-Based Access Control)

Think about a policy manager such as OPA or kyverno

Observability and Alerting



Observability and Alerting

Like anything cloud, make sure you have monitoring (meaning observability) and alerting!



Observability and Alerting



Like anything cloud, make sure you have
monitoring (meaning observability) and alerting!

Prometheus Exporter, Log Collector, Aggregation, Analysis, Traceability, ...

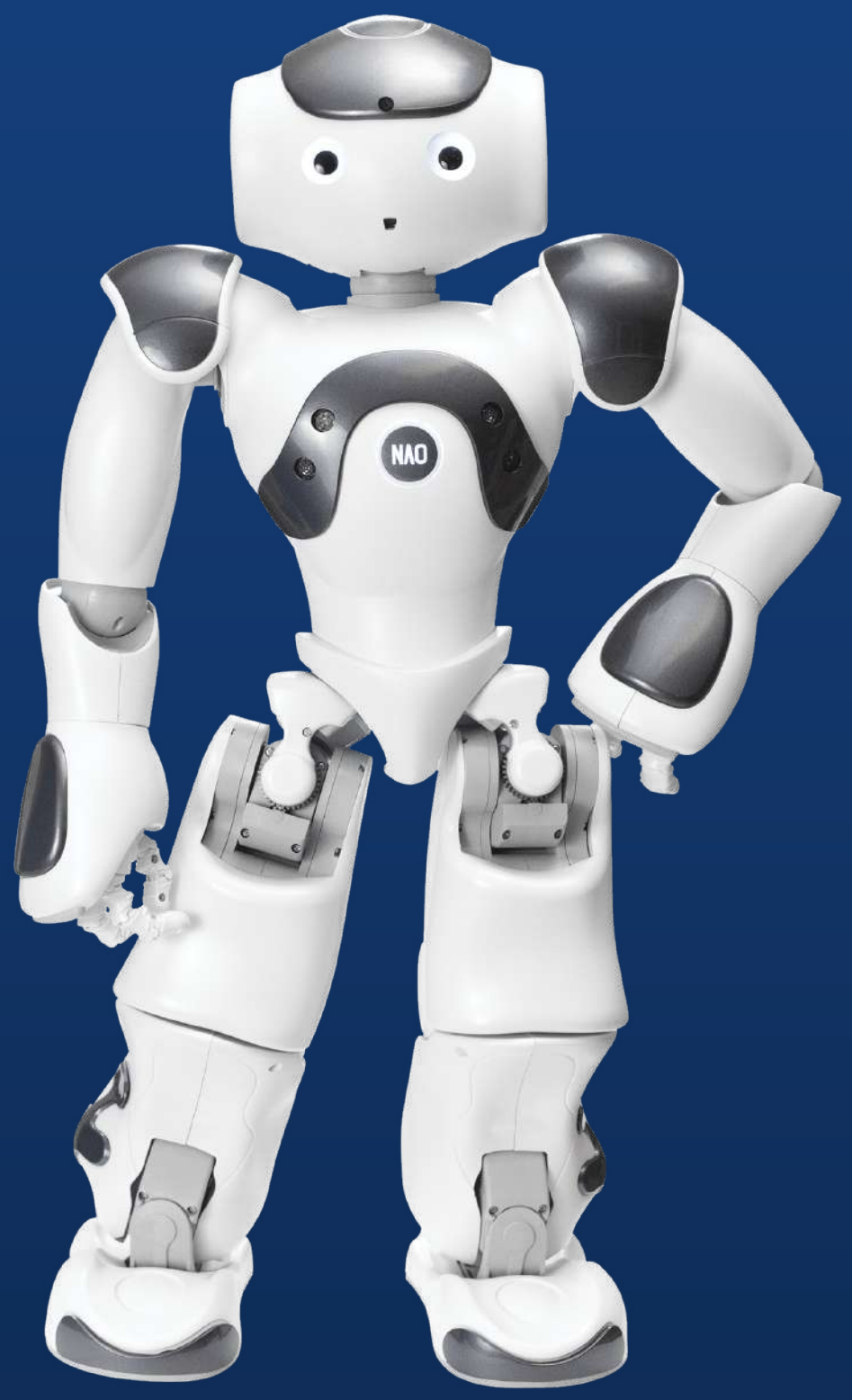
Observability and Alerting



Like anything cloud, make sure you have
monitoring (meaning observability) and alerting!

Prometheus Exporter, Log Collector, Aggregation, Analysis, Traceability, ...

Datadog, Instana, DynaTrace, Grafana, ...





Use a Postgres Kubernetes Operator



Use a Postgres Kubernetes Operator

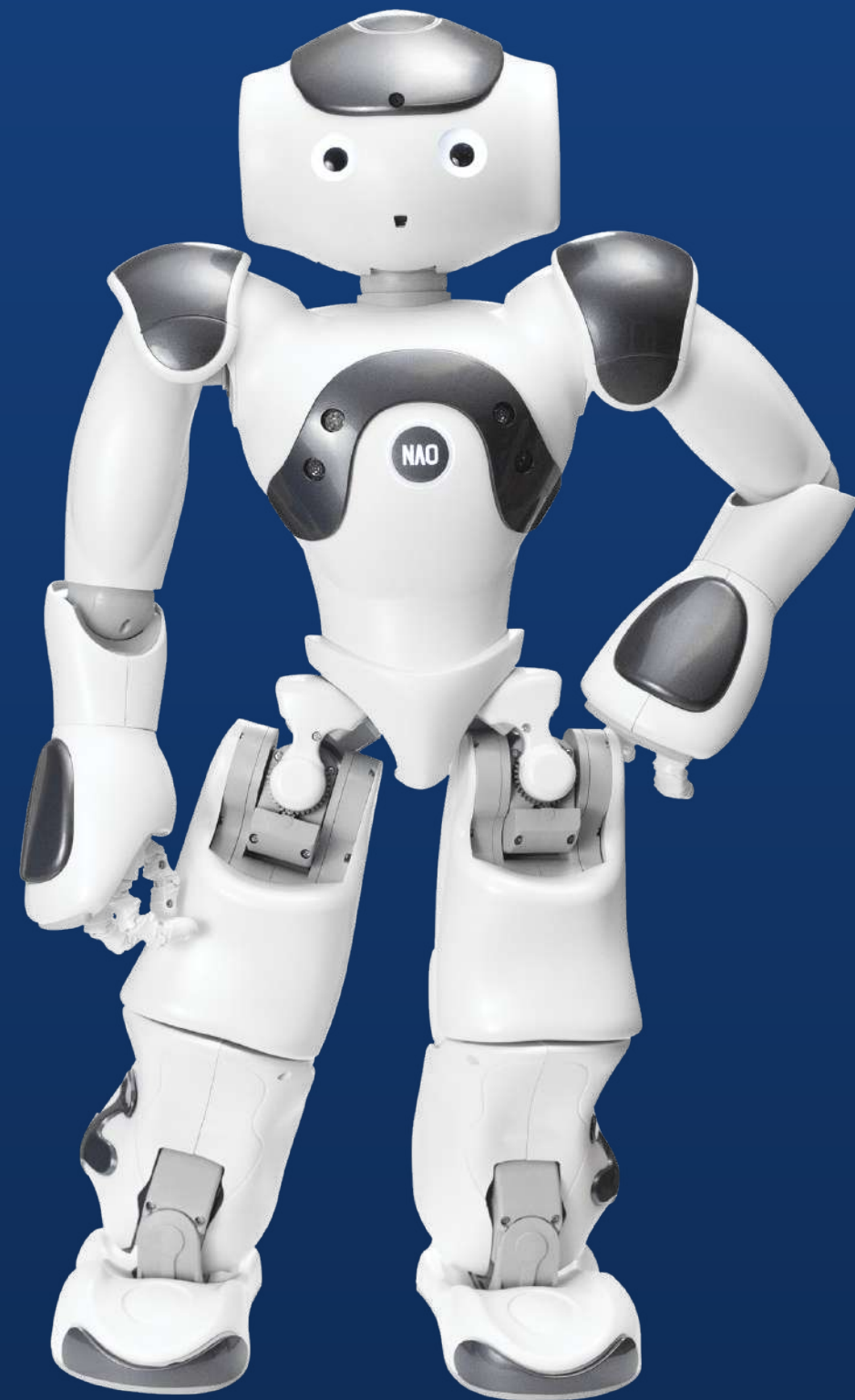
Handles or configures many of the typical tasks (HA, backup, ...)



Use a Postgres Kubernetes Operator

Handles or configures many of the typical tasks (HA, backup, ...)

Brings cloud-nativeness to PG



Use a Postgres Kubernetes Operator

Handles or configures many of the typical tasks (HA, backup, ...)

Brings cloud-nativeness to PG

Integrates PG into k8s



Use a Postgres Kubernetes Operator

Handles or configures many of the typical tasks (HA, backup, ...)

Brings cloud-nativeness to PG

Integrates PG into k8s

If not, use Helm Charts

	CloudNativePG	Crunchy Postgres for Kubernetes	OnGres StackGres	KubeDB	Zalando Postgres Operator
Supported versions	12, 13, 14, 15, 16	11, 12, 13, 14, 15, 16	12, 13, 14, 15, 16	9.6, 10, 11, 12, 13, 14	11, 12, 13, 14, 15, 16
Postgres Clusters	✓	✓	✓	✓	✓
Streaming replication	✓	✓	✓	✓	✓
Supports Extensions	✓	✓	✓	✓	✓

	CloudNativePG	Crunchy Postgres for Kubernetes	OnGres StackGres	KubeDB	Zalando Postgres Operator
Hot Standby	✓	✓	✓	✓	✓
Warm Standby	✓	✓	✓	✓	✓
Automatic Failover	✓	✓	✓	✓	✓
Continuous Archiving	✓	✓	✓	✓	✓
Restore from WAL archive	✓	✓	✓	✓	✓
Supports PITR	✓	✓	✓	✓	✓
Manual backups	✓	✓	✓	✓	✓
Scheduled backups	✓	✓	✓	✓	✓

	CloudNativePG	Crunchy Postgres for Kubernetes	OnGres StackGres	KubeDB	Zalando Postgres Operator
Backups via Kubernetes	✓	✗	✓	✓	✗
Custom resources	✓	✓	✓	✓	✓
Uses default PG images	✗	✓	✓	✗	✗
CLI access	✓	✓	✓	✓	✗
WebUI	✗	✗	✓	✓	✗
Tolerations	✓	✓	✓	✓	✓
Node affinity	✓	✓	✓	✓	✓



<https://www.simplyblock.io/post/choosing-a-postgres-kubernetes-operator>

<https://operatorhub.io/?keyword=postgres>

Pinning and Tainting

Pinning and Tainting

Always use specific, dedicated machines for your database.

Pinning and Tainting

Always use specific, dedicated machines for your database.
(except you're running super small databases)

Pinning and Tainting

Always use specific, dedicated machines for your database.
(except you're running super small databases)

Pin your database containers to those hosts.

Pinning and Tainting

Always use specific, dedicated machines for your database.
(except you're running super small databases)

Pin your database containers to those hosts.

Taint the hosts to prevent anything else from running on it.

Pinning and Tainting

Always use specific, dedicated machines for your database.
(except you're running super small databases)

Pin your database containers to those hosts.

Taint the hosts to prevent anything else from running on it.
(except the minimum necessary Kubernetes services, like KubeProxy)

Trust me, I'm Kelsey!



Kelsey Hightower  
@kelseyhightower

Soham Dasgupta  @thesobercoder · Feb 10, 2023

@kelseyhightower Bust a myth for us please - running any sort of database on a Kubernetes instance is bad idea. I've heard this enough times to actually start believing it. #kubernetes #mythbuster

5:21 PM · Feb 10, 2023 · **326.7K** Views

43 149 839 224 

<https://x.com/kelseyhightower/status/1624081136073994240>

Trust me, I'm Kelsey!



Kelsey Hightower

@kelseyhightower



You can run databases on Kubernetes because it's fundamentally the same as running a database on a VM. The biggest challenge is understanding that rubbing Kubernetes on Postgres won't turn it into Cloud SQL. 🧵



Soham Dasgupta @thesobercoder · Feb 10, 2023

@kelseyhightower Bust a myth for us please - running any sort of database on a Kubernetes instance is bad idea. I've heard this enough times to actually start believing it. #kubernetes #mythbuster

5:21 PM · Feb 10, 2023 · **326.7K** Views



<https://x.com/kelseyhightower/status/1624081136073994240>

Data on Kubernetes Community: <https://dok.community>

[Data on Kubernetes Whitepaper](#)

 @noctarius2k

 @noctarius2k@mastodon.online

 @noctarius.com

Thank you very much!
Questions?

