PlanetScale

Vitess for us all

Enable near infinite performance and scalability while enhancing your database management and reducing infrastructure cost.

PlanetScale.com

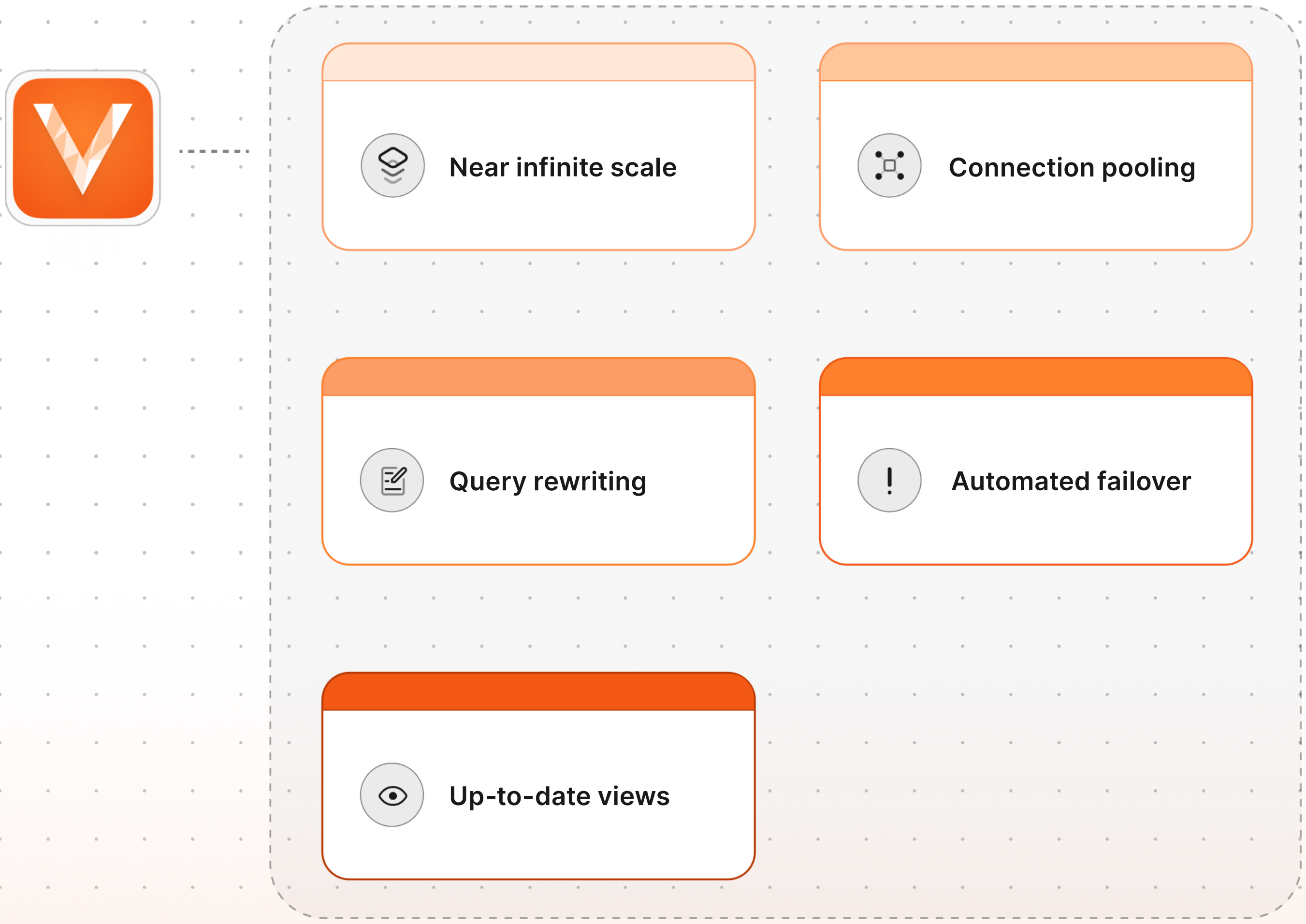


Table of Contents

Vitess for us all

Vitess for us all

Enable near infinite performance and scalability while enhancing your database management and reducing infrastructure cost.

Executive summary

Efficiently managing and scaling databases has become a challenge that many businesses are fighting to overcome. MySQL databases pose a challenge as they aren't pre-packaged with built-in support for sharding. Horizontal scaling via sharding is a scaling solution many businesses will resort to as they experience scaling pains.

The age-old solution of purchasing larger instance sizes – or vertical scaling – is expensive and limited. There are only so many resources that can be purchased on a single server. As more resources are purchased and some dimensions become overprovisioned, costs rise out of proportion to your requirements. Because of this, engineering organizations often resort to horizontal scaling techniques, which distribute the workload across numerous servers as opposed to purchasing more resources on a single server.

Adding more commodity-grade servers is cheaper than purchasing specialized resources for a single server. Horizontal sharding enables the business to scale linearly with a pay-as-you-grow model with more granular control over your costs versus upgrading a single server for a premium and even then, hitting scale limits.

Cost savings point to horizontal sharding as the ideal solution however, implementing horizontal sharding is a difficult feat. Horizontal sharding is typically implemented at the application-level which introduces a mix of complexity and unforeseen costs to your infrastructure and the subsequent management of it.

Vitess, a database management and scaling system developed at YouTube, is the answer to horizontal scaling of MySQL databases with sharding abstracted from your application. Vitess is an open-source solution that abstracts the complexity of sharding-routing logic from the application layer, allowing both application code and database queries to be agnostic to data distribution across multiple shards.

This provides numerous long-term benefits to the business, including quantifiable cost savings and the simplification of scaling operations, permitting the database to evolve to handle more scale as the organization grows and requirements shift. Additional benefits include team productivity and improved application performance, as database administrators are provided the flexibility to re-shard with minimal downtime, and developers can spend time building and improving on products that your customers love as opposed to wrangling the application to properly route queries to the right shards.

In this piece, we explore the architecture and features of Vitess, why companies choose it, and the symbiotic relationship between Vitess and PlanetScale.

The MySQL scaling challenge and Vitess

According to DB-Engines, MySQL is the most popular open-source relational database management system (RDBMS) due to its relative ease of use, the high degree of flexibility that comes with open-source, and its maturity. It has been proven in production for nearly 30 years and has been adopted as the backend of choice by the likes of Yelp, GitHub, Blizzard, and other hyperscalers.

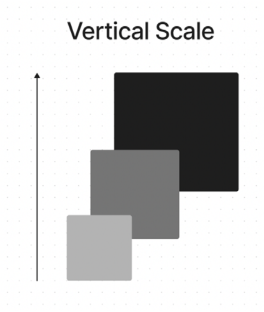
Despite MySQL's many strengths, it does not provide built-in support for sharding. With horizontal sharding and MySQL, companies can get the best of both worlds with a proven and mature database and the scale that comes with horizontal scale.

The following section outlines the various approaches to database scaling that teams often consider.

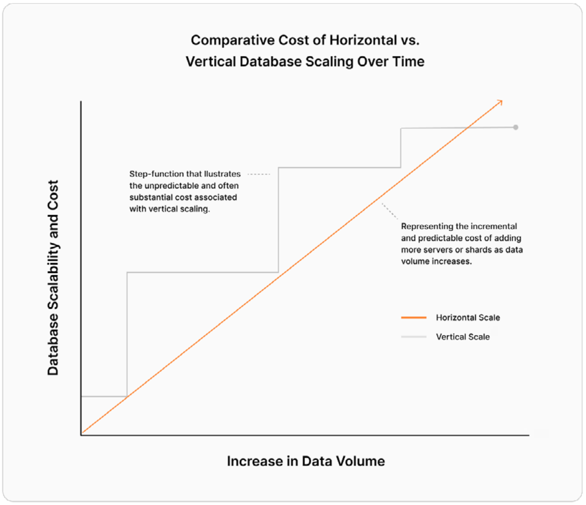
Vertical scale

Increasing the size of your cloud instance or buying bigger machines generally makes more CPU cores, RAM, and other storage space available to you. This can improve the speed and capacity of your MySQL database, enabling it to handle more connections, execute queries faster, and scale up more effectively.

This seems great, but it's the age-old short-term solution: just purchase more resources to make room for scale. This becomes very expensive quickly, and the amount of memory on each server is inherently limited. You will hit more scale issues in the long-term with this solution.



(Figure 1: Vertically scale by increasing the instance size including RAM, CPU, etc.)



(Figure 2: Representing the vertical scale step-function. Leaps in hardware to provision for a spike in workload also leads to a spike in infrastructure cost.)

It’s inefficient to make big leaps in hardware to overprovision for potential spikes in traffic or use. With this method, you will end up paying for resources that you don’t have an immediate need for just to prepare for anticipated spikes in traffic. Once you outgrow your current machine, the next you invest in could easily be 50% larger while you really only end up using 10% of it.

The cost of jumping up instance sizes is high. A typical workload may require the use of a 16xl instance with general purpose storage, costing around $12,251 per month. To prepare for a spike in traffic, it’s plausible to jump up to a larger instance size with Provisioned IOPS to handle a more intensive workload. With a 32xl instance size on Amazon RDS for MySQL at 1 Terabyte (TB) of storage and Provisioned IOPS, the monthly cost would jump up to $24,181 – a 97% increase – and with the nature of vertical scaling, there’s a chance you might not even use all of these resources:

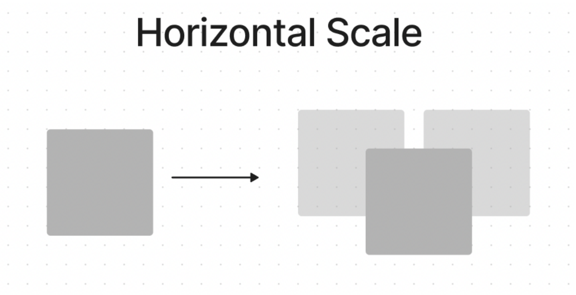
RDS for MySQL

1TB	db.r6i16xlarge	Additional backup storage cost (Monthly): 6.08 USD	db.r6i.32xlarge	RDS MySQL cost (Monthly): 22,425.60 USD
	VCPU: 64	Monthly Cost for RDS Proxy (Monthly): 700.80 USD	VCPU: 128	Monthly Cost for RDS Proxy (Monthly): 1,401.60 USD
	Memory: 512 GiB	RDS MySQL cost (Monthly): 11,212.80 USD	Memory: 1024 GiB	Storage pricing (Monthly): 256.00 USD
	Storage: General Purpose SSD (gp2)	Cost for RDS Performance Insights (Monthly): 96.00 USD	Storage: Provisioned 10PS SSD io	Additional backup storage cost (Monthly): 97.28 USD
		Storage pricing (Monthly): 235.52 USD		
		Total Monthly cost: 12,251.20 USD		**Total Monthly cost: 24,180.48 USD**

(Depiction of the cost of a single db.r6i.16xlarge instance with General Purpose SSD (gp2) storage compared to AWS RDS’s largest instance size available, db.r6i.32xlarge, with Provisioned IOPS SSD io1 storage and (1) 1TB of storage (2) 1 month retention for performance insights and (3) 1TB of backup storage as of May 2023.)

Horizontal scale

When infrastructure costs no longer align with the business requirements due to constant over provisioning, teams often explore horizontal scaling methods where instead of adding more resources on a single instance, you add more instances to handle increasing workloads. This level of granularity enables you to add smaller hosts and invest in your infrastructure more efficiently.



(Figure 1: Horizontal scale by adding more instances and spread out your workload across them)

There are a couple popular options to implement horizontal scale:

1. If you have a read-intensive workload, you can use replicas to scale out,
2. Or you can implement sharding at the application level.

Implementing read-replicas can improve your performance but will likely add lag between your primary and your replicas, which can lead to performance or correctness issues for your application. These complexities can at times require major architectural changes, leading to a suboptimal user experience and difficult compromises, having to choose between application performance or data consistency.

Horizontally sharding at the application level enables you to distribute your workload across numerous instances but this method also has its shortcomings. When you horizontally shard, you have to consider two aspects to your sharding strategy:

1. How to split your database into shards
2. How to route your queries to the correct shard

There's an increasing need to separate sharding logic from the application as it introduces a plethora of complexity, making the application and your database harder to manage which in turn, drains developer capacity and pulls your team away from building and improving on great products for your customer base.

Scaling up Cash App

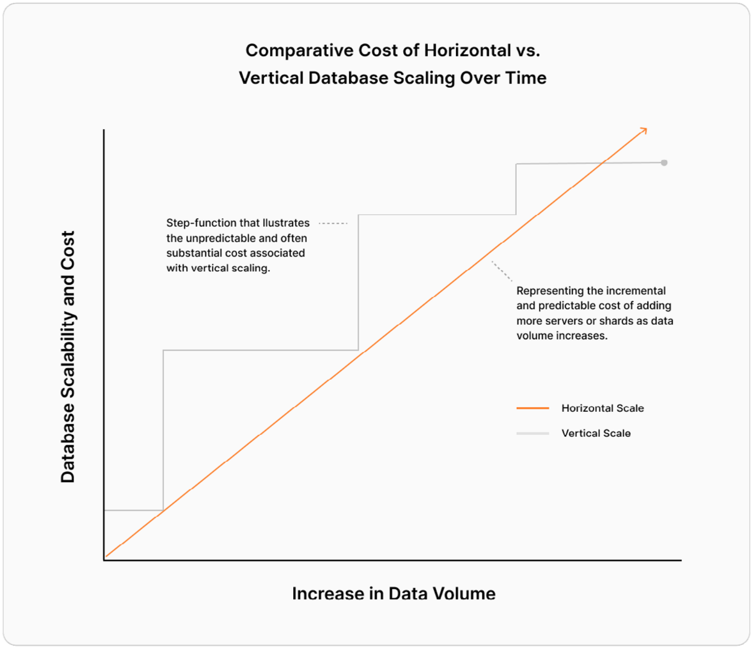
Cash was running through all of the considerations for scaling their monolithic MySQL database - read replicas, buying expensive machines, app-level sharding, and more. These served as great short-term solutions, but were time-consuming to implement and wouldn't enable the near infinite scalability the company needed to scale CashApp, Payments and other features experiencing rapid transactional growth.

Square ultimately decided to split this monolithic database into many smaller databases through horizontal sharding. Aware of the shortcomings of application level sharding, they attempted to find a middleware that solved this. Ultimately, they ended up choosing Vitess.

Vitess is an open-source database management and scaling system developed at YouTube to manage its massive MySQL database requirements. It's designed to serve scale-out applications and has been used to split monolithic or otherwise large MySQL databases into more manageable shards. It is used today at high growth organizations like JD.com, Square, and many more.

Vitess is a middleware that can be placed between your application and your MySQL databases and handles the routing of database queries to the correct shards. It saves you from having to add sharding logic to your application.

There are tangible cost benefits associated with Vitess and horizontal sharding:



(Figure 1: Vertical scale step-function compared to the more linear fashion of horizontal scale.)

Vertical scaling is a step function, providing an immediate boost in resources while horizontal scaling provides more linearity in the resources required to meet database performance requirements and the cost of your infrastructure.

Below is a breakdown of the costs to scale up a commodity sized instance with the same requirements:

8× 125GB	db.r6g.2xlarge VCPU: 8 Memory: 64 GiB Storage: General Purpose SSD (gp2)	Monthly Cost for RDS Proxy (Monthly): 87.60 USD Additional backup storage cost (Monthly): 97.28 USD Cost for RDS Performance Insights (Monthly): 96.00 USD Storage pricing (Monthly): 184.00 USD RDS MySQL cost (Monthly): 12,541.40 USD **Total Monthly cost: 13,006.28 USD**
9 × 112GB	db.r6g.2xlarge VvCPU: 8 Memory: 64 Gig Storage: General Purpose SSD (gp2)	Monthly Cost for RDS Proxy (Monthly): 87.60 USD Additional backup storage cost (Monthly): 97.28 USD Cost for RDS Performance Insights (Monthly): 108.00 USD Storage pricing (Monthly): 207.00 USD RDS MySQL cost (Monthly): 12,641.73 USD **Total Monthly cost: 13,141.61USD**
10 × 100GB	db.rég.2xlarge CPU: 8 Memory: 64 Gi Storage: General Purpose SSD (gp2)	Monthly Cost for RDS Proxy (Monthly): 87.60 USD RDS MySQL cost (Monthly): 12,541.40 USD Storage pricing (Monthly): 230.00 USD Cost for RDS Performance Insights (Monthly): 120.00 USD Additional backup storage cost (Monthly): 97.28 USD **Total Monthly cost: 13,076.28 USD**
11 × 91GB	db.r6g.2xlarge YCPU:8 Memory: 64 Gig Storage: General Purpose SSD (gp2)	Monthly Cost for RDS Proxy (Monthly): 87.60 USD Additional backup storage cost (Monthly): 97.28 USD RDS MySQL cost (Monthly): 13,795.54 USD Cost for RDS Performance Insights (Monthly): 132.00 USD Storage pricing (Monthly): 230.23 USD **Total Monthly cost: 14,342.65 USD**

(Depiction of the cost of many commodity sized servers with General Purpose SSD (gp2) storage on AWS RDS with: (1) 1TB of storage (2) 1 month retention for performance insights and (3) 1TB of backup storage as of May 2023. Not modeled to represent any PlanetScale customer or other realistic workload.)

There is a more linear fashion to scaling using many smaller instance sizes that incrementally align with the requirements of your workload versus over provisioning one very large instance to handle anticipated traffic spikes.

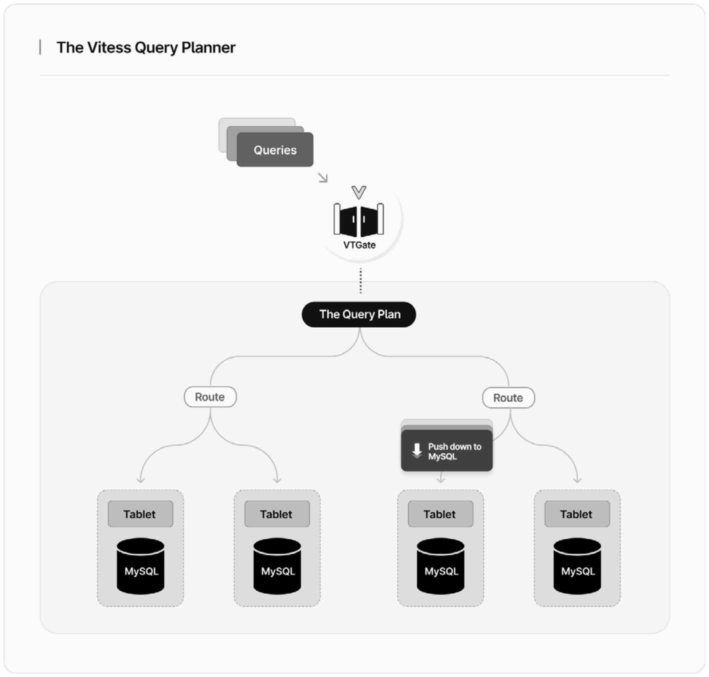
What’s more, Vitess significantly reduces the complexity that comes with scaling database systems, giving your database administrators the power to re-shard with minimal application downtime and your developers their time back so they can focus on what moves the needle for your business: Building and improving on products that your customers love.

In addition to horizontal sharding, Vitess provides a number of capabilities that improve the workflow around the database:

- **Near Infinite Scale:** You will never outscale Vitess. Battle-tested by today’s largest internet companies, Vitess enables you to scale your database to scale out to match your specific requirements and spikes in workload
- **Connection Pooling:** Eliminate the high-memory overhead of MySQL connections by easily handling thousands of connections at once
- **Query Rewriting:** Optimize database queries by dynamically rewriting those that would otherwise impact database performance
- **Automated Failover:** Vitess automatically handles functions like failovers and backups, improving database manageability and enabling your application to be blissfully ignorant of database topology
- **Up-to-Date Views:** Keep track of all metadata related to your cluster configuration to have an always-up-to-date view and consistency

Vitess and the act of sharding the database into smaller and thus, more manageable pieces can drastically improve the manageability of your data. When data is sharded, routine schema changes are much faster. This is because a schema change can be made on individual shards often at the same time versus making a single change to one large dataset, which can be slow, resource-intensive, and is risky as the change affects the entire dataset versus a small portion of it. There’s less fear around introducing a breaking change with schema changes because Vitess offers lossless revert for online schema migrations. With this ability, your team can freely make the necessary schema changes that improve the functionality and performance of your application and easily rollback the table’s schema to the previous state without any data loss if needed.

Access to your data is quicker due to the nature of query distribution across multiple shards, which improves load balancing and reduces the likelihood that your application slows or goes down. With just 40 minutes of downtime resulting in $224,000 on average in revenue loss, businesses can’t afford not to have this level of flexibility with the database. When indexes are introduced, scans can be expedited because scanning much smaller datasets is inherently faster than scanning a monolithic database. Backups and restores are also more reliable, since these operations will be running on smaller databases.



(Figure 1: The use of VTGate, the Vitess feature that routes queries to the appropriate servers and consequently improves load balancing and reduces the likelihood that your application slows or goes down. To learn more about VTGates, checkout the Vitess docs.)

Working with smaller datasets has so many advantages, and Vitess unlocks this ability. Today, you can build and run Vitess on your own or you can run it with the assistance of PlanetScale, the only MySQL-compatible database platform that is built on top of Vitess. With PlanetScale, you not only get all of the power of Vitess, but also benefit from managed backups, query monitoring, caching, and much more.

Why today's largest sites choose Vitess

When infrastructure costs no longer align with the business requirements due to constant over provisioning, teams often explore horizontal scaling methods where instead of adding more resources on a single instance, you add more instances to handle increasing workloads. This level of granularity enables you to add smaller hosts and invest in your infrastructure more efficiently.

Avoiding app-level sharding

Square

Square extended its services to include peer-to-peer transactions with the launch of CashApp which garnered millions of users and transactions in a short period of time. At the time, Square possessed a single large monolith built on top of one single MySQL database. This system wasn't designed to scale due to the inherent limitations around managing a single large instance size. Like many, their team resorted to exploring horizontal sharding techniques and explicitly wanted to avoid application-level sharding. Using Vitess as the middleware between their application and the database seemed like the long-term solution to the scale problems they were experiencing.

The engineering team at Square spent over a year adapting Vitess, rebuilding the shard-splitting workflows and putting in the upfront work to prepare their infrastructure for the change. Implementing Vitess was no easy feat, but Square's first shard split with Vitess caused < 1 second of downtime and continues to support CashApp's ongoing success.

"You have to keep on working on it, but Vitess does provide you essentially with near unlimited scale." - Engineering Manager Jon Tirsen

- 5% of the system had to be changed rather than 95%
- 10 shard splits a week
- Shard splits with < 1 second of downtime

Massive scale and staying on AWS

Slack faced tremendous challenges managing a high volume of MySQL queries per second (QPS) and thousands of MySQL database hosts in production. The burden of handling increasing queries from large customers led to significant strain on the company's busiest database hosts, with a substantial portion remaining idle.

What's more, due to a large and concurrently active user base they faced challenges around managing connections. This arrangement strained the operations team and thus, to alleviate the burden of managing numerous database hosts and to continue using MySQL while meeting growing demand and product needs, they opted for Vitess.

Vitess was appealing because it enabled Slack to continue hosting instances in AWS while meeting the company's scale and MySQL requirements. The impact of Vitess has been overwhelmingly positive in spite of the project presenting more complexities than anticipated. It now supports crucial production application features such as @-mentions, stars, reactions, and more.

> "The project has both been more complicated and harder to do than anybody could have forecast, but at the same time Vitess has performed in its promised role a lot better than people had hoped for."
>
> - Michael Demmer, Principal Engineer

- 500,000 QPS
- 20 billion total queries per day
- Connection latency with Vitess is 1ms on average

Microservices and static headcount

HubSpot has witnessed tremendous growth with more than 64,500 customers across more than 100 countries. This growth translated into the engineering team struggling to manage very large databases, large loads, and significant amounts of data. The company had to tackle the complex task of sharding MySQL manually, an extremely difficult feat that requires extensive expertise. HubSpot sought a solution that wouldn't disrupt business operations or hinder engineering productivity. Vitess emerged as this solution, aligning well with HubSpot's transition to Kubernetes.

Vitess paved the way for a unified data storage and a microservices infrastructure for HubSpot. It's made horizontal scaling of MySQL easy with much shorter turnaround times and the ability to automate away a bunch of daily difficulties data infrastructure engineers encounter. Upgrades that used to take days now only require a couple of hours and Vitess automation has reduced downtime from minutes to seconds during application or other crashes.

"We've been able to almost double our production load and the number of databases [from about 400 MySQL clusters to 700] while keeping the size of the infrastructure team relatively static, between 3 and 5 people." Leo Lin, Technical Lead

- Updates took 2 hours, instead of days
- Downtime lasts seconds instead of minutes
- Doubled production load and # of databases with the same team size

Reduction in hardware and labor costs

JD.com, the world's 3rd largest internet company by revenue and China's largest retailer, was experiencing performance degradation and escalating costs due to the management of multiple very large MySQL databases.

> "We needed a solution that would enable us to easily and quickly scale MySQL, facilitate operation and maintenance, and reduce hardware and labor costs."
>
> - JD Retail Chief Architect Haifeng Liu

- 300+ million active customers
- Tens of thousands of MySQL containers
- Millions of tables, trillions of records

To overcome these challenges, the company transitioned to running MySQL databases in containerized environments managed by Kubernetes and began using Vitess for scalable cluster management, enabling the handling of large volumes of complex transactional data.

Since then, JD has reaped the benefits of linear scalability by increasing their resource utilization and provisioning for only the resources needed leading to greater efficiency and a notable reduction in labor and resource costs.

Hitting limits on GCP and implementing Vitess

Etsy was facing a crucial need to scale its Payments databases at the end of 2020. Two of the company's databases had reached the maximum resource tier on the Google Cloud Platform (GCP), making vertical scaling infeasible. These databases were integral to process payments on a daily basis and were at a high risk of performance issues and potential transaction losses due to traffic spikes. Etsy's Payments Platform, Database Reliability Engineering, and Data Access Platform teams transitioned 23 tables with over 40 billion rows from four unsharded databases to a single sharded environment managed by Vitess.

This transition was a two stage project:

1. Migrate the seller ledger infrastructure that calculates seller bills and payouts
2. Reduce the load on Etsy's primary database - a 10+ year old system hosting 90 tables and encompassing transaction, payment, and other data.

Although Etsy successfully pulled off these two stages, they encountered challenges across the stack that made implementing Vitess more difficult than anticipated. First, establishing the ideal data model that would inform their sharding strategy required deep expertise to carefully modify constraints like unique keys, indexes, and more on an already resource-constrained database. Once this was complete, reducing the load on Etsy's primary database was an even more complex process that again required extensive expertise to create new data models that accounted for even more data growth, new features, and tight deadlines. After overcoming these and many more challenges, Etsy made the cutover and since then, has used Vitess to efficiently manage and scale their Payments database.

Reducing maintenance costs with scale

Twitter was facing the challenge of meeting strict service level objectives on queries per second (QPS), latency, success rates, and cross data center consistency as they scaled up. Seeking a cost-effective way to support Twitter's main application features and desiring to stick with MySQL due to its already wide use within the organization, the team decided to explore adding replicas for redundancy. Despite this being a potential solution, it wouldn't enable Twitter to scale reads up to millions of QPS.

Twitter then decided to employ Vitess to not only shard and scale up writes, but also to leverage Vtgates to scale up reads. This choice was motivated because of the open-source nature of Vitess and its seamless integration with MySQL. Twitter was able to include a topology service to store configuration data with Vitess, utilizing its highly available Zookeeper clusters for this service and integrating it with Orchestrator for cluster maintenance. Twitter now has Vitess running in production with extremely high availability for both reads and writes.

- Achieved millions of queries per second
- Scaled up to a couple thousand connections
- Stuck to strict encryption-in-transit requirements

These are a few of the stories of the many organizations that have chosen Vitess as their long-term solution to gain control over and even decrease infrastructure costs, increase internal team productivity, and meet scale requirements.

Vitess is trusted by

affirm	airbnb	AXON	Bloomberg	Bolt	box
bumble	Community	CloudSigma	Etsy	Flipkart	BetterCloud
GitHub	HubSpot	JD.COM	myfitnesspal	NAVER	New Relic.
nozzle	peak GAMES	pinkoi	Pinterest	QuizofKings	Recurly
SharpSpring	shopify	slack	Square	stitchlabs	Trustwave
tumblr.		Vinted	weave	yelp	YouTube

(The above is a list of Vitess users. Please note that while certain companies included on the list are indeed customers of PlanetScale, the list should not be construed as a comprehensive or complete list of PlanetScale's customer base.)

Challenge of managing Vitess

As recounted through the stories of real Vitess users above, introducing a middleware and distributed system like Vitess is an excellent solution to scale-related and database management issues but it also requires a time investment to build and upkeep. Designing sharding and schema is complex. Even though Vitess is the ideal solution with a conceptually simple implementation, migrating to it is no small feat.

This is why the co-creator of Vitess - Sugu Sougoumarane - went on to co-found PlanetScale, making Vitess accessible to everyone.

Vitess under the hood: symbiotic relationship between PlanetScale and Vitess

PlanetScale is the only MySQL-compatible database platform that is built on top of Vitess. With PlanetScale, every database you spin up, whether it's free or paid, gets Vitess under the hood. Because of this, PlanetScale democratizes many Vitess features and capabilities, including horizontal sharding, online schema migrations, and more. With PlanetScale, you can unlock all of the power of Vitess in a much shorter period of time and without all of the required expertise, risk, and potential errors that come with running it yourself.

Behind the impressive features of PlanetScale is a team of Vitess maintainers that contribute to the open-source project and ensure that PlanetScale remains at the cutting-edge of database innovation. PlanetScale's expertise in Vitess and contributions to the project have solidified their position as a trusted partner for organizations seeking a managed Vitess solution. The relationship between Vitess and PlanetScale is symbiotic, with PlanetScale actively collaborating with the Vitess community to enhance the system's functionality and to roll out Vitess updates and new features in a timely manner.

The easiest way to deploy Vitess

PlanetScale's contributions to Vitess extend beyond maintenance and support. Vitess is an extraordinary technology and PlanetScale is the easiest way to deploy it, bringing additional value-added features and services that provide all of the cost-benefits, ease of database management, and the plethora of other benefits that accompany it:

> "We wanted PlanetScale and Vitess to bring to MyFitnessPal what Kubernetes brought to application delivery and deployment. Databases are hard. We would rather PlanetScale manage them."
>
> - Chris Karper, Engineering Director at MyFitnessPal

✓ **Ease of database management**

With built-in features like branching, non-blocking schema changes, and schema reverts, managing the database is simple. With a truly managed database service that enables your engineering team to move more effectively, and support tiers that make sense for your business, PlanetScale makes database administration easy.

✓ **Long-term MySQL scaling strategy**

With Vitess under the hood, PlanetScale offers horizontal scaling via sharding with minimal application changes. Because of this, you're able to keep sharding logic out of the application layer, providing you with a long-term scaling strategy for MySQL that enables near infinite scale and that is easier to manage.

✓ **Linear scalability and sensible cost structure**

By the nature of horizontal scale, PlanetScale enables you to scale out transparently and with resources that optimally serve your requests. This enables you to keep infrastructure costs close to your business needs. With PlanetScale, it's easier to prove a solid return on investment on infrastructure costs.

✓ **Safe migrations and sensible developer workflows**

With a git-like branching workflow that's paired with safe migrations, you can roll out schema changes to your production database with ease of mind knowing your team can review and approve of changes using deploy requests and then easily roll the change back if a breaking change is introduced.

✓ **Connection pooling**

PlanetScale can support nearly infinite connections utilizing Vitess's built-in connection pooling and unique edge infrastructure that ensures connection limits are never an issue. This leads to an enhanced user experience and improved application performance.

✓ **Query monitoring and analytics**

PlanetScale Insights is an in-platform query performance analytics tool that allows you to view 100% of queries versus a randomly sampled subset. This enables your team to streamline and improve the workflow around database query optimization and debugging.

✓ **Caching with PlanetScale Boost**

PlanetScale Boost is a groundbreaking technology that is not available on open-source Vitess that enables you to significantly improve the performance of queries at the click of a button and without having to build tedious cache invalidation logic or maintain caching infrastructure. Useful for many of the same use cases as an application cache backed by Redis or memcached, but native to the PlanetScale platform.

✓ **Data security and compliance**

PlanetScale provides encryption at every level, audit logs, SSL connection, and more to ensure your data is safe. With PlanetScale Managed, you can deploy the same scalable MySQL database platform but run in an AWS or GCP account owned by you. If you have a sensitive security posture that requires HIPAA compliance, PlanetScale can sign BAAs. PlanetScale reports and monitors using SOC Type II.

Deploy Vitess without the hassle, expertise, or operational burdens today with PlanetScale. Contact PlanetScale to learn more.

Get started today with PlanetScale, the most reliable way to scale your MySQL database in the cloud.

Call us at 1-408-214-1997
or send an email to sales@planetscale.com.