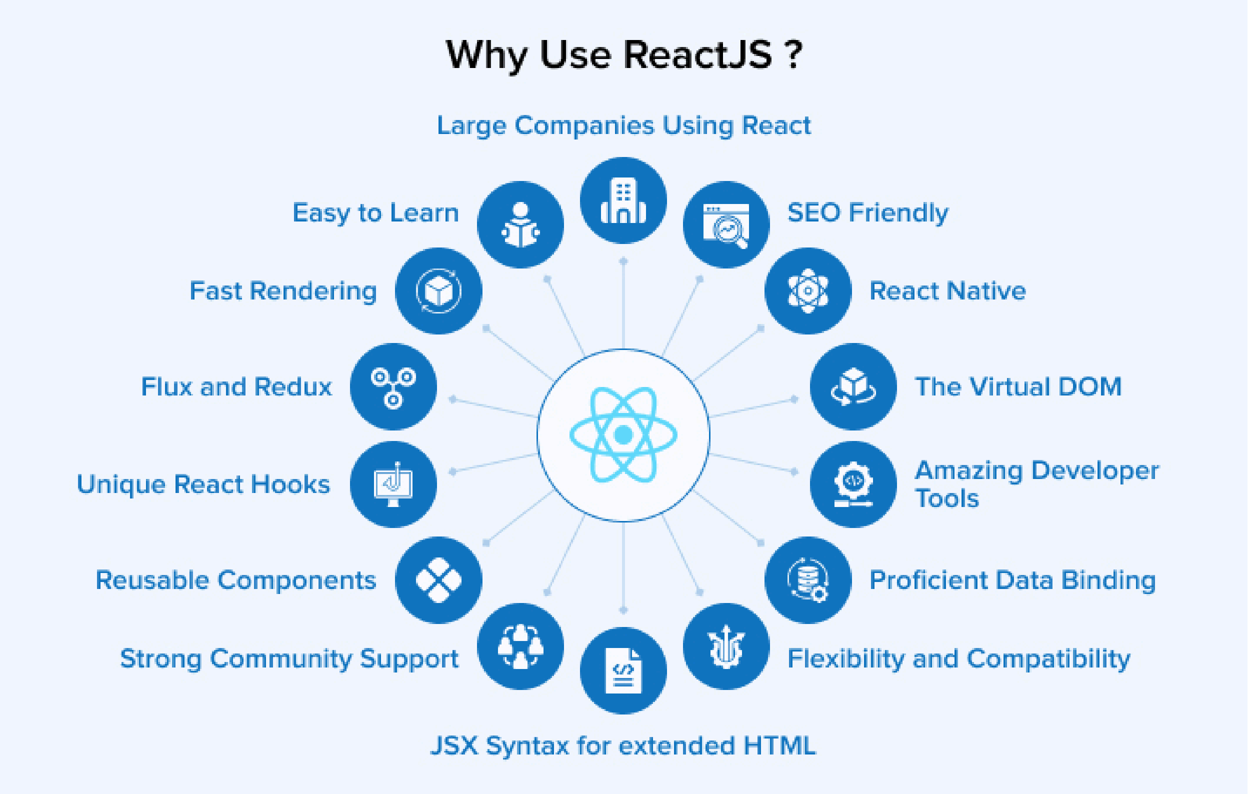
Frontend: React.js, HTML5, CSS3, JavaScript



Because ReactJs is one of the most widely used front-end JavaScript libraries for creating Web Applications. Around 8,000 businesses around the world have chosen React over other popular libraries and frameworks for building rich user interfaces. React is a Facebook-developed JavaScript package that was used to create Instagram, among other things. It is like a panacea for developers to build user interfaces for websites and apps swiftly. The virtual DOM is the core notion of ReactJs.

React has grown in popularity as a front-end web development tool among react development companies since it only requires a basic familiarity of HTML and JavaScript. In March, 2015, Facebook released React Native, an open source UI framework, to develop applications for android, iOS, mac, and other various OS. We'll discover what ReactJs is and why React is better than other JavaScript frameworks in this Article.

What is ReactJS?
The best fit for developing user interfaces is React, as it is a very declarative, fast, and flexible JavaScript library. It allows you to create complex user interfaces using "components," or small, self-contained pieces of code. It controls the view layer in web applications. Despite the fact that React is more of a library than a language, it is frequently used in web development.

React is a novel technology when compared to previous available options. The JavaScript library was established in 2011 by Jordan Walke, a Facebook software engineer of the time. Frameworks like XHP, a simple HTML component framework for PHP, have influenced React. React was used in Facebook's newsfeed in 2011 at the beginning. It was later picked up by Instagram and used in their system. React evolved over time, and in May 2013, at JSConf US, Facebook chose to open source it.

ReactJS provides a smooth solution to some of front-end programming's most vexing issues, making it simple to create dynamic, interactive web programmes. A strong development community exists that is quick, scalable, versatile, powerful, and rapidly growing. There's never been a better time to learn React.

Nowadays, it is a challenge to find a platform that is simple to use, provides an optimized development interface, has clear abstractions and also renders virtual DOMs comparatively faster than other platforms. Not just this, but ReactJS, as a development platform, is capable of offering much more than usual. Hence, it is important to understand why you should consider using ReactJS for your next development project.

Why Use ReactJS?

The developers have complete control over React JS. It can also be used to create SPA, and web apps. You can include any number of external libraries and tools to create large and complex web applications.

Many JavaScript developers have swiftly adopted React, and it appears to be here to stay. ReactJS has a rich ecosystem that includes must-have tools like Redux or flux, as well as Node.js on the backend, all of which are available on NPM. Developers can construct states in the DOM using the scalable React architecture pattern, which makes it easy to break down and test React components.

For good reason, React's popularity and usage are growing by the day. As a developer, coding in React improves your knowledge of JavaScript, a language that accounts for nearly 90% of web development today. Its functionality in effortlessly managing different frameworks gives it a considerable edge for those who desire some code flexibility.

The front-end development ecosystem is constantly evolving. New tools with so many frameworks and libraries are releasing day by day, making it hard for businessmen to rely on correct choices. React JS is second to none in web development all the way along. React provides cost effective, high performance, reusable wrapper components, and many other benefits for web application development.

Here are eight key reasons to use ReactJs for your next project; I guarantee you'll discover more once you start using it.

1. **Easy to Learn:** Unlike Angular and Vue, learning React is like five-finger exercise, well of course on your keyboard! No wonder why, that's a focal reason for React's rapid adoption. It allows businesses to complete projects more quickly. React is more credible to be used by large firms as it is such a framework that is easy to get going with.

2. **Reusable Components:** ReactJS provides developers with reusable components that they can use to create new applications. Reusability is like a developer's miracle cure. This platform enables developers to reuse any react component created for another application that performs the same function. As a result, development effort is reduced while flawless performance is maintained.

   One of the chief benefits of using React JS is that developers can reuse components as many times as they need to. Furthermore, changes made in one section of the application have no effect on other sections of the application. If you thought React was only for web development, you were dead wrong! Facebook has already developed the React Native framework for creating mobile apps for both the Android and iOS platforms.

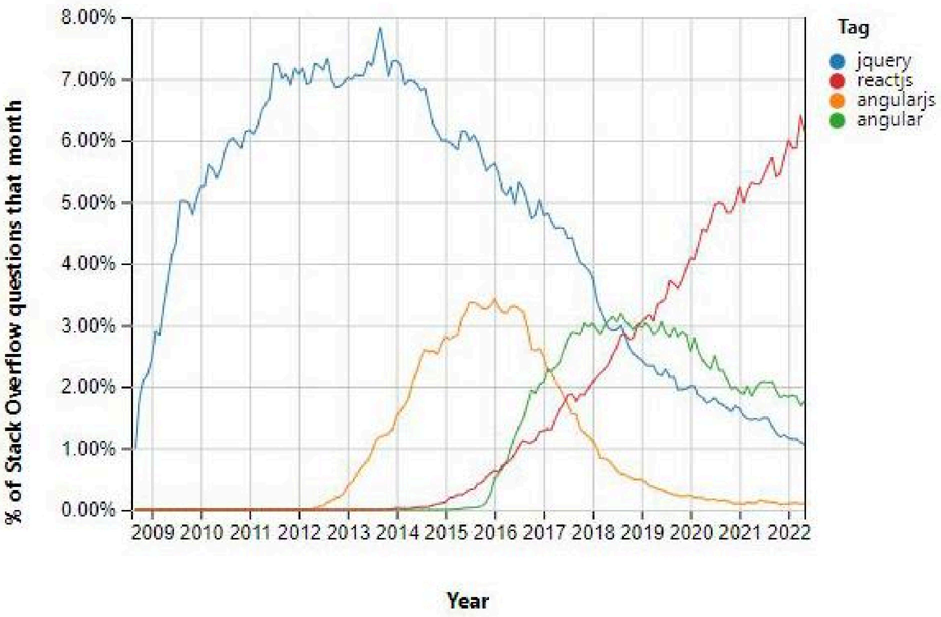
3. **Fast Rendering:** When developing a complex, high-load app, it is essential to define the structure of the app at the beginning as it can greatly influence the app's performance. Simply put, As the DOM is tree-structured, even little changes at the root can cause major impacts on others. To solve this problem, Facebook has brought up a virtual DOM feature. Virtual DOM, which allows all modifications to be tested first for calculating risks. This approach contributes to app performance and a better user experience.

4. **SEO Friendly:** As you know, Success of every online business relies on how well your search engine is optimised. Another benefit of React js is that it can cope with a common search engine failure to comprehend JavaScript-heavy apps. React significantly reduces load time of websites, which helps businesses to lead on the top spot on Google's Search Engine Result Page.

   There are some of the frameworks that React possesses which improves SEO , one of them is NextJs. The rendering part of NextJs is moved to the server, removing the need for the client to process the data. Since the server pre-renders the pages before sending the final HTML to the client, there is less JS to load, which improves performance and SEO.

5. **Strong Community Support:** One of the main reasons to choose React for your project is a strong community behind it. A massive range of individual React developers contribute to creating a more robust frontend framework on a daily basis. React JS currently has around 189k stars on GitHub and around 38.9k active contributors. You can find a vast collection of React tutorials on Youtube and explanatory articles on the internet from professionals and experts. For example, a Google search for "Free React tutorial" brings around 5 crore results. Besides this, React professionals solve doubts on Stack Overflow and Quora, so whenever you feel stuck while working on React , you know where to find solutions.

Stack Overflow Trends



As shown in the above graph, there are clear results that use of React is continuously increasing, as the questions related to React are increasing. This graph shows the Stack Overflow Trends.

6. **Proficient Data Binding:** ReactJS lags behind one-way data binding. So that anyone can track all the changes made on particular portions of the data. This is often an illustration of its simplicity.

7. **The Virtual DOM:** The virtual DOM(Document Object Model) is one of the most intriguing aspects of React. While creating an app with a lot of user interaction and data changes, you should think about how the structure of the app will affect speed. Extensive DOM manipulation can cause congestion in performance. Worse, As the DOM is tree-structured, even little changes at the root can cause major impacts on others.

   A virtual DOM is used by React to solve this problem. This is a virtual version of the DOM, as the name implies. Any new view updates are made on the virtual DOM first, which is stored in memory rather than on your screen. It then finds the most efficient method of making these modifications, and only those changes are applied to the real DOM. This ensures that the real DOM is updated as quickly as possible, resulting in improved performance and a cleaner user experience overall.

8. **Flux and Redux:** React offerings include out-of-the-box Flux and Redux capacities which makes it an ideal choice for web development. The introduction of Flux-based software architecture provided the optimal structure of actions for standard React components and improved it with unidirectional data flow capabilities.

The orchestration of created actions and store updates is done using a central dispatcher. And as per the store changes, it updates the views. This keeps all your data in the store without generating any duplicates. All the model data remains well-synced throughout the app without going far.

The Flux is just an architectural pattern to create a convenient UI design workflow. But you can't use it as a fully-fledged library. For a convenient implementation of Flux, you need to use Redux. It provides a single store object that can manage all of your app data. This makes data management manipulations pretty straightforward. The changes in the store trigger Renders and the view syncs with the related data.

9. **JSX Syntax for extended HTML:** Using ReactJS allows you to use declarative HTML syntax directly in your JS code. Web browsers decode HTML texts to display user interfaces. Creating DOM trees and then manipulating them with JavaScript helps build interactive UI.

   The use of JSX makes the DOM manipulation more efficient. Adding HTML and ReactJS components in the tree structure of the browser may help make your code tidy and maintainable. The Virtual DOM and JSX enable the ReactJS apps to be fast and more efficient.

10. **Unique React Hooks:** Hooks allow JavaScript programmers to add states and other functionalities in function components. No more coping with difficult classes. Transferring data between components that don't have classes or props, handling group comparable logic in a single component, and managing state logic between components becomes easy with React Hooks.
11. **Amazing Developer Tools:** React comes with a large array of tools that can help the developer improve efficiency, productivity, and performance. When using Redux, its Redux developer tools help keep track of the state. Meanwhile, React developer tools, browser extensions, and JSX syntax are used to inspect the app's component architecture. Another reason why many developers love to use React is because its ecosystem is filled with tons of open-source libraries.

12. **Flexibility and Compatibility:** ReactJS for web development is convenient because you can use it across various platforms. The main goal of this natural library is to create separate web design components and elements.

    Moreover, with the significant contribution of React's long-established community, its ecosystem has become vast enough to help developers build mobile apps, desktop solutions, and static websites, manage server rendering as well as integrate advanced technological concepts with web solutions using simple React web development principles.

13. **React Native:** If you have learned programming with React then you can easily create native applications for iOS and Android platforms using React Native. Although you can't use the web code, the architecture, and the methodology remain the same.

14. **Large Companies Using React:** Examples of some large companies using ReactJS include Facebook, Netflix, and Codecademy.

- **Facebook:** One of the earliest adopters of React was Facebook. It is also the birthplace of React. So, it would be obvious that the company would be using it. Apart from Facebook, its subsidiaries such as Instagram and WhatsApp are also using ReactJS for their web client apps and React Native for mobile apps.
- **Netflix:** Netflix opted for React to integrate extensive testing in their components and offer the best performance to the users. Their engineering team claimed that among many advantages, React helped them boost the speed, runtime performance, and modularity of their streaming platform.
- **Codecademy:** In 2014, Codecademy rebuilt its platform using React. They shared that React's rich experience in JSX, its one-way data binding, and its ability to handle DOM manipulation were some of the reasons behind choosing the framework.

Backend: Django(python)

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Here are several detailed reasons why Django is a preferred choice for backend development:

1. Batteries-Included Framework

- **Built-in Features:** Django comes with a wide array of built-in features that facilitate common web development tasks. These include an ORM (Object-Relational Mapping), authentication system, form handling, URL routing, template engine, and more. This reduces the need for third-party dependencies and accelerates development.
- **Admin Interface:** Django's admin interface is automatically generated from your models and provides a powerful tool for managing application data. It's highly customizable and helps developers quickly set up administrative tasks without additional coding.

2. Scalability

- **Modular Architecture:** Django's modular architecture allows developers to scale applications effectively. Components can be developed, tested, and deployed independently, facilitating horizontal scaling.
- **Efficient Codebase:** The use of Django's ORM enables efficient database operations and optimization. Combined with Python's inherent strengths, Django can handle high traffic loads and large volumes of data efficiently.

3. Security

- **Built-in Protections:** Django includes several built-in security features to protect against common vulnerabilities such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and clickjacking. This ensures a robust security baseline for applications.
- **Authentication System:** Django's authentication system supports various methods such as usernames and passwords, third-party authentication, and single sign-on (SSO). This makes it easier to implement secure user authentication and authorization.

4. ORM (Object-Relational Mapping)

- **Database Abstraction:** Django's ORM provides a powerful and intuitive way to interact with databases using Python code instead of SQL. This abstraction simplifies database operations and reduces the likelihood of errors.
- **Database Flexibility:** Django supports multiple databases (e.g., PostgreSQL, MySQL, SQLite, Oracle) and allows for seamless database migrations and schema changes using Django's migration system.

5. Community and Ecosystem

- **Extensive Documentation:** Django has comprehensive and well-organized documentation, making it easier for developers to learn and troubleshoot.
- **Community Support:** A large and active community contributes to Django's continuous improvement and provides a wealth of third-party packages and reusable apps. This ecosystem supports rapid development and a wide array of use cases.
- **Django Packages:** The Django Packages website lists many reusable apps and tools that extend Django's functionality, ranging from e-commerce solutions to content management systems (CMS).

6. Rapid Development

- **Convention Over Configuration:** Django emphasizes a clean and pragmatic design by following the "convention over configuration" principle. This reduces boilerplate code and speeds up the development process.
- **Code Reusability:** Django's design encourages code reuse through components such as reusable apps and templates. This enhances productivity and maintainability.

7. RESTful APIs and GraphQL

- **Django REST Framework (DRF):** DRF is a powerful toolkit for building Web APIs. It integrates seamlessly with Django, providing features such as serialization, authentication, and viewsets that simplify API development.
- **GraphQL Support:** Django can be integrated with Graphene, a Python library for building GraphQL APIs, enabling efficient querying and manipulation of data.

8. Versatility and Use Cases

- **Versatility:** Django is used across various domains, including content management, e-commerce, social networking, scientific computing, and more. Its versatility makes it suitable for both small-scale projects and large, complex applications.
- **Proven Track Record:** Many high-profile websites and applications, such as Instagram, Pinterest, and Disqus, use Django, demonstrating its capability to handle large-scale and mission-critical applications.

9. Testing and Quality Assurance

- **Built-in Testing Tools:** Django includes a robust testing framework that supports unit tests, integration tests, and system tests. This promotes a test-driven development (TDD) approach, ensuring higher code quality and reliability.
- **Continuous Integration:** Django can be easily integrated with continuous integration (CI) tools to automate testing and deployment processes, facilitating smoother and more reliable development cycles.

10. **Large Companies Using React**
    jango is the choice of many, and it’s also the choice of the Internet’s giants. Here are some of the popular web applications enjoy Django’s speed, flexibility, and ease of use:
    - **Instagram:** Another popular social network that deals with a great deal of media data and user interactions. Django enables the functionality that makes the web application work seamlessly, add new features, and fix issues in no time.
    - **Spotify:** A large media library with huge volume of information that allows users to listen to music free of charge or on an ad-free subscription basis. On the technical side of things, Spotify also uses machine learning, where Python is one of the best choices. The creators chose to combine it with the Django framework.
    - **The Washington Post:** It’s no wonder that The Washington Post would use Django to handle its heavy traffic, since the framework itself was created to power an online newspaper. Several other online periodicals also use Django.
    - **Dropbox:** A cloud technology for file storage requires high-performance functionality. Django provides Dropbox with the tools it needs to provide sharing and synchronization, as well as scalability.
    - **Mozilla:** One of the most popular web browsers, and another web application that switched from PHP to Python, and thus to Django. Now, it is better able to handle huge traffic and API hits.
    - **NASA:** They use Django for some of the functional elements that require the most reliability. Their website is not as popular as Instagram, for example, yet it handles high-resolution imagery and an average of 2 million views per month.
    - **Pinterest:** This platform has well over 250 million active users per month, handles huge volumes of media data, and yet manages to sustain its user-friendly interface. Thanks to its open-source framework, the creators were able to modify it to suit their needs.
    - **Reddit:** What’s the first site go to for social news and discussions? For many people, it’s Reddit — a web content platform where people ask questions, share opinions, and where posts are promoted based on a rating system. Reddit relies on Django for many of its functional capabilities, like servicing heavy user traffic.

Database: MongoDB

MongoDB is a document database built on a horizontal scale-out architecture that uses a flexible schema for storing data. Founded in 2007, MongoDB has a worldwide following in the developer community.

Instead of storing data in tables of rows or columns like SQL databases, each record in a MongoDB database is a document described in BSON, a binary representation of the data. Applications can then retrieve this information in a JSON format.

Here’s a simple JSON document describing a historical figure.

```
{
        "_id": 1,
        "name": {
                "first": "Ada",
                "last": "Lovelace"
        },
        "title": "The First Programmer",
        "interests": ["mathematics", "programming"]
}
```

Document databases are highly flexible, allowing variations in the structure of documents and storing documents that are partially complete. One document can have others embedded in it. Fields in a document play the role of columns in a SQL database, and like columns, they can be indexed to increase search performance.

From its founding, MongoDB was built on a scale-out architecture, a structure that allows many small machines to work together to create fast systems and handle huge amounts of data.

MongoDB has always focused on providing developers with an excellent user experience, which, in addition to all its other properties, has made MongoDB a favorite of developers worldwide for a wide variety of applications.

Why Use MongoDB?

MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications with evolving data schemas.

As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a JSON-like format to store documents. This format directly maps to native objects in most modern programming languages, making it a natural choice for developers, as they don’t need to think about normalizing data. MongoDB can also handle high volume and can scale both vertically or horizontally to accommodate large data loads.

MongoDB was built for people building internet and business applications who need to evolve quickly and scale elegantly. Companies and development teams of all sizes use MongoDB for a wide variety of reasons.

Document Model

The document data model is a powerful way to store and retrieve data in any modern programming language, allowing developers to move quickly.

Deployment Options

MongoDB is available in any major public cloud (such as AWS, Azure, and Google Cloud) through MongoDB Atlas, in large data centers through the Enterprise Advanced edition, or free through the source-available Community edition.

Get Started Quickly

MongoDB has a great user experience for developers who can install MongoDB and start writing code immediately.

Fully Scalable

MongoDB's horizontal, scale-out architecture can support huge volumes of both data and traffic.

Find Community

MongoDB has developed a large and mature platform ecosystem. It has a worldwide community of developers and consultants, making it easy to get help. It also has an enterprise-grade support offering.

Using MongoDB enables your team to go further and faster when developing software applications that handle data of all sorts in a scalable way.

MongoDB is an excellent choice if you need to: Support rapid iterative development. Enable collaboration of a large number of teams. Scale to high levels of read and write traffic. Scale your data repository to a massive size. Evolve the type of deployment as the business changes. Store, manage, and search data with text, geospatial, or time-series dimensions.

MongoDB as a company has grown because the number of use cases with these characteristics continues to grow.

Features of MongoDB

1. **Scale Cheaper**

   You can scale cheaper with MongoDB. Why?

Let's begin by talking about scaling SQL databases. Typically, SQL databases scale vertically-when a database becomes too big for its server, it is migrated to a larger server.

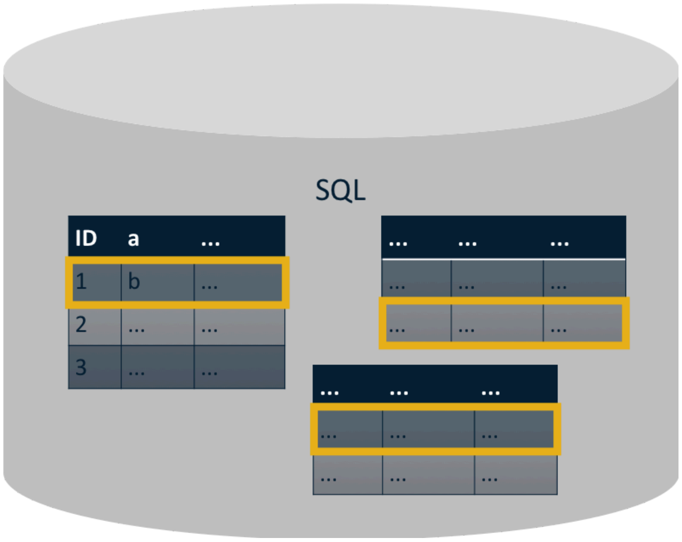
A few key problems arise with vertical scaling:

- Large servers tend to be more expensive than two smaller servers with the same total capacity.
- Large servers may not be available due to cost limitations, cloud provider limitations, or technology limitations (a server the size you need may not exist).
- Migrating to a larger server may require application downtime.

When you use MongoDB, you have the flexibility to scale horizontally through sharding
. Sharding is a method for distributing data across multiple servers. When your database exceeds the capacity of its current server, you can begin sharding and split it over two servers. As your database continues to grow, you can continue to add more servers. The advantage is that these new servers don't need to be big, expensive machines-they can be cheaper, commodity hardware. Plus, no downtime is required.

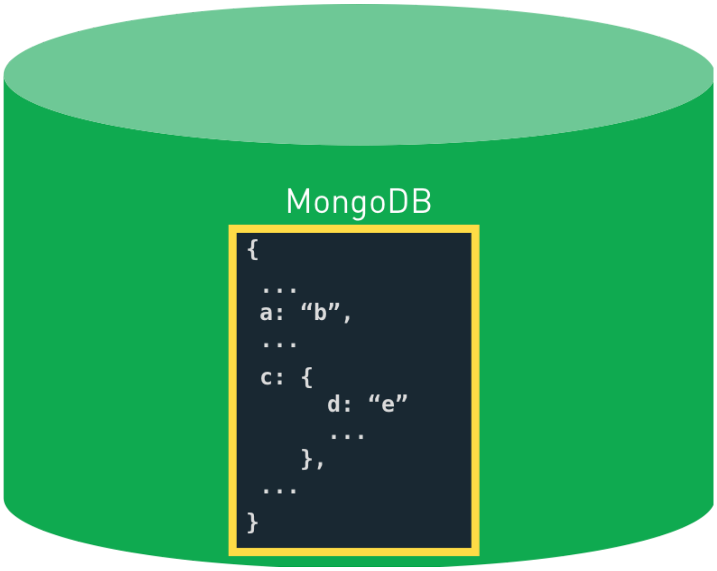
2. Query Faster

Your queries will typically be faster with MongoDB. Let's examine why.
Even in our simple example in the previous post
where we modeled Leslie's data in SQL, we saw that her information was spread across three tables. Whenever we want to query for Leslie's information, we'll need to join three tables together.
In these three small tables, the join will be very fast. However, as the tables grow and our queries become more complex, joining tables together becomes very expensive.



Recall our rule of thumb when modeling data in MongoDB: data that is accessed together should be stored together. When you follow this rule of thumb, most queries will not require you to join any data together.
Continuing with our earlier example, if we want to retrieve Leslie's information from MongoDB, we can simply query for a single document in the Users collection. As a result, our query will be very fast.

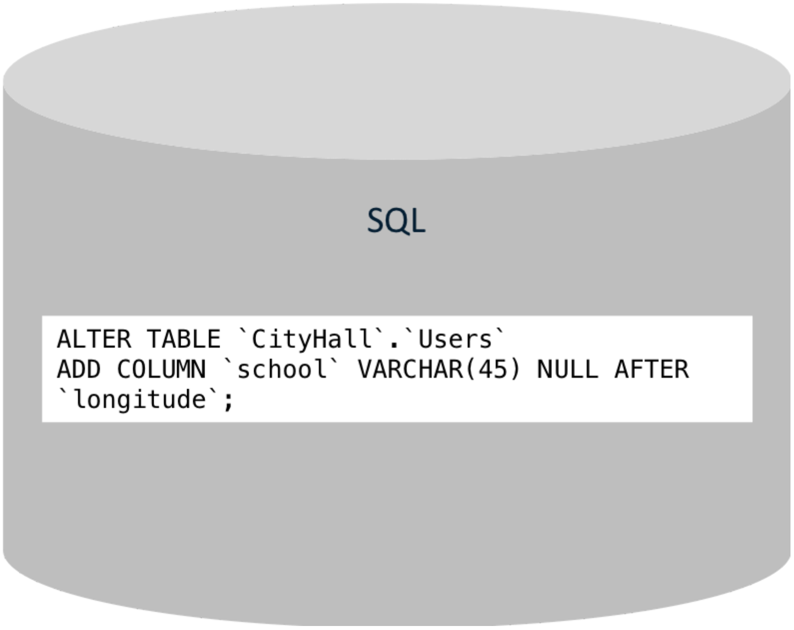
As our documents and collections grow larger, we don't have to worry about our queries slowing down as long as we are using indexes and continue following our rule of thumb: data that is accessed together should be stored together.



```
{
 ...
 a: “b”,
 ...
 c: {
       d: “e”
       ...
    },
 ...
}
```

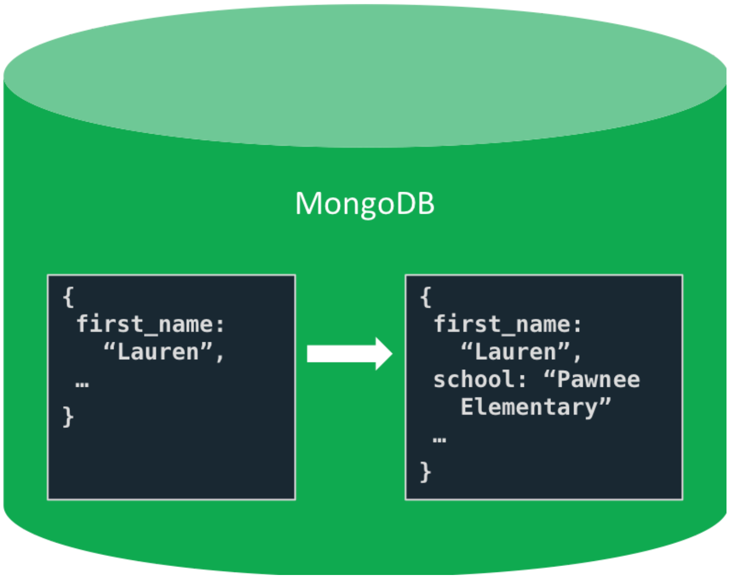
3. **Pivot Easier**

Requirements change. Sometimes the changes are simple and require only a few tweaks to the user interface. But sometimes changes go all the way down to the database.
In the previous post in this series, we discovered—after implementing our application—that we needed to
store information about Lauren's school
. Let's take a look at this example a little more closely.
To add a new school column in our SQL database, we're going to have to alter the Users table. Executing the Alter Table command could take a couple of hours depending on how much data is in the table. The performance of our application could be decreased while the table is being altered, and we may need to schedule downtime for our application.



```
ALTER TABLE `CityHall`.`Users`
ADD COLUMN `school` VARCHAR(45) NULL AFTER
`longitude`;
```

Now let's examine how we can do something similar in MongoDB. When our requirements change and we need to begin storing the name of a user's school in a User document, we can simply begin doing so. We can choose if and when to update existing documents in the collection. If we had implemented schema validation, we would have the option of applying the validation to all inserts and updates or only to inserts and updates to documents that already meet the schema requirements. We would also have the choice of throwing an error or a warning if a validation rule is violated.
With MongoDB, you can easily change the shape of your data as your app evolves.



```
{
 first_name:
   “Lauren”,
 …
}
```

```
{
 first_name:
   “Lauren”,
 school: “Pawnee
   Elementary”
 …
}
```

4. **Program Faster**
   MongoDB documents map to data structures in most popular programming languages. This sounds like such a simple thing, but it makes a humongous difference when you're writing code. A friend encouraged me to test this out, so I did. I implemented the code to retrieve and update user profile information. My code has some simplifications in it to enable me to focus on the interactions with the database rather than the user interface. I also limited the user profile information to just contact information and hobbies.

Advantages of MongoDB
The Power of Document-Oriented Databases

MongoDB is the pioneer of what has come to be called NoSQL databases, which developed because RDBMS systems based on SQL did not support the scale or rapid development cycles needed for creating modern applications.

NoSQL is an umbrella term; it includes document-oriented databases like MongoDB, columnar databases, in-memory databases, and more.

In MongoDB, records are stored as documents in compressed BSON files. The documents can be retrieved directly in JSON format, which has many benefits:

It is a natural form to store data.

It is human-readable.
Structured and unstructured information can be stored in the same document.
You can nest JSON to store complex data objects.
JSON has a flexible and dynamic schema, so adding fields or leaving a field out is not a problem.
Documents map to objects in most popular programming languages.
Most developers find it easy to work with JSON because it is a simple and powerful way to describe and store data.

Perhaps most importantly, the developer controls the database schema. Developers adjust and reformat the database schema as the application evolves without the help of a database administrator. When needed, MongoDB can coordinate and control changes to the structure of documents using schema validation.

MongoDB created Binary JSON format (BSON) to support more data types than JSON. This new format allows for faster parsing of the data. Data stored in BSON can be searched and indexed, tremendously increasing performance. MongoDB supports a wide variety of indexing methods, including text, decimal, geospatial, and partial.

Developer User Experience

MongoDB has always devoted abundant time and energy to making sure developers have a great experience. Developers appreciate that MongoDB has made sure the database can be used from various programming languages, including C, C# and .NET, C++, Go, Java, JavaScript, PHP, Python, Ruby, Rust, Scala, and Swift.

As more and more business users have joined the MongoDB community, features have been added to support the use and operation of MongoDB in enterprise IT departments. MongoDB now also offers first-class support for customers who need it.

With MongoDB Atlas, the database-as-a-service at the center of the MongoDB Cloud, it is easier than ever to use MongoDB. You can provision a cluster with a few clicks from the web interface and start writing code almost immediately.

MongoDB Atlas allows developers to get started right away in any major public cloud and easily migrate on-premise MongoDB instances to the cloud.

MongoDB Atlas also embeds powerful capabilities like:

MongoDB Atlas Search (powered by the Lucene search engine) to enable full-text search.
Atlas App Services, fully-managed back-end services for building mobile and web apps.
MongoDB Atlas Data Lake, which allows developers to query and combine data stored in Atlas with data stored in Amazon S3 or an HTTPS store.

Scalability and Transactionality

MongoDB's scale-out architecture, which distributes work across many smaller (and cheaper) computers, means that you can create an application that will handle spikes in traffic as your business grows.

Engineering innovations by MongoDB support massive numbers of reads and writes. MongoDB's approach to sharding is at the heart of these innovations, allowing clusters of information to be stored together as the information is spread across the cluster of computers. By comparison, most SQL databases use a scale-up architecture that is limited because it relies on creating faster and more powerful computers.

When modeling data in MongoDB, it is common to embed objects within each other. What used to take multiple transactions to update in traditional relational databases can sometimes be achieved in a single transaction with MongoDB.

If still needed, MongoDB also supports database transactions that allow many changes to a database to be grouped and either made or rejected in a batch.

Platform and Ecosystem Maturity

MongoDB has been around since 2007 and has been deployed at thousands of companies for a wide range of use cases. A natural result of that usage level is that the platform has been extended to meet a massive number of new demands. Most large organizations want to make sure it is easy to get help using any technology that becomes the foundation of their business.

MongoDB has a large and thriving community of developers across the open-source community, academia, and among system integrators and consulting firms across the globe.

Authentication: JWT, OTP via third-party services (e.g., Twilio)

JWT (JSON Web Tokens)

Why Use JWT?

- **Stateless Authentication:** JWT enables stateless authentication, meaning the server does not need to store session information. This reduces server load and improves scalability, making it ideal for large-scale applications and microservices architectures.
- **Compact and Efficient:** JWTs are compact and can be easily passed through URLs, POST parameters, or inside HTTP headers. This efficiency is especially useful in mobile applications and single-page applications (SPAs) where performance is critical.
- **Security:** JWTs are signed using a secret (with the HMAC algorithm) or a public/private key pair (using RSA or ECDSA). This ensures the integrity and authenticity of the token. Additionally, JWTs can be encrypted to protect sensitive information.
- **Flexibility:** JWTs can carry any type of information (claims), allowing them to be used for various purposes such as user authentication, information exchange, and access control. Custom claims can be added to meet specific application requirements.
- **Decentralized Verification:** JWTs allow decentralized verification of tokens. Any service that possesses the public key or the shared secret can verify the token without relying on a central authentication server, improving performance and reliability in distributed systems.
- **Standardization:** JWT is an open standard (RFC 7519) that is widely adopted and supported by many programming languages and frameworks. This ensures interoperability and simplifies integration with different systems.

Use Cases for JWT

- **Single Sign-On (SSO):** JWTs are commonly used in SSO systems to securely transmit user identity information between multiple services.
- **API Authentication:** JWTs are ideal for authenticating users in RESTful APIs, where each request can include a token to validate the user.
- **Session Management:** JWTs can manage user sessions in web and mobile applications, providing a seamless and secure user experience.

OTP (One-Time Password) via Third-Party Services (e.g., Twilio)

Why Use OTP via Third-Party Services?

- **Enhanced Security:** OTP adds an extra layer of security by requiring a user to provide a one-time password in addition to their regular password. This two-factor authentication (2FA) method significantly reduces the risk of unauthorized access, even if a user's password is compromised.

- **User Convenience:** Third-party services like Twilio handle the complexities of generating and sending OTPs via SMS, email, or voice calls, ensuring reliable and timely delivery. This improves the user experience by providing a seamless and efficient way to receive OTPs.
- **Global Reach:** Services like Twilio have extensive global reach and infrastructure, allowing OTPs to be sent to users worldwide quickly and reliably. This is essential for applications with a global user base.
- **Scalability:** Third-party services are designed to handle high volumes of OTP requests, ensuring scalability as your user base grows. They provide robust infrastructure and load balancing to manage peak loads and ensure high availability.
- **Compliance and Security:** Reputable third-party providers comply with industry standards and regulations, ensuring that the OTP delivery process meets security and privacy requirements. This reduces the burden on developers to implement and maintain secure OTP systems.
- **Ease of Integration:** Services like Twilio offer easy-to-use APIs and SDKs, simplifying the integration of OTP functionality into your application. This reduces development time and effort, allowing developers to focus on core application features.

Use Cases for OTP

- **User Authentication:** OTP can be used for two-factor authentication (2FA) during user login to enhance security.
- **Transaction Verification:** OTP is commonly used to verify sensitive transactions, such as financial transfers, to ensure the identity of the user initiating the transaction.
- **Password Recovery:** OTP can be sent to users for password recovery or account verification processes, providing a secure way to reset passwords or verify user identity.

Hosting: Google Cloud or AWS or similar cloud services

Why use cloud hosting?

Cloud hosting allows website and applications operators to add or remove resources when necessary. That includes more RAM, storage space, or support services such as security or data storage. Cloud hosting provides reliability and flexibility at a manageable cost. Cloud hosting also provides robust data backup and disaster recovery compared to shared or dedicated hosting on a single server.

Advantages of Choosing Google Cloud Hosting

1. Outstanding Availability and Uptime

A fundamental factor in the success of your website is its availability. You can have the best content or product in the marketplace, but none of that will matter if visitors can't access your website.

The last thing you need is losing visitors and revenue because of website downtime. This is the time when prospective followers and customers will most likely seek out your competitors.

Regular periods of website downtime will not only impact your bottom line but can also negatively affect your search engine rankings. Google and Bing spiders regularly visit your website to crawl your site, check content, page speed, and availability. Frequent or lengthy outages will impact your SEO (Search Engine Optimization).

Your hosting provider has a direct influence on your website uptime. There's very little you can do to increase uptime on your own. Beyond avoiding errors in your code, you're entirely reliant on your hosting provider.

Achieving 99.99% Uptime
A massive advantage in choosing Google Cloud hosting solutions is their commitment to achieving uptime upwards of 99.99%. This is reflected in their Compute Engine Service Level Agreement (SLA), where any dip below a monthly uptime of 99.99% will result in financial credit.

Here at Kinsta, moving our infrastructure to Google Cloud Platform (GCP) allows us to sit on the shoulders of giants. We gain access to one of the largest networks in the world, which is resistant to downtime through countless failover scenarios for system and server outages. We then directly pass these benefits to our customers, offering premium hosting solutions with monthly uptime approaching 100%.

Other leading cloud providers, including Amazon Web Services (AWS) and Microsoft Azure, have infrastructure rivaling Google Cloud's uptime availability. However, after significant research, there doesn't appear to be a definitive or reputable study crowning the most reliable cloud platform (yet).

2. Live Migration of Virtual Machines

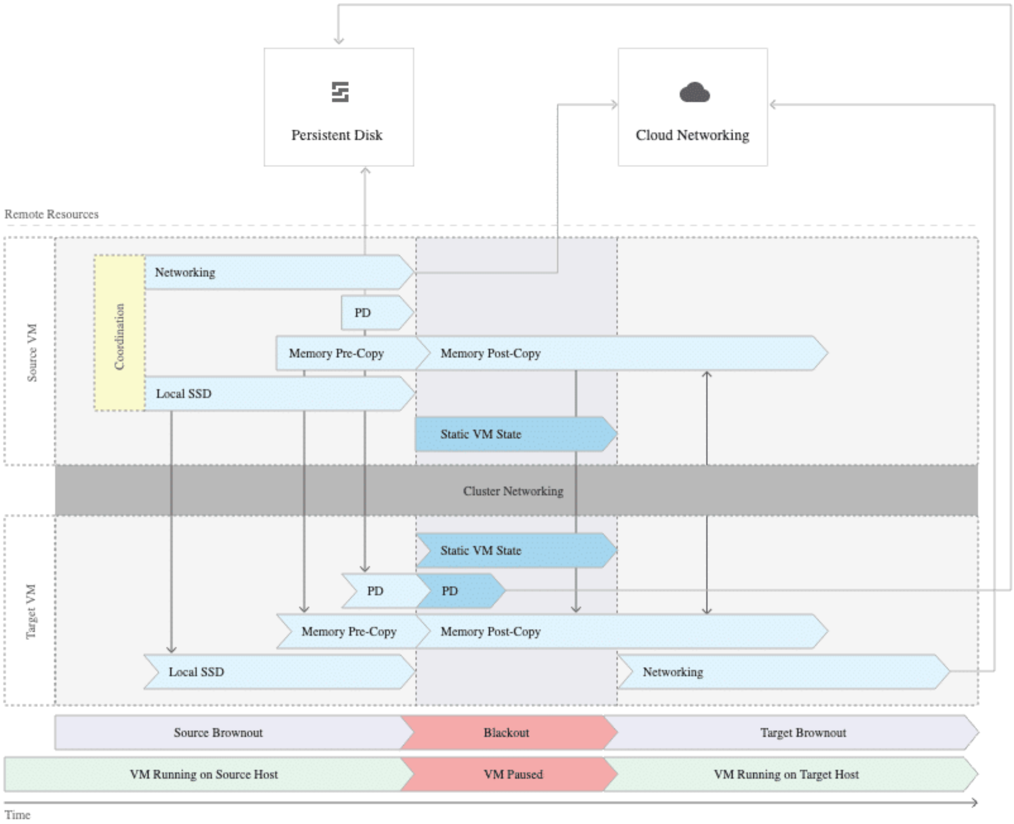
Another huge advantage of Google Cloud hosting, especially here at Kinsta, is the live migration of virtual machines (VMs). This allows us as a hosting provider to maintain the availability of your website — with no noticeable degradation in performance — when live-migrating VMs between host machines.

Patching, repairing, and updating software and hardware is a must in delivering a cutting-edge service. Live migration with Google keeps your instances and website available during:

- Regular infrastructure and maintenance upgrades
- Network and power grid maintenance in the data centers
- Hardware failure including CPU, memory, disks, network interface, power, and more (though this is not always possible; in instances where hardware prevents live migration or fails, the VM will crash and restart automatically)

- Host operating system (OS) and BIOS upgrades
- High importance security-related updates
- System configuration changes

This involves many components and is a highly complex process. To better appreciate the moving parts involved, check out this illustration of the high-level steps:



Although initially a unique offering from Google Cloud, competitors have begun to see the benefits: Live migration is a service that Microsoft Azure has subsequently replicated.

3. Free Uptime Monitoring

Considering the potential impacts of downtime, monitoring your infrastructure and website availability is a vital practice.

Implementing a solution to perform uptime checks lets you automatically track the availability of your hosting setup and be alerted in the event of any outages. This allows you rapidly respond to any issues inside your control and minimize downtime while also tracking whether your provider meets their SLA.

Another advantage of Google cloud hosting is the operations suite, which allows you to monitor the performance of your platform and applications. Creating uptime checks and alerting policies is a simple process. It’s also completely free to monitor your Google infrastructure (though monitoring technology outside Google Cloud will cost extra).

The platform lets you create uptime checks across the following resource types:

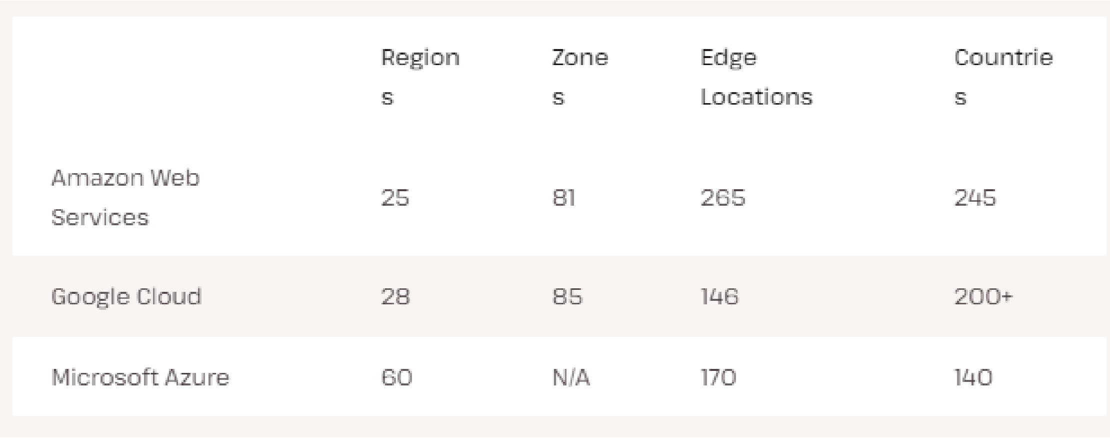
- URL
- Kubernetes LoadBalancer Service
- VM instance
- App Engine service
- AWS load balancer

4. Leading Global Infrastructure

One of the biggest things setting Google Cloud hosting apart from other players is their global network. It’s easily one of the biggest in the world, rivaling the global infrastructure of both Microsoft and Amazon.

Comparing the global infrastructure of the top three cloud providers is no easy task. They each use different terminology to describe their global network, and certain data points aren’t publicly available.

Although we can distinguish no clear winner, you can see from the data we’ve gathered in the table below that Google Cloud offers a leading global network:



	Regions	Zones	Edge Locations	Countries
Amazon Web Services	25	81	265	245
Google Cloud	28	85	146	200+
Microsoft Azure	60	N/A	170	140

The existence of a global network will play a pivotal role in your user experience. Geographical proximity plays a key role in the speed at which your users can access your website and resources.

A larger and more geographically distributed network lets you offer local access to more people. This means more of your end users will enjoy a faster and better user experience when visiting your website.

5. Unrivaled Throughput

Throughput is the rate at which data processes and transfers between two locations. In networking, it is used to measure the performance — meaning speed — of VMs and hard drives along with internet and network connections.

Network throughput measures how many packets of data successfully arrive at their destination in a set timeframe. Packet loss creates slow and poor-performing networks. This, in turn, will create slow speed in your applications and impact your users.

In short, your website will take longer to load, especially in peak traffic situations.

A major advantage of choosing Google Cloud hosting is the platform's unrivaled network throughput. A recent comparison of the top 3 cloud providers showed Google Cloud VMs have nearly 3x the network throughput of their AWS and Azure counterparts. Google's bottom-performing network throughput machine performed 65% and 105% better than AWS' and Azure's top-performing machines.

The winning VM type for network throughput performance was Google's Compute-Optimized VMs (C3). Here at Kinsta, we've made these new GCP machines available on all WordPress plans.

It's these machines that have helped make Kinsta clients like Hardbacon and Enventys Partners aware of decreases in page load time — 63% and 50%, respectively — while also making performance dips due to traffic spikes a thing of the past.

6. Performance Optimization With Network Service Tiers

A key benefit of Google cloud hosting is the opportunity to optimize your network for performance or price through a choice of Network Service Tiers.

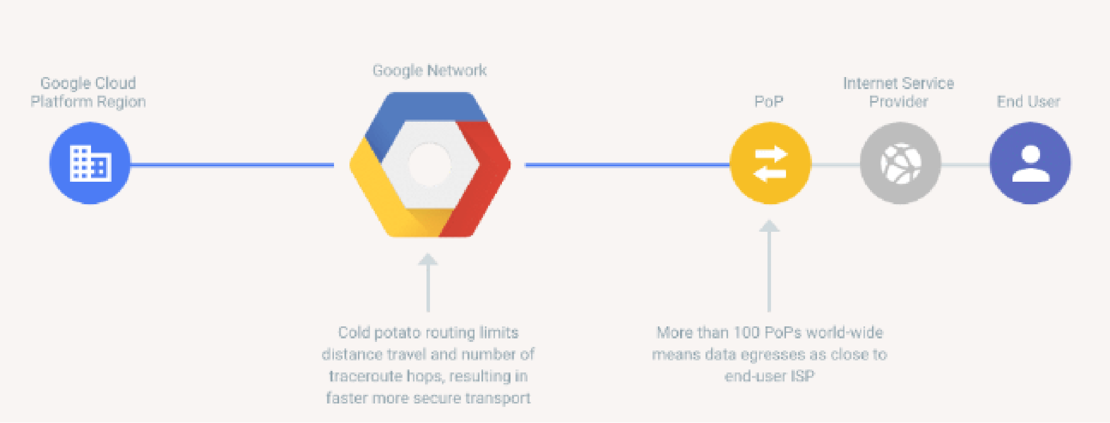
Having created a massive network of real-world infrastructure, the Google Cloud Platform can support the vast majority of the journey from the user to your resources. Customers gain access to their cutting-edge network, bypassing congested public internet routing and minimizing the number of traceroute hops for faster performance.

Let's look at how the Google Cloud Platform Network Service Tiers break down.

Premium Tier

The Premium tier, the more expensive of the two options, is designed for high-performance routing. Traffic is conducted through Google's global private fiber network with more than 100 points of presence (POPs).

The Premium tier is built for speed. The user is picked up from the internet service provider (ISP) at the nearest PoP. They then travel through the Google Network using cold-potato routing, an approach that limits the distance and the number of traceroute hops for faster and more secure transport.


Google Cloud Platform Region
Google Network
PoP
Internet Service Provider
End User
Cold potato routing limits distance travel and number of traceroute hops, resulting in faster more secure transport
More than 100 PoPs world-wide means data egresses as close to end-user ISP

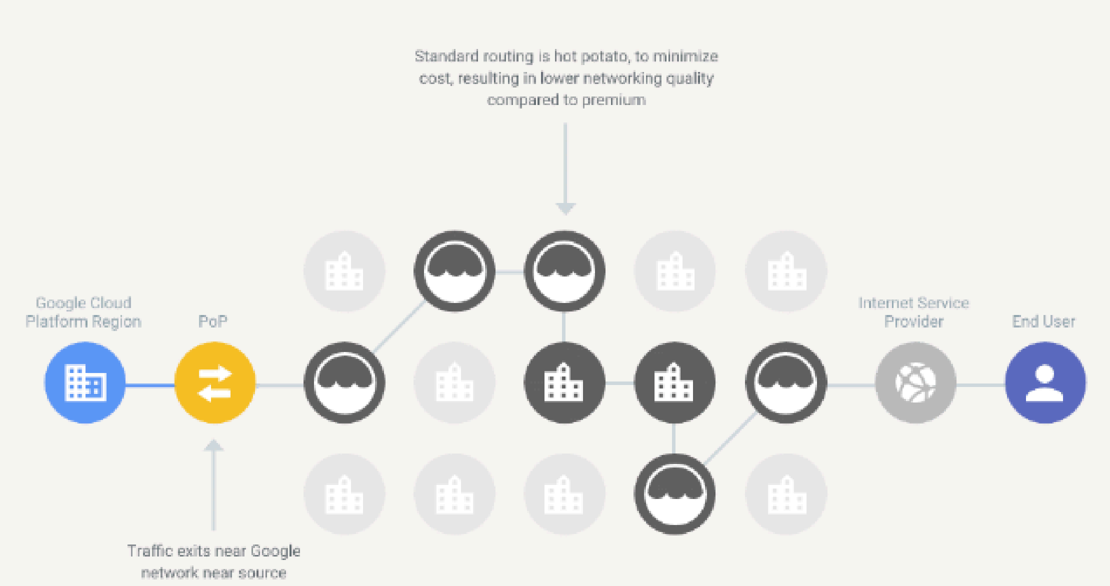
You also get the added benefit of global load balancing. One massive load balancer splits your traffic among servers into different regions instead of multiple load balancers for each region. This lets you use a single anycast IP address for your whole network, which is only possible when you have total control over the underlying hardware.

You're protected with a Global Service Level Agreement (SLA) in this setup. This binds Google through a contract to deliver a certain level of service, where you'd be compensated if they fail to reach those levels.

Standard Tier

The Standard tier is about optimizing your network for cost at the expense of peak performance. However, you will still access performance levels that are competitive with other cloud providers.

Most of your traffic travels the existing internet service provider's (ISP) networks with standard routing. Traffic is handled with a hot-potato approach where it's passed through many networks, autonomous systems, and ISPs before reaching its final destination. Speed is compromised when there are more hops before data reaches the users.


Standard routing is hot potato, to minimize cost, resulting in lower networking quality compared to premium
Google Cloud Platform Region
PoP
Internet Service Provider
End User
Traffic exits near Google network near source

There is also more complexity in the Standard tier, as you will only have access to regional cloud load balancing services. This means more "footwork" for you as the client, whereas you have global load balancing for your setup in the Premium tier.

The Standard tier also doesn't come with the Global SLA offered by the Premium tier. So, in the rare event of performance standards not being met, you won't be compensated for lack of service.

Competition to Google Cloud Network Tiers

The Google Cloud Platform was the first major public cloud provider to offer a tiered cloud network. Like any new and unique product launched in the cloud space, it wasn't long until the other significant players joined the party.

At the end of 2018, Amazon quietly introduced AWS Global Accelerator. Then, midway through 2020, Microsoft completed the holy trinity with Azure routing preference.

Both providers offer a similar solution to the GCP Network Service Tiers. Traffic is accepted close to the user through one of their many PoPs. It's then routed over their global networks.

Each offers a cold-potato approach to route the traffic through as few hops as possible to accelerate performance and ensure low latency. Like the GCP, both Azure and AWS solutions come with additional data transfer costs.

7. Continued Dedication to Network Expansion

Despite already offering planet-scale infrastructure, Google remains committed to growing and improving its global network. They focus on expanding into new regions and countries to produce more network edge locations.

This growth trend will, in turn, improve your ability to deliver an exceptional user experience to customers visiting your website. You'll be able to offer ever-lowering latency and ever-heightening performance through more locations across the world.

As of November 2021, Google has announced the opening of five new regions. These include Warsaw (Poland), Delhi (India), Melbourne (Australia), and Toronto (Canada).

But it doesn't stop there. In addition to plans for a new region in Israel, the following are coming soon to Google Cloud:

- **Doha (Qatar)**
- **Paris (France)**
- **Milan (Italy)**
- **Santiago (Chile)**
- **Madrid (Spain**
- **Turin (Italy)**
- **Columbus (US)**
- **Berlin (Germany)**

It's not just new regions worth mentioning, but also Google's commitment to better connecting their infrastructure with subsea cables. These new connections accelerate data delivery as they carry our digital lifeblood from one side of the globe to the other.

In 2021 alone, Google has announced the plans for six new subsea cables that will connect Africa, the Middle East, Europe, Asia, and the US:

- **Apricot:** A new subsea cable that will connect Singapore, Japan, Guam, the Philippines, Taiwan, and Indonesia — planned for arrival in 2024.
- **Blue:** A collaboration with Sparkle and others, connecting Italy, France, Greece, and Israel.
- **Dunant:** A submarine cable system crossing the Atlantic and connecting Virginia Beach in the US to Saint-Hilaire-de-Riez on the French Atlantic coast.
- **Echo:** A subsea cable from the United States to Asia, connecting California to Singapore, with a stopover in Guam.
- **Firmina:** An open subsea cable connecting the East Coast of the US to Las Toninas, Argentina, with additional landings in Praia Grande, Brazil, and Punta del Este, Uruguay.
- **Raman:** Another collaboration with Sparkle and others, connecting Jordan, Saudi Arabia, Djibouti, Oman, and India.

8. Ease of Setup

Setting up the infrastructure can be a daunting prospect when getting started with building a website, blog, or application. A series of challenges and decisions can quickly stack up, such as:

- Where to buy the domain
- Which content management system to use
- What type of server is needed
- How much storage you'll require
- Which database to attach

A top advantage of Google Cloud hosting is the ease of deploying a new web hosting solution. The service lets you choose from over 100 click-to-deploy web hosting solutions, greatly simplifying the setup and launch process.

These click-to-deploy solutions offer a preconfigured setup that typically comprises a virtual machine server, operating system, storage disk, content management system, and database. They're available at an affordable price without expending the time and effort required to source what you need to get started.

All Google Cloud's click-to-deploy hosting solutions are customizable. So, once you're up and running your day-one deployment, you can come back at a later time to adjust and further optimize the solution.

If you're considering launching your first blog or website, the Google Click to Deploy solution for WordPress is certainly worth a look. This solution offers an out-of-the-box setup of WordPress on the Compute Engine platform, available with a range of hosting options at various price points.

When comparing the top three cloud providers, Google Cloud seems to be the leader in easy setup. AWS offers competitive out-of-the-box web hosting solutions through Amazon Lightsail, but the process feels more involved; more configuration and thought are required to get started.

Azure is a bit of an enigma here. It also offers web hosting services, though it's not immediately apparent. It's contained within their Web Apps service, and it's pretty easy to miss since their site talks about web applications and not websites.

9. Amazing Support Service

A critical factor in choosing a web hosting provider is the quality of support you'll receive. Along the way, you're likely to encounter issues with your website hosting setup. When this happens, you'll want a quick, simple, and comprehensive support solution to troubleshoot whatever problems you've run into.

The support available from Google is undoubtedly one of the main advantages of Google Cloud hosting. This comes in three forms, which we've detailed below.

Comprehensive Paid Support

Choosing one of Google's paid Customer Care plans unlocks advanced support for the full range of their cloud technologies. There are three paid support plans available — Standard, Enhanced, and Premium support.

As you move upwards through the paid support plans, you unlock additional support formats, accelerated response times, more communication channels, greater availability, and escalation choices for more immediate issues.

With premium support, you get guaranteed response times in 15 minutes and 24/7 support 365 days of the year for critical-impact issues. It also comes with a dedicated Technical Account Manager to help optimize your Google Cloud Platform usage. You can fully customize and tailor your support across every product and service, even gaining access to hands-on training.

However, with support prices that can scale beyond $150,000 a year, it’s not for everyone.

Community Support
Even if you don’t choose a paid support plan, you gain free access to a vast community of enthusiasts and experts who can provide support as a Google Cloud customer.

The Google Cloud Community is home to 20,000 members. Here, you can meet industry peers, ask questions, and collaborate to find answers for the Google products you’ll use every day. You can also head over to Google Groups, where you’ll find a range of forums to discuss Google Cloud.

Extensive Support Documentation
The Google Cloud Platform has one of the most comprehensive collections of support documentation out there. If you have a mind for technology and prefer a DIY approach, you’ll find everything you need in that documentation to troubleshoot your web hosting issues.

The platform offers guides, code samples, architectural diagrams, best practices, tutorials, API references, and more, all so that you can learn about and troubleshoot every aspect of your Google Cloud setup. Best of all, it’s completely free.

10. State-of-the-Art Security

Security is a primary concern for any business. When choosing the Google Cloud Platform, you benefit from a security model developed over 15 years of continuous improvement through innovation. The same system secures their top products, including Gmail and Google Search.

You can also rest easy knowing that Google’s team of industry-leading security and privacy experts is dedicated to enhancing the platform’s state-of-the-art security.

Below is a selection of the top security features you’ll benefit from when choosing the Google Cloud platform:

- Encryption of your data both in transit and at rest, ensuring your sensitive information is protected by industry-leading 256-bit AES encryption, regardless of whether your data is stored on a Google persistent disk or traveling between you and their data centers.
- A team of vulnerability management experts dedicated to continuously scanning, identifying, and patching vulnerabilities in the Google Cloud Platform.
- Its own established global network and strong relationships with the world’s leading ISPs, improving the security of your data in transit with fewer hops across the public internet.

- Continued independent verification of their security, privacy, and compliance controls.
- Achieving industry-leading certifications including ISO/IEC 27001/27017/27018/27701, PCI DSS, GDPR, and HIPAA, and providing peace of mind that the platform is secure while also helping to further your compliance efforts through proxy.

Version Control: Git

What is Git?

Git is a distributed version control system that enables software development teams to have multiple local copies of the project's codebase independent of each other. These copies, or branches, can be created, merged, and deleted quickly, empowering teams to experiment, with little compute cost, before merging into the main branch (sometimes referred to as the master branch). Git is known for its speed, workflow compatibility, and open source foundation.

Most Git actions only add data to the database, and Git makes it easy to undo changes during the three main states.

Git has three file states: modified, staged, and committed.

1. A modified file has been changed but isn't committed to the database yet.
2. A staged file is set to go into the next commit.
3. When a file is committed, the data has been stored in the database.

With Git, software teams can experiment without fearing that they'll create lasting damage to the source code, because teams can always revert to a previous version if there are any problems.

What is version control and how can you document changes?

Version control — also known as source control or revision control — is software used to track revisions, solve integration conflicts in code, and manage different artifacts involved in software projects (e.g. design, data, images). Version control also enables frictionless communication, change, and reproducibility between developers and other team members.

With version control, you're able to track and merge branches, audit changes, and enable concurrent work to accelerate software delivery. Version control is the foundation and the stepping stone of DevOps, software architecture, and release patterns, without which an org cannot start. Using version control results in remarkable team productivity and software quality improvements.

Why should you use Git?

Software development teams prefer Git over other version control systems, like CVS, Mercurial, and Perforce, because Git has the adaptability, speed, and stability required to thrive in fast-paced markets. It's no wonder that 87.2% of developers use Git for version control. If organizations want to quickly meet customer demand and business goals, Git version control is the simplest way to empower developers.

Git's ability to store a complete history of a project locally is a benefit, because Git branches are lightweight and the protocol is fast, so contributors can be anywhere, even with a poor connection and still sync a local copy with any other team member.

The benefits of Git and a distributed version control system

1. **Git has workflow flexibility**

   With Git, teams can work together using various branching strategies that are not as possible in other version control systems. Git's branching capabilities offer users the ability to select a workflow depending on the size of a project or team or unique processes. Some of the most popular Git workflows include centralized, feature branching, trunk-based development, and GitFlow.

2. **Git is fast**

   Users have a local repository on their machine with a full history, so there's no lag in communicating with a server, which is an experience that users of centralized systems, such as CVS, Subversion, and Perforce, often encounter.

   Having a local version control means Git doesn't have to ping a server to view a project's history to identify changes made between versions. Git can immediately do a local difference calculation.

3. **Git is reliable**

   Git inherently has multiple backups, because each user has a local repository. If there's a crash, a copy could replace the main server. Another benefit of local repositories is that users can continue to commit offline if they're traveling or dealing with network issues.

4. **Git is collaborative**

   Git makes collaborative development easy with its branching model. People on your team can create a branch, experiment, and merge the code to the main branch if it works well. Git tracks changes during code reviews and team members can collaborate in merge requests (or pull requests).