 **Introduction**

* What HTTP is, why it matters.
  + - Port 80 and port 443 are both used for web traffic, but they differ significantly in their security. Port 80 is the default port for HTTP, which transmits data in plain text and is therefore unencrypted. Port 443 is the default port for HTTPS, which encrypts data using SSL/TLS, making it a more secure protocol which we discuss more below
  + HTTP stands for Hypertext Transfer Protocol
  + It establishes a set of rules for how web browsers and servers exchange data, dictating how clients (like your browser) request resources (like web pages) from servers
  + Operates on the client server model, and the core of http lies in requests and responses
  + Client specifies the source (webpage) the method (GET RETRIEVE POST to submit data)
  + Server reponds with error message or resource
  + Defines the format of the requests and repponses to make sure they understand eachother
  + It is used often with TCP/IP to establish reliable connections and secure connections like HTTPS, which is the secure version.
  + Without HTTP computers could not understand eachother or request and return resources effectively or at all—unless all computers (clients and servers alike) were the same type and spoke the same language. Additionally you wouldn’t be able to transfer/receive the files necessary to visit websites, communicate/send a text message, search, or even open your browser.
  + HTTP is important for the internet as well, giving us the foundation /back bone of communication over the internet.
* Scope of essay (evolution from early days → today → future).
  + We will be talking about the background, where it came from, how long its been around, what inspired the creator, who the creator is, modern vs old http, current role of http, where the future or http lies, and the market for http and how its impact affects us everyday people.

 **Background & Early History**

* HTTP/0.9 → 1.0 → 1.1
  + HTTP 0.9
    - HTTP 0.9 was the first version released in 1991
    - It was very simple designed to retrieve HTML documents
    - Html—or hyper text mark up language was initially developed by TIM BERNERS-LEE in late 1991. The first version was 1.0 and was released officially in 1993. While this discrepancy may seem strange—how can HTTP be created to retrieve HTML documents when HTML wasn’t around, that’s because they were created concurrently wile he was working at CERN in the late 1980s and early 1990s. HTML was made to structure and link documents based on SGML (Standard generalize mark up language) which preexisted HTML. HTTP is the proptocol to exchange these hypertext documents in real time over the internet. WWW bowser was the first web browser which could display and edit these documents. And HTTPD serve was to provide access to these documents. These composents were developed and implemented together so HTML was not around in the retrievable form before HTTP thy wer edesigned to work in tandem forom the inception of the world wide web
    - The first and only method it supported was GET allowing clients to request resources from a server
    - There were no headers, and therefore had no information about content type status code or other meta data
    - Responses was expected to only consist of requested HTML, which would lead to a lot of errors if the document was not that—they were handled by the server, returning a situation specific HTML file, but no distinct status codes (like 404, or 200 OK)
    - The connection as typically terminated immediately after the response was sent
  + HTTP 1.0
    - The second version and the application level protocol that forms the foundation of data communication on the World Wide Web introduced in 1006.
    - Stateless, interaction is independent meaning the server doesn’t retain informationabout the previous requests from the same client.
    - A separate TCP connection is established then closed after the response is delivered. Leads to overhead due to repeated connection set up and tear down
    - Defines fundamental request methods beyond get, like POST (send data to the server) and HEAD (to retrieve only header information.
    - Introduced the use of headers to convey meta data and status codes from requests and responses to indicate the outcomeof a request.
    - Host header wasn’t required by this specification, but became cruitial for virtual hosting allowing a single sevr to host multiple websites on the same IP
  + HTTP 1.1
    - Persistent connections implemented where a single TCP connection can be used for multiple requests and responses. This reduces latency as it avoids the overhead of repeatedly establusheing and closing TCP connections
    - Send multiple requests over the single connection improiving performance and allowing the server to process requests in parallel
    - Name based virtual hostng which enables multiple websites (virtual hosts) to be served from a single IP address officially through the host head which is crucial for modern web hosting where multiple website share the same server infrastructure
    - Enhanced caching – more sophisticated caching allowing clients and sevrers to move more effectively and utilize cached resources reduing the need to repeatedly fetch resources from the orgin sever
    - Also standardizes other aspects such as request methods (PUT DELETE OPTIONS) ERROR codes and content encoding
* Problems these versions solved.
  + HTTP 0.9 🡪 HTTP 1.0
    - Solved the problem OF LIMITED FUNCTIONALITY, Lack of meta data, inability to handle various file tupes, no feedback
    - Introduced clear status codes request processing feadback
    - http headers for flexible message structure
    - content type header to support various file formats
    - new request methods like POST HEAD
  + HTTP 1.0 🡪 HTTP 1.1
    - High connection overhead
    - Slow page load times from previous indivisual tcp connection requirement
    - Persistent connections (Keep-Alive)
    - \pipelining (send multiple requests)
    - Chunked transfer encoding (server can send a response before the total length is known)
    - Content negotiating (servers and clients can agree on the contents language, encoding, and tyupe to ensure the most appropriate content is echanged
    - More robust cache support

 **Modern HTTP**

* HTTP/2 innovations.
  + HTTP/2 innovations center on improving web performance through multiplexing, which allows multiple requests/responses over a single connection; header compression (HPACK) to reduce overhead; a binary protocol for more efficient data transmission; and server push, enabling servers to proactively send resources to the client, all leading to faster page loads and reduced latency. These features work together to create more efficient and scalable web communication, though they require a secure HTTPS connection to function.
  + Key Innovations:
  + Multiplexing:
  + Instead of opening multiple TCP connections for different resources (like an image and CSS), HTTP/2 uses a single, persistent connection to handle multiple requests and responses simultaneously. This reduces the overhead associated with establishing and tearing down connections and helps overcome the limitations of HTTP/1.1.
  + Header Compression (HPACK):
  + HTTP/1.x repeated many headers with each request and response, creating significant overhead. HTTP/2 uses the HPACK mechanism to compress these headers, significantly reducing the amount of redundant data sent over the network.
  + Binary Protocol:
  + The underlying data is transmitted in a binary format, making it more efficient, less error-prone, and easier for computers to process compared to the text-based format of HTTP/1.x.
  + Server Push:
  + This feature allows the server to send resources that it anticipates the client will need, even before the client explicitly requests them. For example, when a client requests an HTML page, the server can also push associated CSS and JavaScript files, speeding up rendering.
  + Stream Prioritization & Dependencies:
  + HTTP/2 includes mechanisms for prioritizing streams, ensuring that the most important resources are delivered first.
  + Flow Control:
  + To prevent one slow stream from blocking others on the same connection, HTTP/2 implements flow control to manage data transfer limits on a per-stream basis.
  + How They Work Together:
  + These innovations enable HTTP/2 to use a single TCP connection for all requests and responses. A stream ID tags each packet, allowing the server to identify and process requests in parallel. Multiplexing and header compression reduce network congestion and latency, while server push and stream prioritization further enhance the efficiency and speed of resource delivery.
* HTTP/3 + QUIC.
  + The most recent major version of the HTTP protocol is HTTP/3, which was standardized and published by the [Internet Engineering Task Force (IETF)](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifMItWL4j0HOXJwttsMoi8NIS5WNiA%3A1756140380514&q=Internet+Engineering+Task+Force+%28IETF%29&sa=X&ved=2ahUKEwjN74fctKaPAxWqEEQIHVdhPDMQxccNegQIBBAC&mstk=AUtExfBBTL7fteSQDIJlMUGDRLRvaEWx27mL6H0KFd8dgbUfg6l0pjqhVw11DgaVwl-Ey3pG9_3TSThMA8WwyeSQg0RxcWDJTJSjRbnYfNepC6IkTNgQAL054TyUWz3nBV1TtiE2--T2P0UKJS9drmkmf671MUi9N2b8s11IXq40qa8IWqoQRCofLgUneQOJaWeLtxoE5CQj5ffS8Cop9Oq323b5lJHN7Y_HbhObc1oFVTddQ8yqEEA-fWIEDRZ1JP7uSWXrKxMeVOegdPeqQp1qiPpXos6DLVqqli_bn-_yVF1aM7NR8TD-IaTcm3Q5oIQXmzCBa_w5w3m_XyC4PWd4zZE6ver2jnxMncTp6Jqqd_19jtWa7nz4dFHvrJE5XIOOJKNCE4KAETCxKkvZE9l1vg&csui=3) in 2022. HTTP/3 uses the [QUIC transport protocol](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifMItWL4j0HOXJwttsMoi8NIS5WNiA%3A1756140380514&q=QUIC+transport+protocol&sa=X&ved=2ahUKEwjN74fctKaPAxWqEEQIHVdhPDMQxccNegQIBBAD&mstk=AUtExfBBTL7fteSQDIJlMUGDRLRvaEWx27mL6H0KFd8dgbUfg6l0pjqhVw11DgaVwl-Ey3pG9_3TSThMA8WwyeSQg0RxcWDJTJSjRbnYfNepC6IkTNgQAL054TyUWz3nBV1TtiE2--T2P0UKJS9drmkmf671MUi9N2b8s11IXq40qa8IWqoQRCofLgUneQOJaWeLtxoE5CQj5ffS8Cop9Oq323b5lJHN7Y_HbhObc1oFVTddQ8yqEEA-fWIEDRZ1JP7uSWXrKxMeVOegdPeqQp1qiPpXos6DLVqqli_bn-_yVF1aM7NR8TD-IaTcm3Q5oIQXmzCBa_w5w3m_XyC4PWd4zZE6ver2jnxMncTp6Jqqd_19jtWa7nz4dFHvrJE5XIOOJKNCE4KAETCxKkvZE9l1vg&csui=3), which runs over [UDP](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifMItWL4j0HOXJwttsMoi8NIS5WNiA%3A1756140380514&q=UDP&sa=X&ved=2ahUKEwjN74fctKaPAxWqEEQIHVdhPDMQxccNegQIBBAE&mstk=AUtExfBBTL7fteSQDIJlMUGDRLRvaEWx27mL6H0KFd8dgbUfg6l0pjqhVw11DgaVwl-Ey3pG9_3TSThMA8WwyeSQg0RxcWDJTJSjRbnYfNepC6IkTNgQAL054TyUWz3nBV1TtiE2--T2P0UKJS9drmkmf671MUi9N2b8s11IXq40qa8IWqoQRCofLgUneQOJaWeLtxoE5CQj5ffS8Cop9Oq323b5lJHN7Y_HbhObc1oFVTddQ8yqEEA-fWIEDRZ1JP7uSWXrKxMeVOegdPeqQp1qiPpXos6DLVqqli_bn-_yVF1aM7NR8TD-IaTcm3Q5oIQXmzCBa_w5w3m_XyC4PWd4zZE6ver2jnxMncTp6Jqqd_19jtWa7nz4dFHvrJE5XIOOJKNCE4KAETCxKkvZE9l1vg&csui=3), to improve performance, reliability, and security over previous versions like HTTP/2 that relied on [TCP](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifMItWL4j0HOXJwttsMoi8NIS5WNiA%3A1756140380514&q=TCP&sa=X&ved=2ahUKEwjN74fctKaPAxWqEEQIHVdhPDMQxccNegQIBBAF&mstk=AUtExfBBTL7fteSQDIJlMUGDRLRvaEWx27mL6H0KFd8dgbUfg6l0pjqhVw11DgaVwl-Ey3pG9_3TSThMA8WwyeSQg0RxcWDJTJSjRbnYfNepC6IkTNgQAL054TyUWz3nBV1TtiE2--T2P0UKJS9drmkmf671MUi9N2b8s11IXq40qa8IWqoQRCofLgUneQOJaWeLtxoE5CQj5ffS8Cop9Oq323b5lJHN7Y_HbhObc1oFVTddQ8yqEEA-fWIEDRZ1JP7uSWXrKxMeVOegdPeqQp1qiPpXos6DLVqqli_bn-_yVF1aM7NR8TD-IaTcm3Q5oIQXmzCBa_w5w3m_XyC4PWd4zZE6ver2jnxMncTp6Jqqd_19jtWa7nz4dFHvrJE5XIOOJKNCE4KAETCxKkvZE9l1vg&csui=3).
  + HTTP/3 offers faster connection establishment and reduced latency, particularly on lossy or congested networks.
  + Reduced Head-of-Line Blocking:
  + Unlike HTTP/2, which could experience head-of-line blocking on a single TCP connection, QUIC's multiplexing over UDP prevents packet-level blocking from impacting other streams, leading to better performance.
  + Built-in security:
  + QUIC includes TLS 1.3 for comparable confidentiality and integrity, enhancing the security of web communication.
  + Maturity and Adoption:
  + Although the QUIC and HTTP/3 specifications were finalized earlier, their official publication in 2022 synchronized the standards with widespread adoption.
  + How to check your HTTP version:
  + You can typically find the HTTP version your browser is using by looking at the URL in the address bar:
  + http://: indicates an HTTP connection.
  + https://: indicates an HTTPS (HTTP Secure) connection, which uses encryption.
  + To see the specific HTTP version (e.g., HTTP/3) your browser is using, you may need to use your browser's developer tools, which can display network request details, including the HTTP version.

 **Current Role in the Web**

* Use than just browsers like APIs, cloud computing.

Here's how HTTP is used in APIs:

* **Request-Response Model:**

HTTP operates on a request-response model. A client sends an HTTP request to an API endpoint on a server, and the server processes the request and sends an HTTP response back to the client.

* **HTTP Methods (Verbs):**

HTTP defines various methods (also known as verbs) that indicate the intended action on a resource. Common methods used in APIs include:

* + **GET:** Retrieves data from the server.
  + **POST:** Submits data to the server, often to create a new resource.
  + **PUT:** Updates an existing resource on the server.
  + **DELETE:** Removes a resource from the server.
  + **PATCH:** Partially updates an existing resource.
* **URLs (Endpoints):**

APIs expose specific resources through URLs, which are the addresses clients use to interact with the API. For example, a GET request to /users/123 might retrieve information about a user with ID 123.

* **Headers:**

HTTP requests and responses include headers that provide metadata about the communication. These can include authentication tokens, content types (e.g., JSON, XML), caching instructions, and more.

* **Status Codes:**

The server's HTTP response includes a status code, a three-digit number indicating the outcome of the request. Examples include:

* + 200 OK: The request was successful.
  + 201 Created: A new resource was successfully created.
  + 400 Bad Request: The client sent an invalid request.
  + 404 Not Found: The requested resource could not be found.
  + 500 Internal Server Error: An error occurred on the server.
* **Body:**

HTTP requests and responses can optionally contain a body, which carries the actual data being sent or received. This data is often formatted as JSON or XML.

* **Statelessness:**

HTTP is inherently stateless, meaning each request from a client to a server is independent. APIs often manage state through mechanisms like sessions or tokens to maintain user context across multiple requests.

In essence, HTTP provides the communication protocol and structure for how clients and servers interact when using many APIs, dictating the methods, addressing, data formatting, and status reporting for these interactions.

Key Uses of HTTP in Cloud Computing

**[Accessing Cloud Services:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Accessing+Cloud+Services&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIEBAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Accessing+Cloud+Services&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIEBAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

Users and applications interact with cloud resources by sending HTTP requests to unique endpoints provided by cloud services.

**[Web-Based APIs:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Web-Based+APIs&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIJBAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Web-Based+APIs&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIJBAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

Cloud providers use HTTP to offer APIs that allow developers to build applications and manage cloud resources, such as creating virtual machines or deploying container images.

**[Serverless Computing:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Serverless+Computing&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIJRAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Serverless+Computing&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIJRAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

Cloud platforms like Google Cloud Run use HTTP to receive requests for serverless functions, automatically scaling the code's execution based on traffic.

**[Microservices Communication:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Microservices+Communication&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIHhAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Microservices+Communication&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIHhAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

With [HTTP/2](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=HTTP%2F2&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIFhAB&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3) and protocols like [gRPC](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=gRPC&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIFhAC&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank), cloud applications can use HTTP for high-performance, low-latency communication between microservices, a common pattern in cloud-native architectures.

**[Data Exchange:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Data+Exchange&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIIxAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Data+Exchange&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIIxAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

HTTP enables the exchange of data between cloud services and users' devices, forming the basis for cloud-based software and applications.

Security Considerations

**[HTTP vs. HTTPS:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=HTTP+vs.+HTTPS&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQILhAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=HTTP+vs.+HTTPS&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQILhAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

While HTTP transmits data in plain text, HTTPS encrypts data using [SSL/TLS](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=SSL%2FTLS&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQIMhAB&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3), providing essential security for cloud-based applications, especially those handling sensitive information.

**[Security Best Practices:](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Security+Best+Practices&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQINRAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&cs=1&sxsrf=AE3TifOiz6MjWB_-RGCJUVJS7PfaGvp_PA%3A1756140756389&q=Security+Best+Practices&sa=X&ved=2ahUKEwjkr66PtqaPAxW3KFkFHWegE7IQxccNegQINRAD&mstk=AUtExfCRX9TtYsJWyMpaQYMhgUdLbPuk1jO8NTeqvy1lD44j2bWPwRad6wjVe6lFf36e3jR8VaIKl6IXiLB1zey2AjXRH2sTXo4MrHPl85Q4VzXrWhiRfW7Su4v2ZTckTsVi7rkM0XVNOrwvtUmWhJ3RA7QRPvM5BoJ_sFoQxyvnVEvY8RUqPj9K-9lXaJ90jIsyPzmn1u93VjzVXLKHYeopyKyW2AjSl8Kr8slLX03oSlDKtG2JATgHJWNqEeiu9YZv9A-rHXtiQNPKNPMymtKmbkr4p1f6i_4SFRDgXVKNjvm7T4l6ZxdSNwlQbZL1wSysRsYk-hBVhzuwV3rj6uUYiE7ad4MFvGnbrFcPAcXjADldH2rvH3NmQin9jGK5UHUbVVw0OxilZldGHiLxo4Q-jw&csui=3" \t "_blank)

Cloud services often redirect HTTP requests to HTTPS to ensure data is encrypted and to enhance user trust and compliance with security standards.

* Security (HTTPS/TLS).

What TLS does for HTTPS:

* **Encryption:** TLS encrypts the data exchanged between your browser and the website's server, preventing attackers from intercepting and stealing information.
* **Authentication:** It verifies that the server you are connecting to is legitimate and not an imposter, using an [SSL/TLS certificate](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=SSL%2FTLS+certificate&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIoQEQAQ&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3).
* **Data Integrity:** It ensures that the data hasn't been tampered with during transit.

How it works:

**[1. TLS Handshake:](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=TLS+Handshake&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIyAIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=TLS+Handshake&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIyAIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)

When you visit an HTTPS site, your browser and the server perform a TLS handshake.

**[2. Key Exchange:](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=Key+Exchange&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIsgIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=Key+Exchange&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIsgIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)

Using public-key cryptography, the server shares a public key with the browser.

**[3. Session Keys:](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=Session+Keys&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIygIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=Session+Keys&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIygIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)

This public key is used to encrypt and send data, allowing the server to decrypt it with its private key. Together with other information, new, temporary [session keys](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=session+keys&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIxgIQAQ&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3) are generated for encrypting all subsequent communication for that session.

**[4. Secure Communication:](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=Secure+Communication&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIyQIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)**

[.](https://www.google.com/search?sca_esv=8e23599b05dcda53&sxsrf=AE3TifNRymwdSqotAyesCTOW2cZMRliX1g%3A1756140806716&q=Secure+Communication&sa=X&ved=2ahUKEwj119antqaPAxXFEGIAHfaKCjUQxccNegUIyQIQAw&mstk=AUtExfDp6PNGIdCLUjALHH3odwzphK2RU227VBPfcYJYKGGUYS9wZgPSvJqA_Xsdx8TgYCZVdt7JOXHqOToJ0PgdKA8g_kFsVL79Jly_cfq72jiFdCeCcJX-j7AXKt0K4B74foZ1cASioEsuc_hkSInRurD0TwZOIXCtFPlpmR8_KeBn1rSeE7QmwUnUqyolu8PYSZt_GV5Oc1m_l7sXcsbpWrA83rYp_WzBmmHSy3i4L_ttuLNJo3VT6SJuegarhE2kKp2AbLXXnzXUwyBocIIV-vjE73ldWtFOwGTaJZBqhrOzboCxouHOP1VnsLStXP4Tsn9TJiJ4x2tllZupNCdUp1ZB7tkLhA6Jn9VjZxWQnRrSaVNzO_5XN5CXrxWdI5peQGdmpyNjVmjWKnJJiLsQ5g&csui=3" \t "_blank)

Now, the browser and server can securely send information back and forth using these session keys.

Why it matters:

* **Privacy:** Keeps your online activity private, protecting usernames, passwords, and other personal information.
* **Security:** Protects sensitive data, especially during online purchases and banking transactions.
* **Trust:** Builds user confidence that the website is secure and authentic.

 **Future Directions**

* Performance goals.
* Integration with IoT, AR/VR, 5G.
* AI-driven optimizations.

 **Market & Industry Impact**

* Companies, organizations, adoption rates.
* Real-world case studies.

 **Conclusion**

* Reflection: from simple text protocol to backbone of modern internet.

 **References (10 minimum)**

<https://en.wikipedia.org/wiki/HTTP#:~:text=HTTP/1.1%20and%20earlier%20have,%2C%20stylesheets%2C%20etc>.).

<https://en.wikipedia.org/wiki/HTTP#:~:text=The%20client%20sends%20its%20HTTP%20request%20message.,or%20other%20information%20may%20also%20be%20returned>.

[https://stackoverflow.com/questions/246859/http-1-0-vs-1-1 - :~:text=HTTP%201.1%20comes%20with%20the,this%20is%20similar%20to%20pragma](https://stackoverflow.com/questions/246859/http-1-0-vs-1-1#:~:text=HTTP%201.1%20comes%20with%20the,this%20is%20similar%20to%20pragma)

[https://www.imperva.com/learn/performance/http2/ - :~:text=HTTP/2%20compatible.-,From%20SPDY%20to%20HTTP/2,and%20improved%20user%20experience%2C%20including:](https://www.imperva.com/learn/performance/http2/%20-%20:~:text=HTTP/2%20compatible.-,From%20SPDY%20to%20HTTP/2,and%20improved%20user%20experience%2C%20including:)

[https://httpwg.org/specs/rfc9112.html - :~:text=An%20HTTP/1.1%20message%20consists,and%20an%20optional%20message%20body.](https://httpwg.org/specs/rfc9112.html#:~:text=An%20HTTP/1.1%20message%20consists,and%20an%20optional%20message%20body.)

<https://www.w3.org/Protocols/HTTP/1.0/spec.html>

[https://www.cs.uic.edu/~ckanich/cs484/f24/readings/chapter-1-http/http-2/index.html - :~:text=a%20single%20connection.-,Header%20Compression:%20Reducing%20Overhead,to%  
https://www.cncf.io/blog/2018/07/03/http-2-smarter-at-scale/ - :~:text=This%20is%20all%20fairly%20standard,care%20about%20cross%2Dstream%20problems.20get%20up%20to%20speed.](https://www.cs.uic.edu/~ckanich/cs484/f24/readings/chapter-1-http/http-2/index.html#:~:text=a%20single%20connection.-,Header%20Compression:%20Reducing%20Overhead,to%20get%20up%20to%20speed.)

<https://www.youtube.com/watch?v=fVKPrDrEwTI&t=17>

<https://cloud.google.com/apis>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Evolution_of_HTTP>

[https://cloud.google.com/run/docs/configuring/http2 - :~:text=and%20manage%20access.-,Supported%20HTTP/2%20services,to%20push%20data%20to%20clients.](https://cloud.google.com/run/docs/configuring/http2#:~:text=and%20manage%20access.-,Supported%20HTTP/2%20services,to%20push%20data%20to%20clients.)

<https://aws.amazon.com/compare/the-difference-between-https-and-http/>

[https://howhttps.works/https-ssl-tls-differences/ - :~:text=Click%20here%20to%20check%20the,you%20know%20the%20hiss%2Dtory.](https://howhttps.works/https-ssl-tls-differences/#:~:text=Click%20here%20to%20check%20the,you%20know%20the%20hiss%2Dtory.)

[https://www.digicert.com/tls-ssl/tls-ssl-certificates - :~:text=Informational%20sites%20&%20blogs,info%20they%20share)%20is%20secure.](https://www.digicert.com/tls-ssl/tls-ssl-certificates#:~:text=Informational%20sites%20&%20blogs,info%20they%20share)%20is%20secure.)

<https://https.cio.gov/technical-guidelines/>