1.Write a blog on Difference between HTTP1.1 vs HTTP2.

HTTP1.1	HTTP2
HTTP/1.1 was created in 1997 and is an improvement over HTTP/1.0. It uses a single persistent connection between the client and server to send multiple requests and responses	HTTP/2 is a protocol for transmitting data over the internet and is the second major version of the HTTP network protocol used by the World Wide Web. It was developed as a successor to the original HTTP/1.1 protocol to address its limitations and improve overall performance. Here are some key features and aspects of HTTP/2:
Persistent connections: HTTP/1.1 supports persistent connections, which allows multiple requests to be sent over the same connection. This reduces the overhead of establishing and tearing down connections for each request and response.	**Multiplexing:** HTTP/2 allows multiple requests and responses to be sent concurrently over a single connection. This helps in reducing latency and improving overall page load times by efficiently using available network resources.
Pipelining: HTTP/1.1 supports pipelining, which allows the client to send multiple requests without waiting for the server to respond to each request. This can improve the performance of applications that require multiple requests to be sent in quick succession.	**Header Compression:** HTTP/2 uses header compression (HPACK) to reduce overhead by compressing header information. This results in lower data transfer requirements and faster communication between clients and servers.
Caching: HTTP/1.1 supports caching, which allows frequently requested resources to be stored on the client or server. This can reduce the amount of data that needs to be transferred over the network and improve the performance of applications.	**Binary Protocol:** The binary framing in HTTP/2 is more efficient than the text-based format of HTTP/1.1. This enables faster parsing and processing of messages, contributing to improved performance.
Range requests: HTTP/1.1 supports range requests, which allows clients to request a specific portion of a resource. This can be useful for applications that need to download large files or portions of files.	**Server Push:** HTTP/2 supports server push, allowing the server to proactively send resources to the client before they are requested. This can lead to faster page loads by preemptively providing necessary assets

Content negotiation: HTTP/1.1 supports content negotiation, which allows the client and server to agree on the format and language of the message. This can be useful for applications that need to support multiple languages or formats.	**Prioritization:** HTTP/2 introduces prioritization of requests, enabling more critical resources to be delivered first. This feature helps optimize the loading of web pages and enhances the user experience.
Status codes: HTTP/1.1 uses status codes to indicate the status of the request or response. This can help developers diagnose and fix issues with their applications.	**Backward Compatibility:** HTTP/2 is designed to be backward compatible with HTTP/1.1. This means that existing websites can upgrade to HTTP/2 without requiring a complete overhaul of their infrastructure, promoting a smooth transition.
Compatibility: HTTP/1.1 is widely supported by web servers, web browsers, and other HTTP clients. This makes it a reliable choice for building web applications.	**TLS Support:** While not mandatory, HTTP/2 is often used with TLS to ensure a secure communication channel. The use of TLS enhances the security of data exchanged between clients and servers.
Human-readable messages: HTTP/1.1 is a text-based protocol, which means that the messages are human-readable and can be easily understood by developers.	**Reduced Latency:** Multiplexing, header compression, and other optimizations in HTTP/2 contribute to a reduction in latency. Faster communication between clients and servers leads to quicker page rendering and a more responsive user experience.
Simple: HTTP/1.1 is a simple protocol that is easy to implement and use.	**Resource Efficiency:** The efficient use of a single connection for multiple requests reduces the need for establishing and tearing down connections repeatedly. This resource efficiency results in faster communication and improved scalability.
Backward compatibility: HTTP/1.1 is backward compatible with HTTP/1.0, which means that applications built using HTTP/1.0 can still work with HTTP/1.1 servers.	**Improved User Experience:** The combination of reduced latency, faster page loads, and enhanced prioritization contributes to an overall improved user experience. Websites that adopt HTTP/2 may provide users with a more seamless and responsive browsing experience.