

Finesse에 대한 WebSocket 연결 이해

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[웹 소켓](#)

[WebSockets는 어떻게 작동합니까?](#)

[HTTP](#)

[HTTP 문제](#)

[SSE](#)

[WebSocket 작업](#)

[WebSocket 디버깅](#)

[관련 정보](#)

소개

이 문서에서는 WebSocket 연결에 대해 완벽하게 설명하므로 트러블슈팅 과정에서 기본 프로세스를 철저히 이해합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

구성 요소 Used

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco Finesse
- UCCX

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

웹 소켓은 클라이언트와 서버 간의 지속적인 연결입니다.

웹 소켓

영구 연결이라는 용어는 무엇을 의미합니까?

클라이언트와 서버의 연결이 설정되면 클라이언트와 서버가 언제든지 데이터를 보내고 받을 수 있다는 의미입니다.

양방향 전이중 연결입니다.

서버는 클라이언트 요청이 데이터를 푸시할 때까지 기다릴 필요가 없습니다.

마찬가지로 클라이언트는 서버에 새 데이터를 전송하기 위해 매번 새 연결을 만들 필요가 없습니다

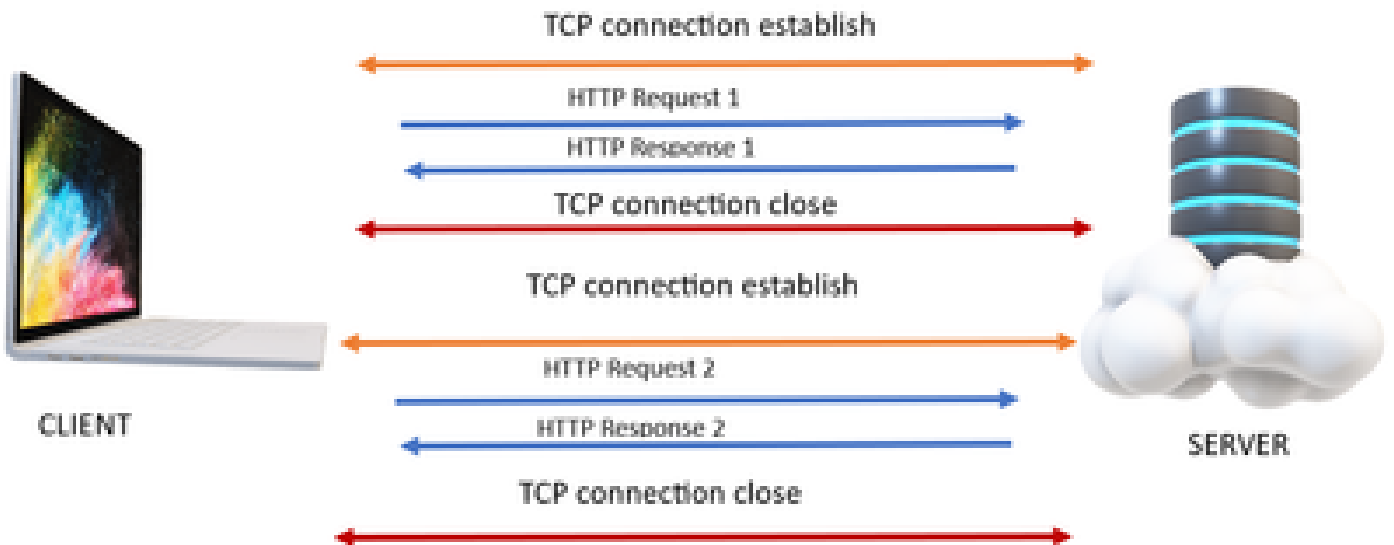


웹 소켓 연결은 실시간 데이터 업데이트가 필요한 응용 프로그램에서 주로 사용됩니다.

예를 들어 주식 거래 앱, 메시징 앱, 당사의 경우 Cisco Finesse입니다.

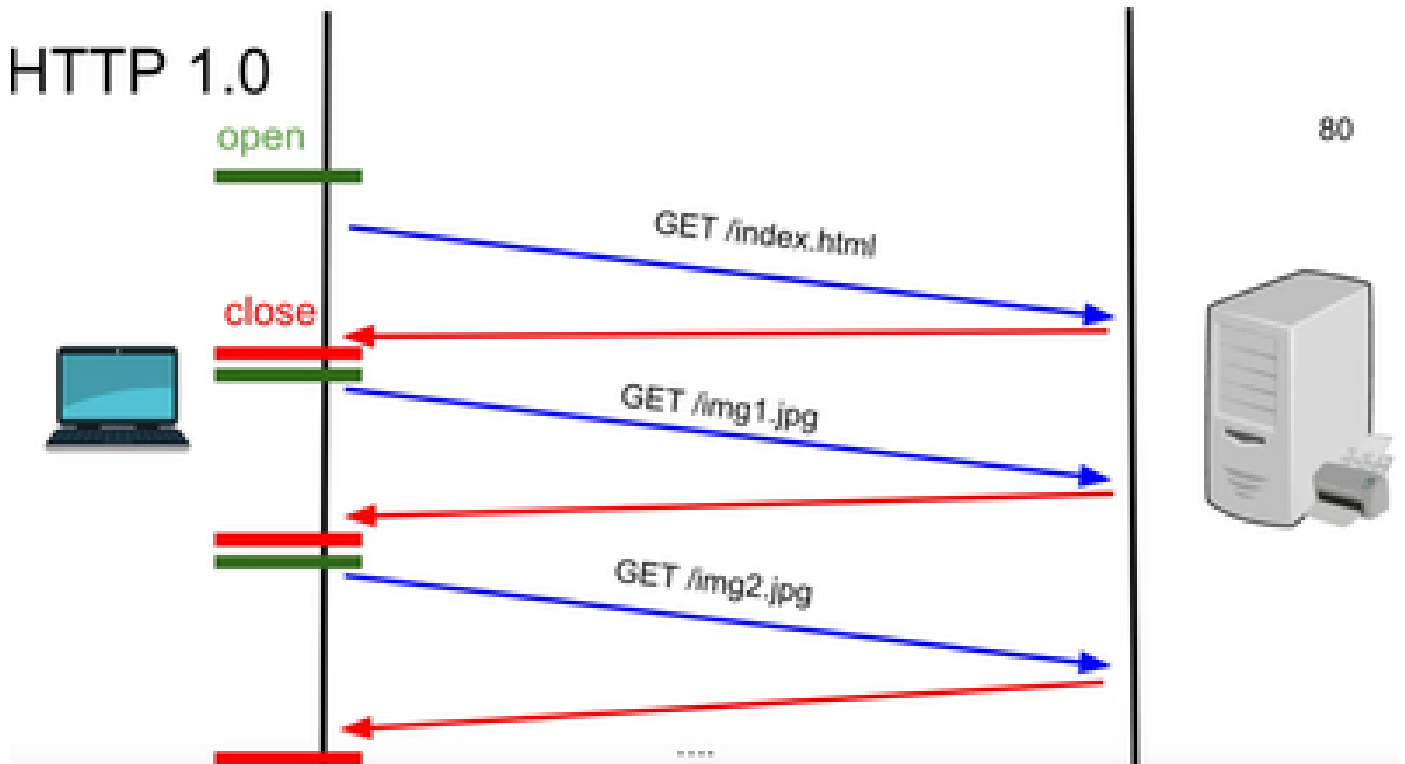
WebSockets는 어떻게 작동합니까?

고려 사항:

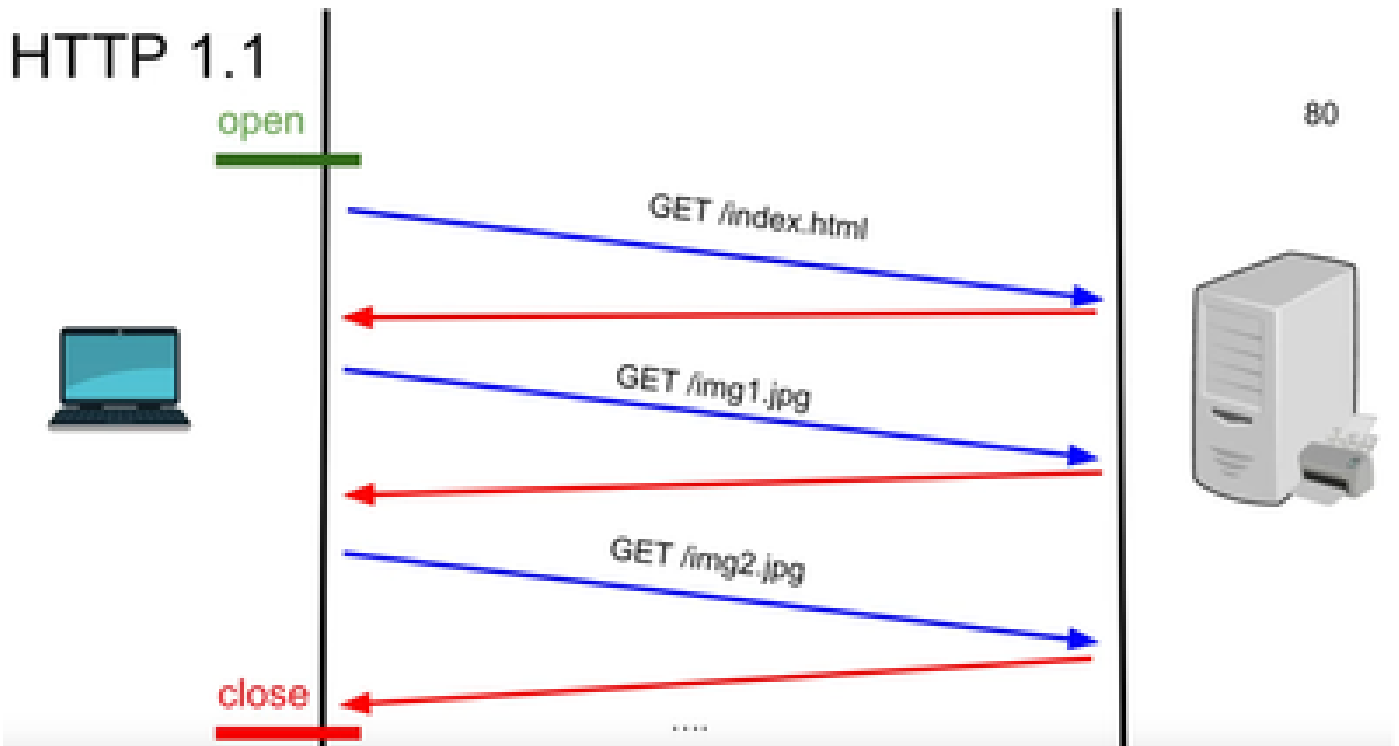


HTTP

1. TCP 연결(3-way handshake)이 발생합니다.
2. 그런 다음 클라이언트가 HTTP 요청을 보냅니다.
3. 서버가 HTTP 응답을 보냅니다.
4. 요청 응답 주기가 한 번 지나면 TCP 연결이 닫힙니다.
5. 새 HTTP 요청의 경우 다시 TCP 연결이 먼저 설정됩니다.



HTTP 1.0 - 모든 요청 응답 후 TCP 핸드셰이크가 다른 HTTP 요청 응답에 대해 다시 시작됩니다.



HTTP 1.1 - 데이터를 보내고 받은 다음 연결을 닫을 수 있으므로 이 연결이 작동했습니다.

이 역시 클라이언트가 요청하지 않아도 서버가 일부 데이터를 보낼 수 있어 실시간 앱에 적합하지 않았다. 따라서 이 모델은 효과적이지 않습니다.

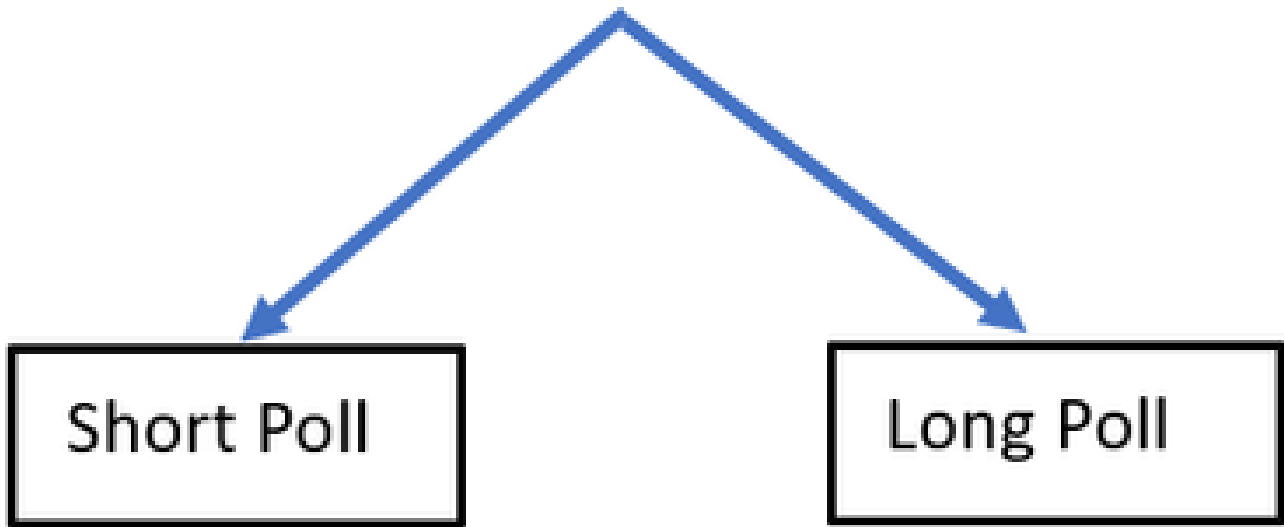
HTTP 문제

문제는 실시간 시스템에서 시작됩니다.

실시간 업데이트가 필요한 웹 사이트의 경우 서버에서 업데이트를 얻기 위해 매번 HTTP 요청을 보내는 것은 매우 어렵고 대역폭을 많이 사용하며 과부하를 유발합니다.

이를 해결하기 위해 Polling이라는 HTTP의 메커니즘을 사용합니다.

POLLING



Short Poll - 는 요청 및 응답에 대해 짧은 고정 타이머가 설정된 경우 구현됩니다. 예를 들어 구현에 따라 05초 또는 1초입니다.

다른 쪽에서 업데이트를 제공하지 않으면 해당 시간 내에 빈 응답을 얻어 리소스를 낭비할 수 있습니다.

긴 설문 조사 - 다소 짧은 설문 조사를 극복하지만, 응답을 기다리는 고정된 시간이 있습니다.

짧은 폴링보다 상대적으로 길지만 여전히 고정되어 있는 기간 내에 응답이 없는 경우 다시 시간 제한을 요청합니다.

따라서 폴링은 이러한 문제를 극복하는 최선의 방법이 아닙니다.

이를 위해 사용할 다른 방법을 SSE라고 합니다.

SSE

서버가 보낸 이벤트

이 경우 서버와 클라이언트 간에 단방향 연결이 설정되므로 서버에서 어느 시점에서든 데이터를 클라이언트에 보낼 수 있습니다.

여기서 주의할 점은 단방향 연결이라는 것입니다. 즉, 서버만 클라이언트로 데이터를 전송할 수 있고 그 반대로는 전송할 수 없습니다.

활용 사례의 예로는 서버에서 클라이언트로 대량 알림 또는 업데이트가 있습니다. 예를 들어 뉴스 라이브 업데이트, Instagram 라이브 등이 있습니다.

이는 실시간 업데이트 및 메시징과 관련된 애플리케이션에는 그다지 효과적이지 않습니다.

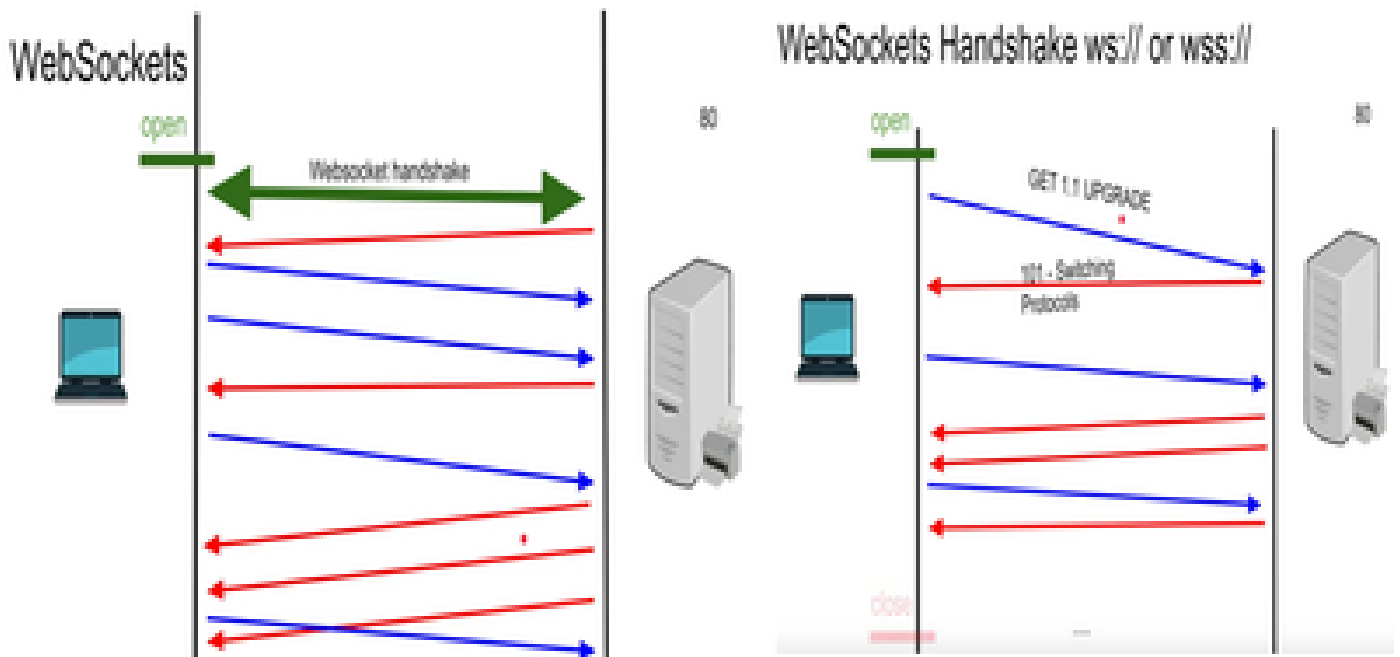
웹 소켓 연결은 지속적인 양방향 전이중 연결입니다.

서버와 클라이언트 간의 전화 통화일 수 있으며, 이 경우 모든 상대방이 언제든지 다른 상대방과 통화할 수 있습니다.

WebSocket 작업

1. 웹소켓 연결을 설정하기 위해 클라이언트는 업그레이드되거나 업데이트된 헤더와 함께 HTTP 핸드셰이크 요청을 보냅니다.
 1. 즉, 클라이언트는 지금 HTTP를 통해 실행되지만 이제부터는 Websocket 연결로 이동한다고 서버에 알립니다.
2. 그런 다음 서버는 HTTP 101 응답으로 응답합니다. 즉 sit가 프로토콜 응답을 교환합니다.
3. 그런 다음 WebSocket 연결이 설정됩니다.

이제 서버와 클라이언트가 동일한 연결을 사용하여 언제든지 서로 데이터를 전송할 수 있습니다.



WebSocket 디버깅

이때 Finesse 클라이언트에 로그인하고 네트워크 디버그가 표시되면 다음과 같이 표시됩니다.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:8080/...	/WS/	openfire.js:5282 (websocket)	plain	333 B	0 B

메서드 - GET

도메인 - 서버 이름

파일 - /WS/

개시자 - Openfire.js - 웹소켓

요청 및 응답 검토:

요청

가져오기

체계: wss

호스트: uccxpub.prabhat.com:8445

파일 이름: /ws/

Address(주소): uccx 서버의 IP

상태: 101

스위칭 프로토콜

버전HTTP/1.1

응답 헤더

연결: 업그레이드

업그레이드: 웹소켓

Request – open

Response

PLAIN

http://jabber.org/protocol/caps" hash="sha-1" node="

<https://www.igniterealtime.org/projects/openfire/>" ver="k3mOuil8afx3OTZxYy6yxLmFsok="/>

Request - auth

YWRtaW5pc3RyYXRvckB1Y2N4cHVlLnByYWJoYXQuY29tAGFkbWluaXN0cmF0b3IAMTIzNA==

Response

Request – XMPP Bind Bind request to Bind the resource which in this case is desktop with a jabber id

desktop

Response – XMPP Bind where User ID is given a jabber id

```
administrator@uccxpub.prabhat.com/desktop
```

```
administrator@uccxpub.prabhat.com/desktop
```

Presence request

Presence response

```
http://jabber.org/protocol/caps" hash="sha-1" node="
```

```
http://www.igniterealtime.org/projects/smack" ver="NfJ3fII83zSdUDzCEICtby pursw=">
```

```
http://jabber.org/protocol/caps" hash="sha-1" node="
```

```
http://www.igniterealtime.org/projects/smack" ver="NfJ3fII83zSdUDzCEICtby pursw=">
```

PUBSUB request – Requesting to subscribe the user to the pubsub node so that all the events on the user are monitored.

Response – user subscribed.

```
http://jabber.org/protocol/pubsub">
```

PUBSUB request – Requesting to subscribe the Team to the pubsub node so that all the events on the team are monitored.

Response – Team subscribed

[http://jabber.org/protocol/pubsub">](http://jabber.org/protocol/pubsub)

관련 정보

- [Cisco 기술 지원 및 다운로드](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.