

FinesseのWebSocket接続について

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[Webソケット](#)

[WebSocketsの仕組み](#)

[HTTP](#)

[HTTPに関する問題](#)

[SSE](#)

[WebSocketアクション](#)

[WebSocketのデバッグ](#)

[関連情報](#)

はじめに

このドキュメントでは、トラブルシューティング中に基盤となるプロセスを完全に理解できるように、WebSocket接続について詳しく説明します。

前提条件

要件

このドキュメントに関する固有の要件はありません。

コンポーネント Used

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco Finesse
- Unified CCX

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

Webソケットは、クライアントとサーバ間の固定接続です。

Webソケット

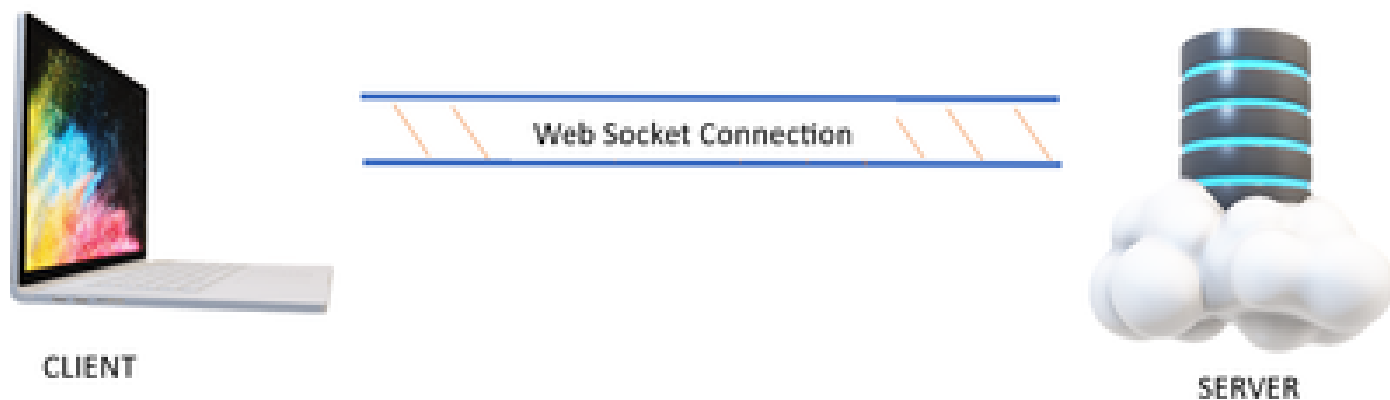
パーシステントコネクションとは何ですか。

つまり、クライアントとサーバ間の接続が確立されると、クライアントとサーバは、いつでもデータを送受信できます。

これは、双方向の全二重接続です。

サーバは、クライアント要求によってデータがプッシュバックされるのを待つ必要はありません。

同様に、クライアントは、新しいデータをサーバに送信するたびに新しい接続を作成する必要もありません。

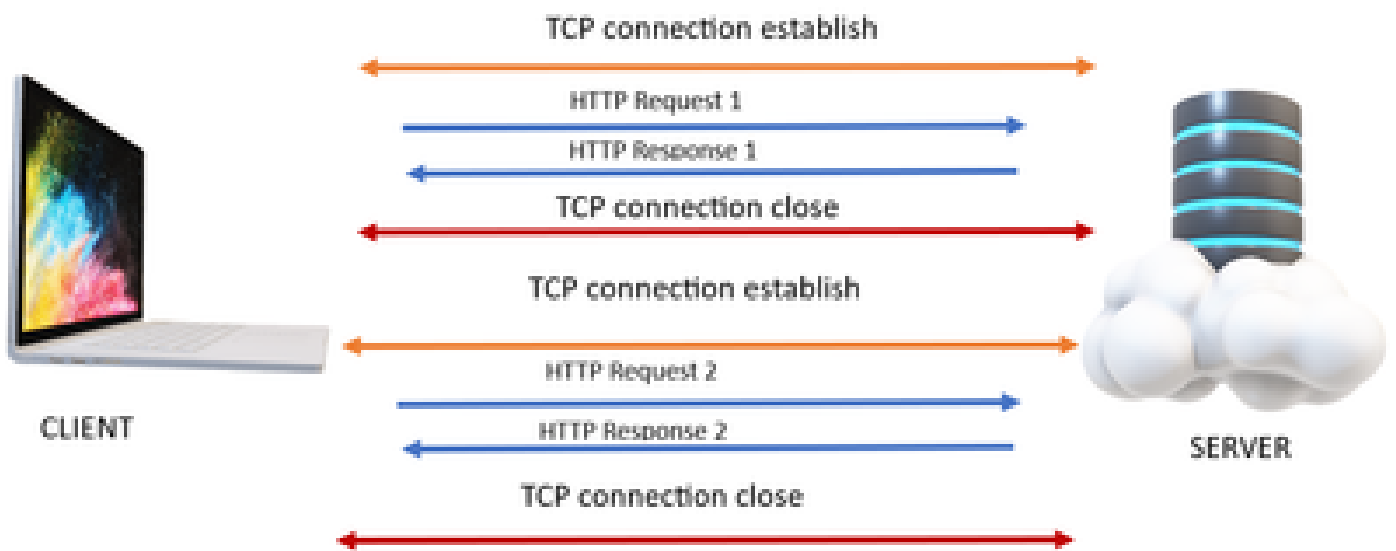


Web Socket接続は、主にリアルタイムのデータ更新が必要なアプリケーションで使用されます。

たとえば、株式取引アプリケーション、メッセージングアプリケーション、およびシスコの場合はCisco Finesseです。

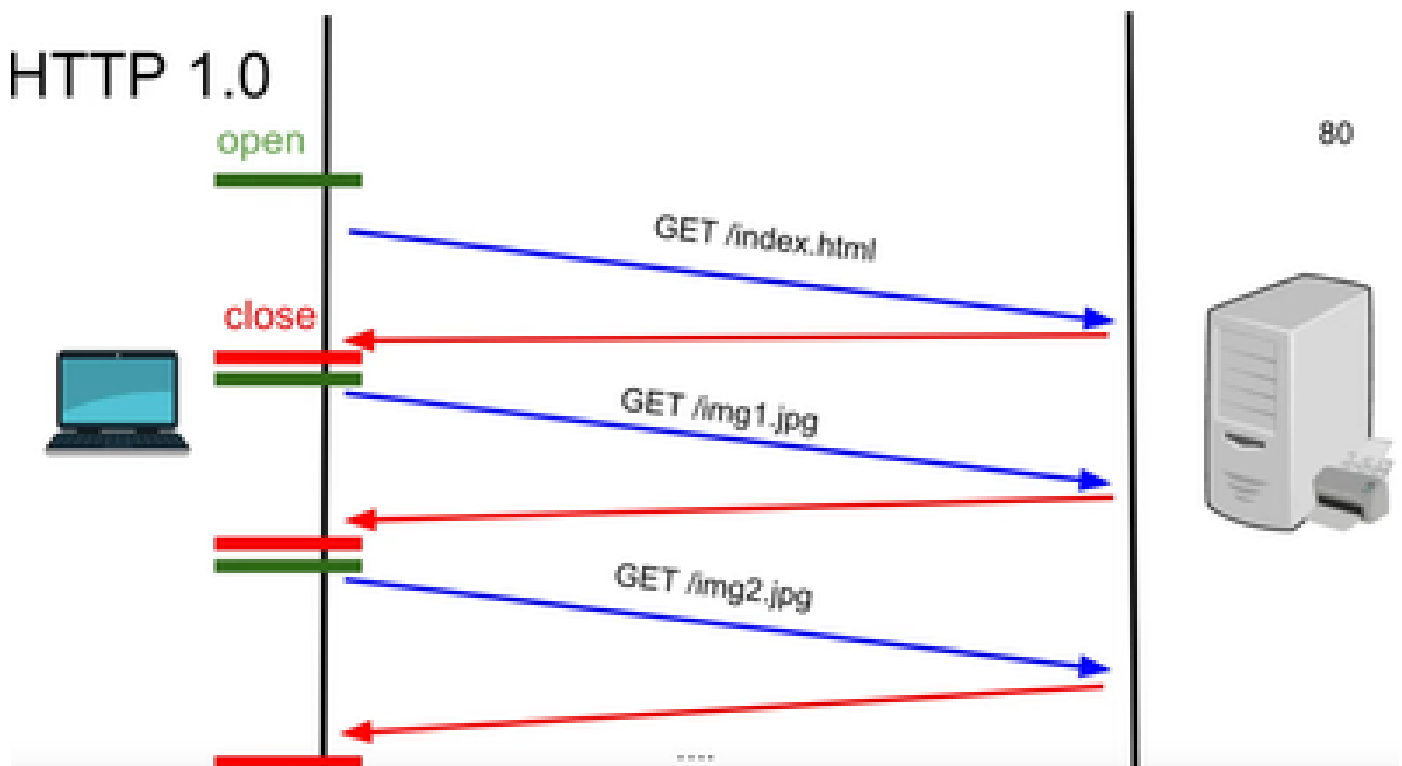
WebSocketsの仕組み

以下についての考慮が必要です。

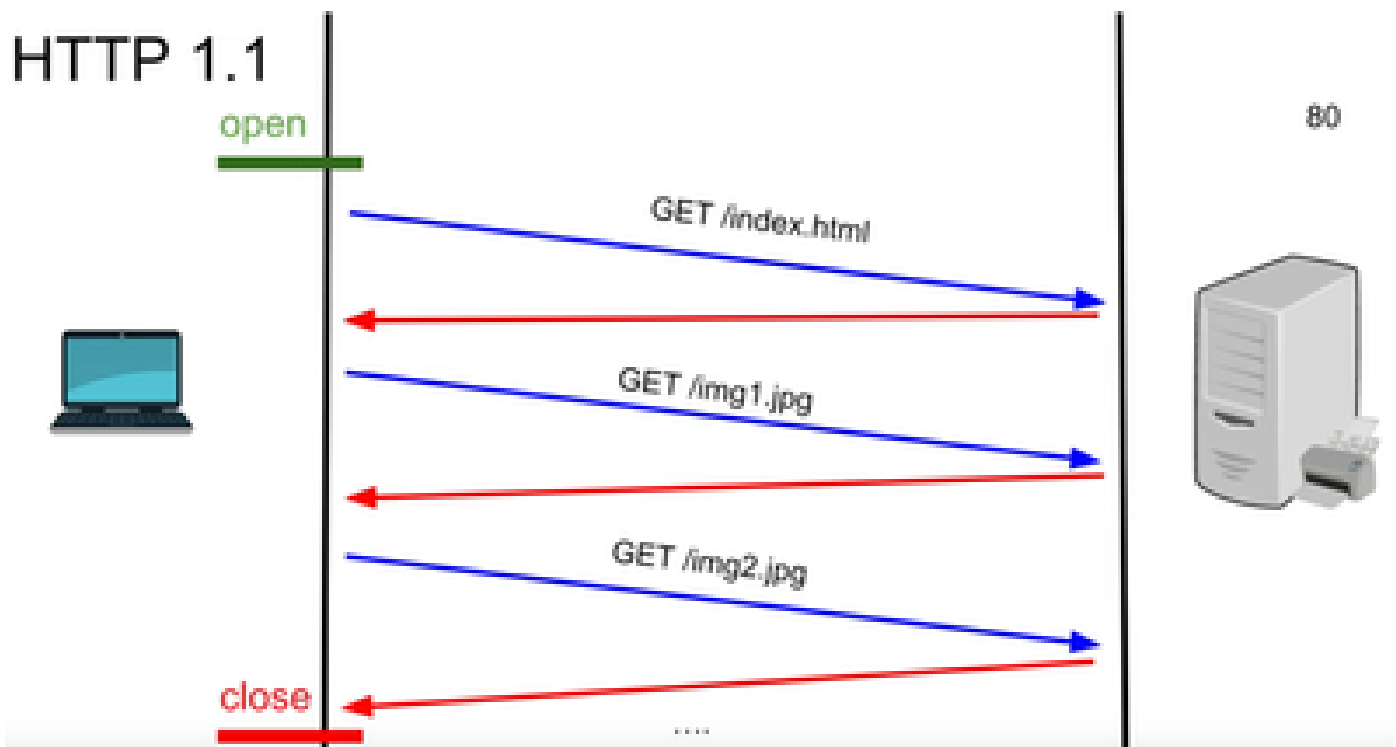


HTTP

1. TCP接続（3ウェイハンドシェイク）が行われます。
2. 次に、クライアントはHTTP要求を送信します。
3. サーバがHTTP応答を送信します。
4. 1回の要求応答サイクルの後、TCP接続が閉じます。
5. 新しいHTTP要求の場合も、TCP接続が最初に確立されます。



HTTP 1.0:要求が応答するたびに、別のHTTP要求応答に対するTCPハンドシェイクが再開されます。



HTTP 1.1 -この接続は、データを送受信してから接続を閉じることができるため、正常に動作しました。

繰り返しますが、これはリアルタイムアプリケーションには適していません。クライアントがデータを要求していない場合でも、サーバがデータを送信できるためです。したがって、このモデルは有効ではありません。

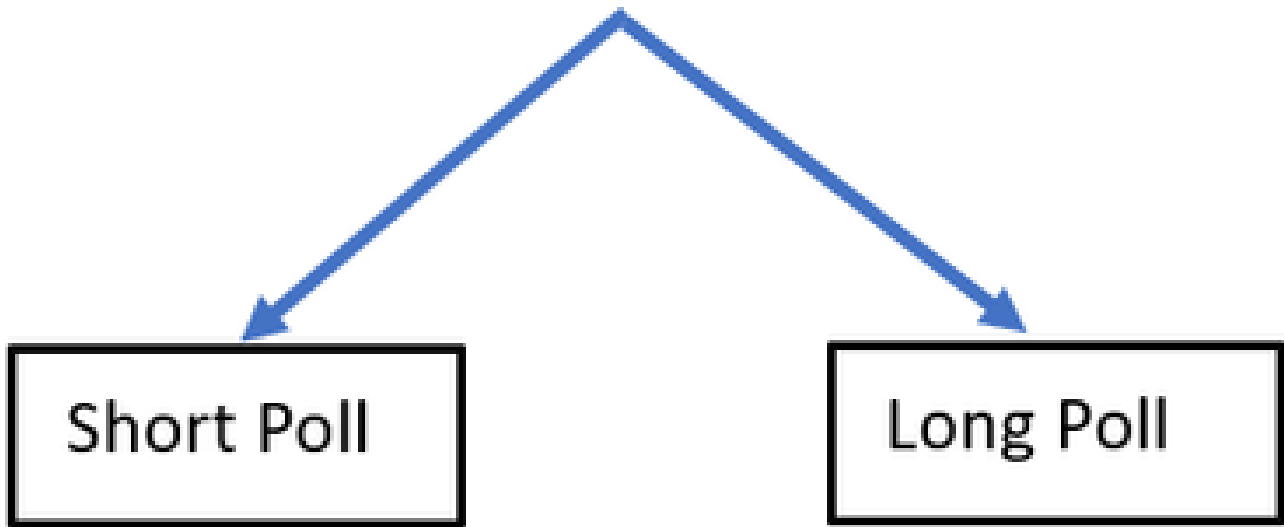
HTTPに関する問題

問題はリアルタイムシステムから始まります。

リアルタイムの更新が必要なWebサイトでは、サーバから更新を取得するたびにHTTP要求を送信することは非常に困難であり、大量の帯域幅を使用して過負荷が発生します。

これを解決するには、ポーリングと呼ばれるHTTPのメカニズムが使用されます。

POLLING



ショートポーリング：これは、要求と応答に短い固定タイマーが設定されているときに実装されます。たとえば、実装に応じて0.5秒または1秒です。

相手側からの更新がない場合は、その時間内に空の応答が得られ、リソースが浪費される可能性があります。

ロングポーリング：ショートポーリングを何らかの方法で克服しますが、応答を待機する固定時間があります。

その時間内に応答がなく、ショートポーリングよりも比較的長い間修正されている場合は、再度タイムアウトを要求します。

したがって、ポーリングはこの問題を解決する最善の方法ではありません。

このためには、使用する他の方法はSSEと呼ばれます。

SSE

サーバー送信イベント

この場合、サーバとクライアントの間に単方向接続が確立され、サーバは任意の時点でクライアントにデータを送信できます。

ここで注意する点は、これは単方向接続であり、サーバだけがクライアントにデータを送信でき、その逆は送信できないことを意味します。

使用例は次のとおりです。サーバからクライアントへの一括通知または更新。たとえば、ニュースライブの更新、Instagramライブなどがあります。

これは、リアルタイムの更新やメッセージングを伴うアプリケーションにはあまり効果的ではあ

りません。

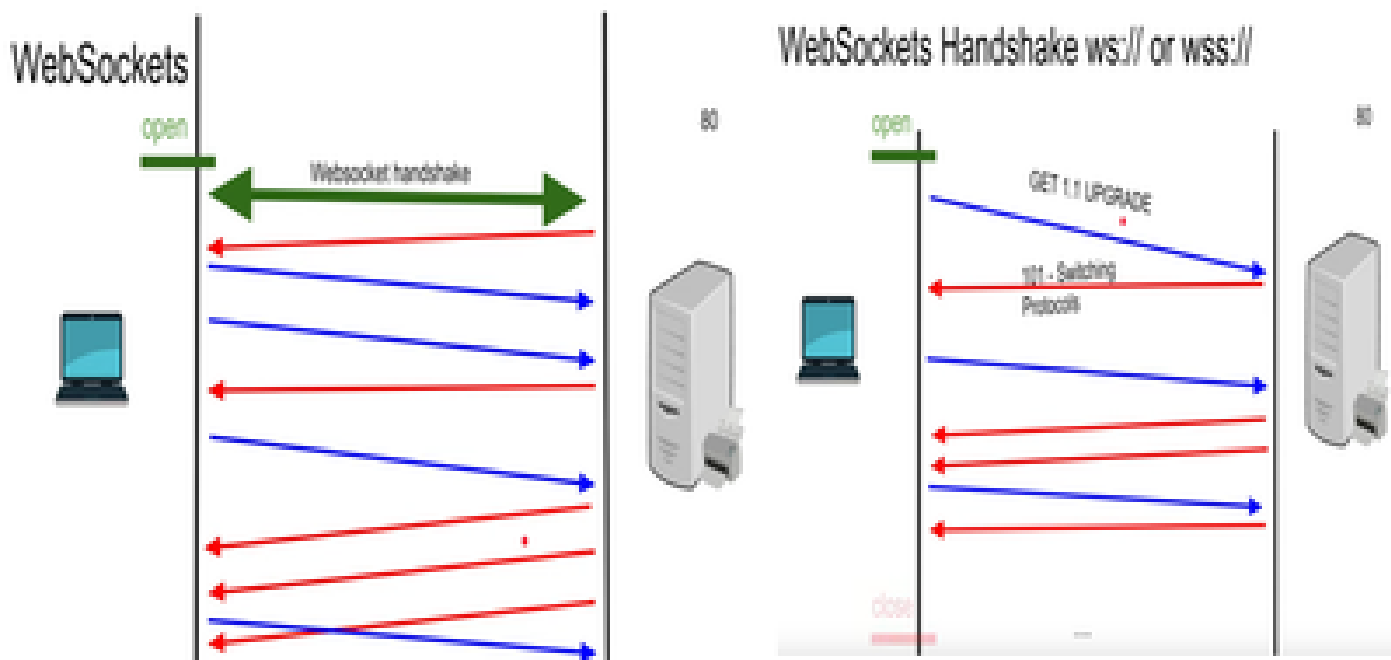
Webソケット接続は、永続的な双方向全二重接続です。

これは、サーバとクライアントの間の電話で、通話相手がいつでも通話できる状態です。

WebSocketアクション

1. WebSocket接続を確立するために、クライアントはアップグレードまたは更新されたヘッダーを含むHTTPハンドシェイク要求を送信します。
 1. これは、クライアントがサーバに対して、現在はHTTPを使用しているが、今後はWebSocket接続に移行すると言っていることを意味します。
2. 次に、サーバはHTTP 101応答で応答します。これは、sitがプロトコル応答を開始していることを意味します。
3. その後、WebSocket接続が確立されます。

これで、サーバとクライアントは同じ接続を使用して、いつでもデータを相互に転送できます。



WebSocketのデバッグ

この時点でFinesseクライアントにログインし、ネットワークのデバッグを確認すると、次のように表示されます。

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:8080/...	/ws/	WebSocket (WebSocket)	plain	333 B	0 B

メソッド- GET

ドメイン: サーバ名

ファイル:/WS/

イニシエータ- Openfire.js - websocket

要求と応答の調査：

Request

GET

スキーマ:wss

ホスト：uccxpub.prabhat.com:8445

ファイル名：/ws/

アドレス：uccxサーバのIP

ステータス:101

スイッチング プロトコル

バージョンHTTP/1.1

応答ヘッダー

接続：アップグレード

アップグレード:WebSocket

Request – open

Response

PLAIN

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

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

Request - auth

YWRtaW5pc3RyYXRvckB1Y2N4cHVlLnByYWJoYXQyY29tAGFkbWluaXN0cmF0b3IAMTIzNA==

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">`

関連情報

- [シスコのテクニカルサポートとダウンロード](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。