

WebSocket Connection voor Finesse begrijpen

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Websocket](#)

[Hoe werken WebSockets?](#)

[HTTP](#)

[Probleem met HTTP](#)

[SSE](#)

[WebSocket-acties](#)

[Webex-debuggs](#)

[Gerelateerde informatie](#)

Inleiding

Dit document beschrijft de WebSocket verbinding volledig zodat, tijdens het oplossen van problemen, de onderliggende processen grondig worden begrepen.

Voorwaarden

Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

Componenten Gebruikt

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Cisco Finesse
- UCCX

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Web Socket is een permanente verbinding tussen client en server.

Websocket

Wat wordt bedoeld met de term persistente verbinding?

Het betekent dat zodra de verbinding tussen de client en de server tot stand is gebracht, client en server op elk moment gegevens kunnen verzenden en/of ontvangen.

Dit is een bidirectionele full duplex verbinding.

De server hoeft niet te wachten op het verzoek van de client om gegevens terug te zetten.

Op dezelfde manier hoeft de client ook niet telkens een nieuwe verbinding te maken om nieuwe gegevens naar de server te verzenden.

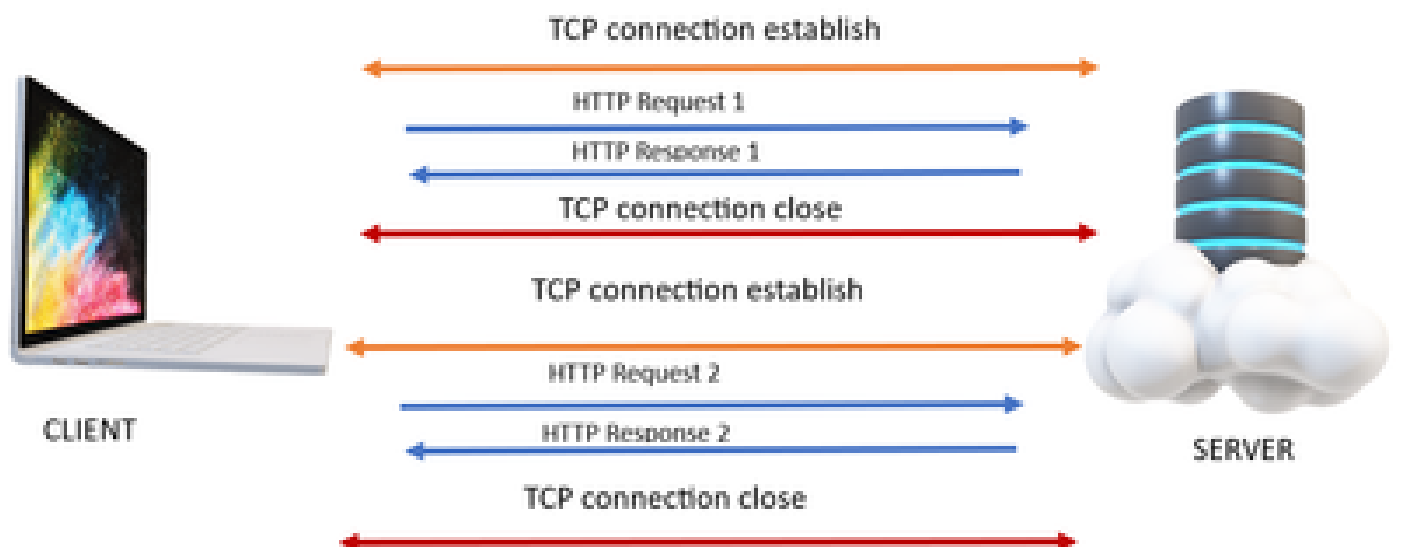


Web Socket wordt voornamelijk gebruikt in toepassingen waar real-time data updates nodig zijn.

Bijvoorbeeld, Stock Trading apps, Messaging apps, en in ons geval Cisco Finesse.

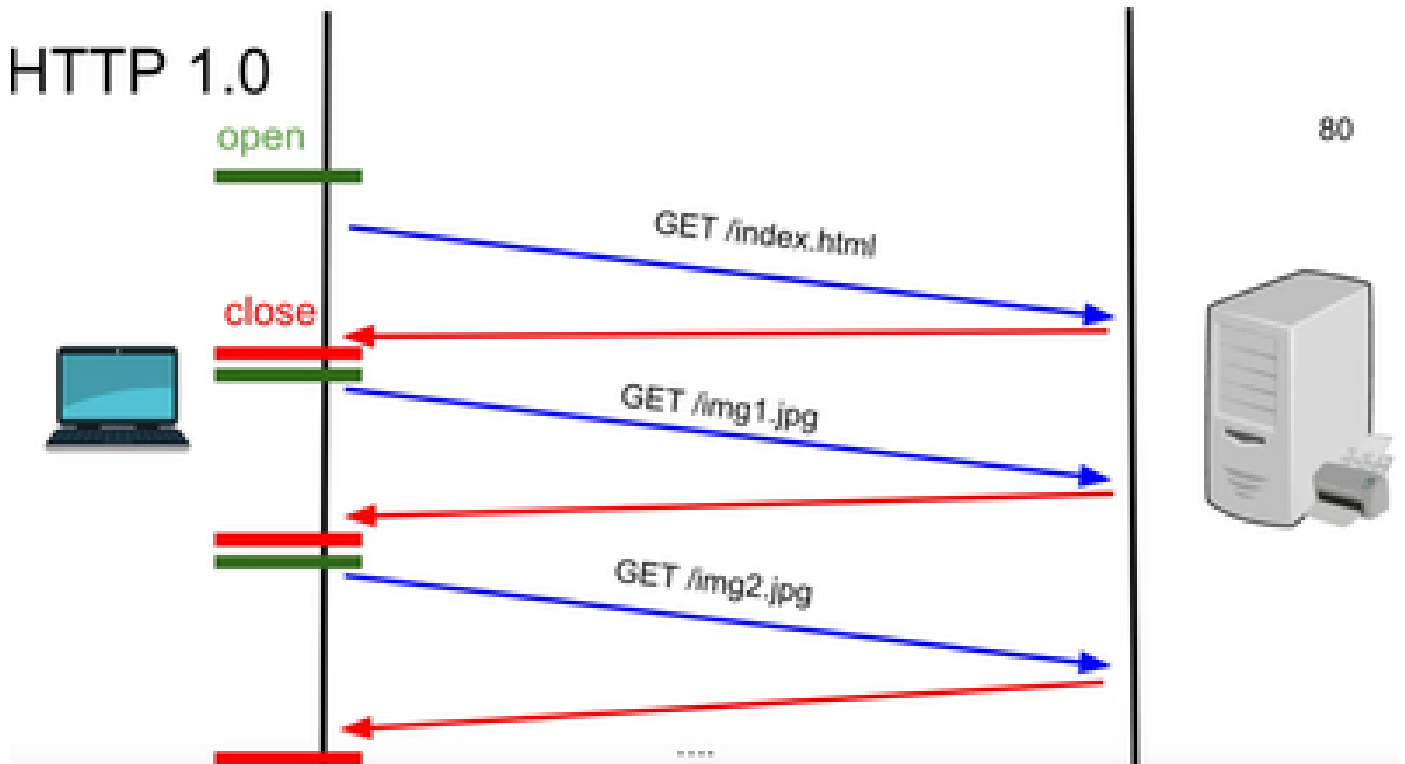
Hoe werken WebSockets?

Overweeg:

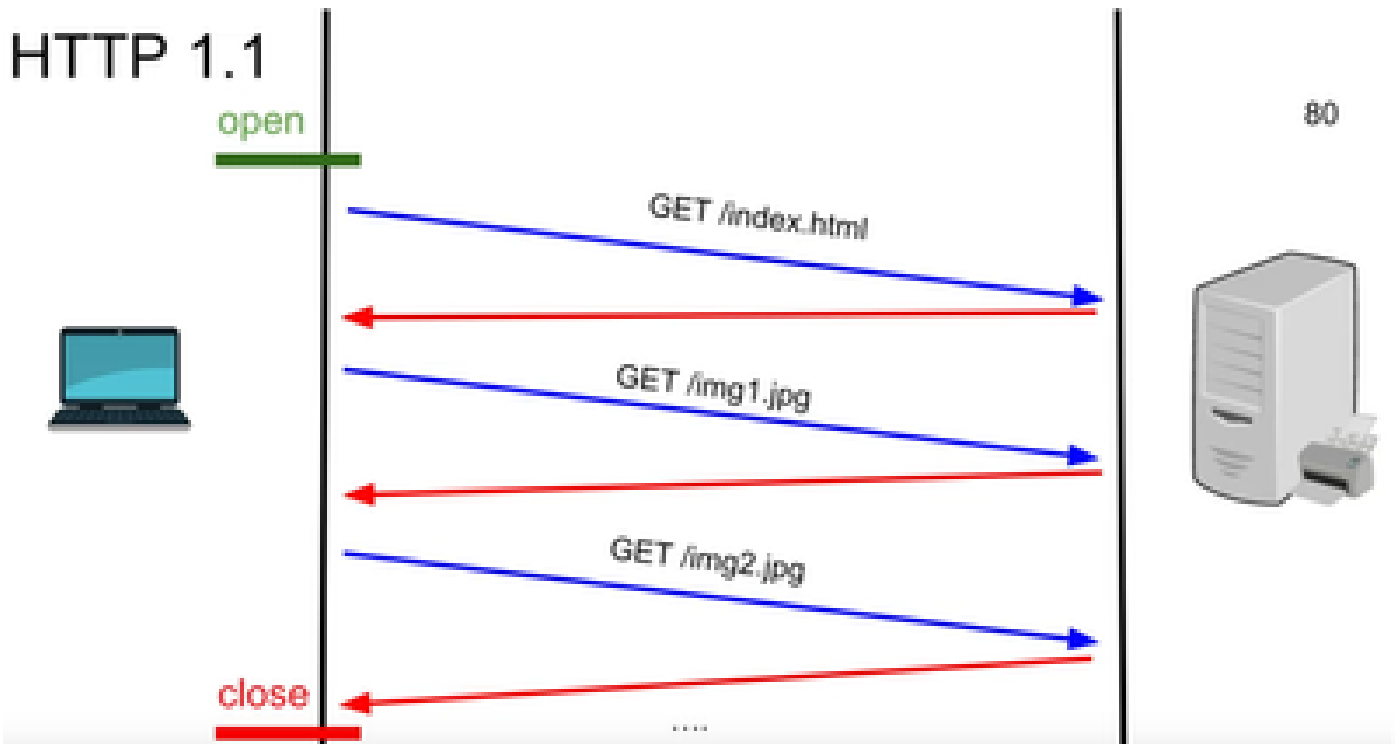


HTTP

1. De TCP-verbinding (3-voudige handdruk) vindt plaats.
2. Vervolgens stuurt de client HTTP-aanvraag.
3. Server verzendt HTTP-respons.
4. Na één reactiecyclus voor verzoek wordt de TCP-verbinding gesloten.
5. Voor een nieuw HTTP verzoek, opnieuw, wordt de verbinding van TCP eerst gevestigd.



HTTP 1.0 - Na elke reactie op het verzoek begint de TCP-handdruk opnieuw voor een andere reactie op het HTTP-verzoek.



HTTP 1.1 - Deze verbinding werkte omdat u gegevens kon verzenden en ontvangen en vervolgens de verbinding kon sluiten.

Opnieuw was dit niet geschikt voor real-time apps, omdat de server bepaalde gegevens kan versturen zelfs wanneer de client niet om het vraagt. Daarom is dit model niet effectief.

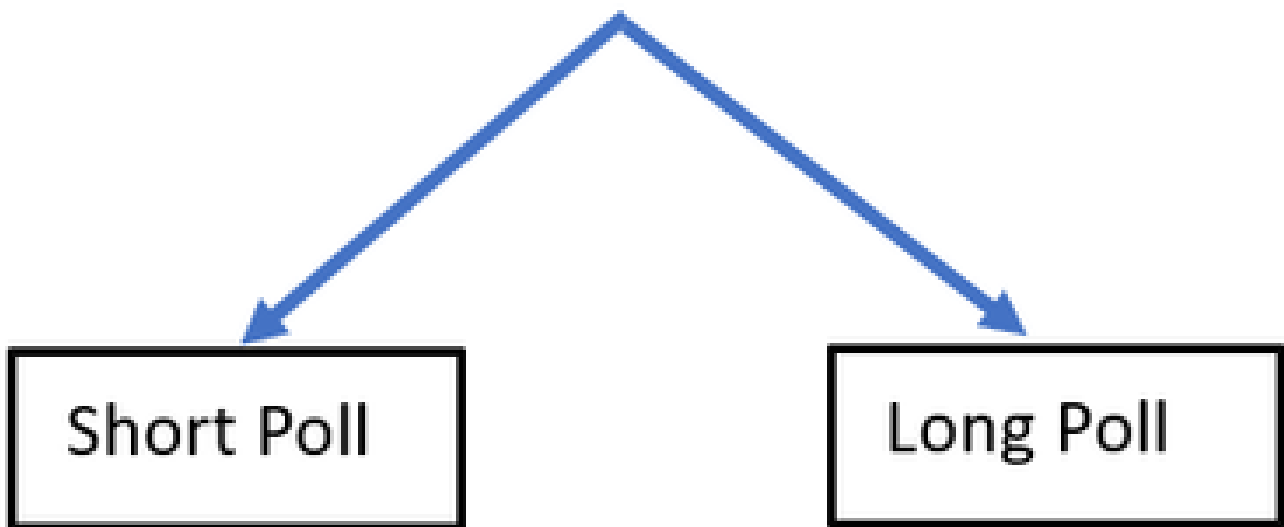
Probleem met HTTP

Het probleem begint bij de real-time systemen.

Voor een website die real-time updates vereist, is het erg moeilijk om HTTP-verzoeken elke keer te verzenden om een update van de server te krijgen en het gebruikt veel bandbreedte en veroorzaakt overbelasting.

Om dit op te lossen, wordt een mechanisme van HTTP, genaamd Polling, gebruikt.

POLLING



Korte opiniepeiling - De enquête wordt uitgevoerd wanneer er een korte vaste timer is ingesteld voor de verzoeken en antwoorden. Bijvoorbeeld, 50 seconden of 1 seconde afhankelijk van de implementatie.

Als er geen update van de andere kant is, dan kunt u lege reacties in dat tijdsbestek krijgen dat middelen kan verspillen.

Lange opiniepeiling - Op de een of andere manier komt de korte opiniepeiling te boven, maar de tijd is beperkt om op een reactie te wachten.

Als er geen reactie is in dat tijdsbestek, dat relatief langer is dan korte enquête maar nog steeds vast, dan vragen we opnieuw om onderbrekingen.

Daarom is een enquête niet de beste manier om dit probleem op te lossen.

De andere methode wordt hiervoor SSE genoemd.

SSE

Verzonden servergebeurtenissen

In dit geval is er een unidirectionele verbinding tussen de server en de client waardoor de server de gegevens naar de client op elk punt kan verzenden.

Wat hier opvalt is dat het een unidirectionele verbinding is, wat betekent dat alleen de server de gegevens naar de klant kan sturen en niet andersom.

Een voorbeeld van een use case is: bulk meldingen of updates van een server naar client. Bijvoorbeeld nieuws live updates, Instagram live, etc.

Dit is niet erg effectief voor applicaties die real-time updates en berichtenuitwisseling omvatten.

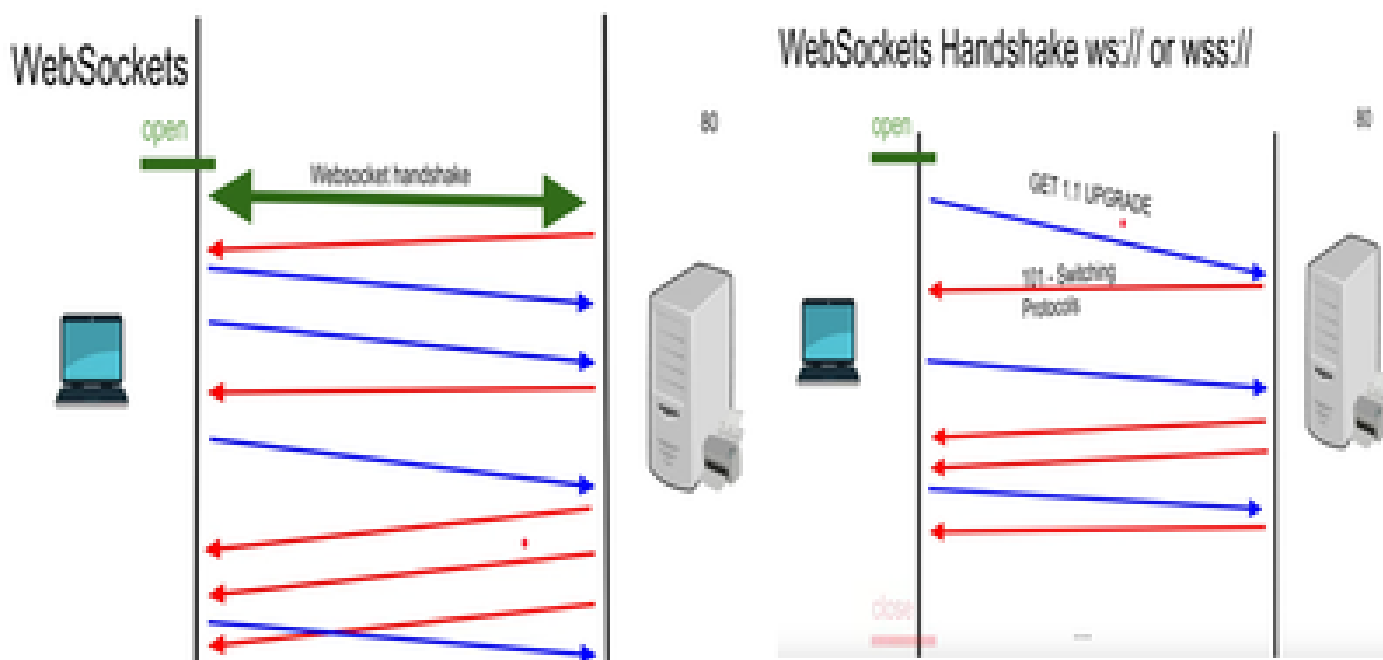
De Web Socket Connection is een hardnekkige bi-directionele full duplex verbinding.

Dit kan een telefoongesprek zijn tussen een server en een klant, waarin elke partij op elk moment met de andere partij kan praten.

WebSocket-acties

1. Om een websocket-verbinding tot stand te brengen, verstuurt de client een HTTP-handshake-verzoek met een geüpgradede of geüpdatete header.
 1. Dit betekent dat de client aan de server zegt dat dit nu over HTTP gaat, maar dat het vanaf nu naar de websocket-verbinding gaat.
2. De server reageert vervolgens met HTTP 101 response, wat betekent dat zij de protocolrespons inschakelt.
3. Hierna wordt de websocket-verbinding tot stand gebracht.

De server en client kunnen nu dezelfde verbinding gebruiken om gegevens op elk moment naar elkaar over te brengen.



Webex-debugs

Als u zich op dit punt aanmeldt bij de Finesse-client en de netwerkdebugs ziet, wordt dit weergegeven als:

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	scgpubprodha...	/ws/	@@@00c20c405262 (websocket)	plain	133 B	0 B

METHODE - GET

Domain - SERVERNAAM

BESTAND - /WS/

INITIATOR - Openfire.js - websocket

Behandeling van het verzoek en antwoord:

Aanvragen

KRIJGEN

Regeling: wss

host: uccxpub.prabhat.com:8445

bestandsnaam : /ws/

Adres: IP van de uccx-server

Status: 101

Switching Protocollen

Versie HTTP/1.1

REACTIEKOP

Verbinding: upgrade

Upgrade: WebSocket

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="NfJ3fII83zSdUDzCEICtbyrsw=">

http://jabber.org/protocol/caps" hash="sha-1" node=""
<http://www.igniterealtime.org/projects/smack>" ver="NfJ3fII83zSdUDzCEICtbyrsw=">

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">
```

Gerelateerde informatie

- [Cisco Technical Support en downloads](#)

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.