

**Version: 4.x**

On this page

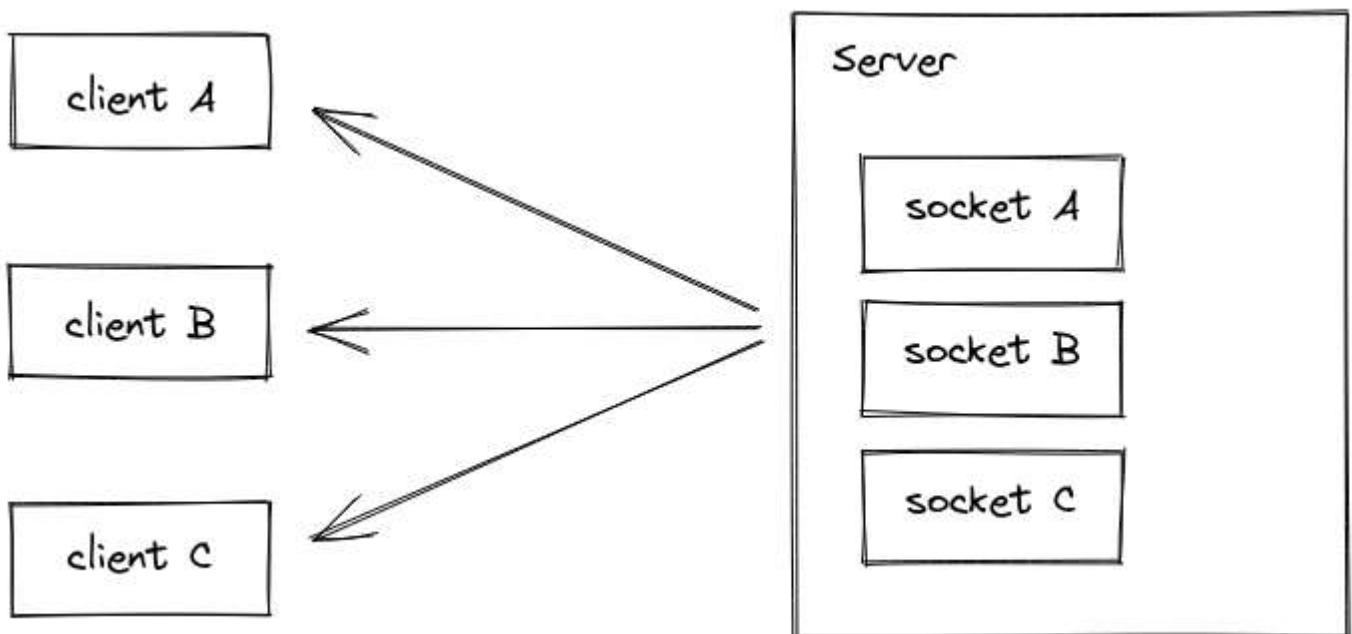
# Broadcasting events

Socket.IO makes it easy to send events to all the connected clients.

## ! INFO

Please note that broadcasting is a **server-only** feature.

## To all connected clients

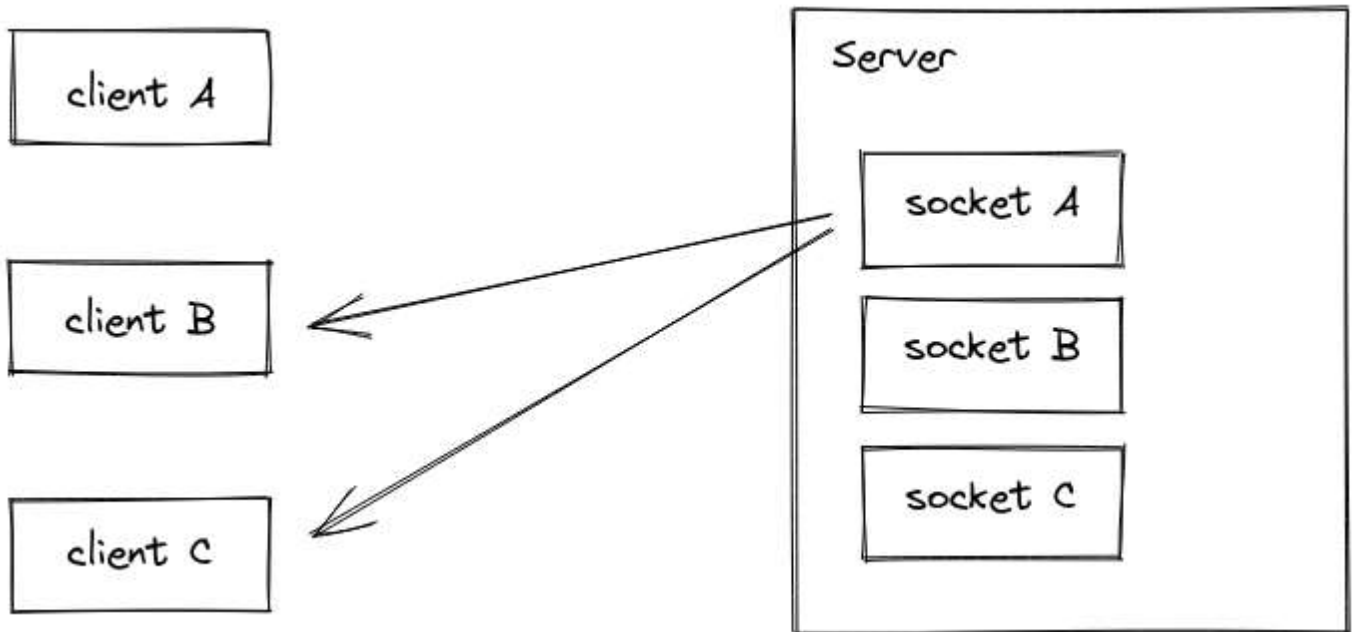


```
io.emit("hello", "world");
```

## ! CAUTION

Clients that are currently disconnected (or in the process of reconnecting) won't receive the event. Storing this event somewhere (in a database, for example) is up to you, depending on your use case.

# To all connected clients except the sender



```
io.on("connection", (socket) => {  
  socket.broadcast.emit("hello", "world");  
});
```

## **i** NOTE

In the example above, using `socket.emit("hello", "world")` (without `broadcast` flag) would send the event to "client A". You can find the list of all the ways to send an event in the [cheatsheet](#).

## With acknowledgements

Starting with Socket.IO 4.5.0, you can now broadcast an event to multiple clients and expect an acknowledgement from each one of them:

```
io.timeout(5000).emit("hello", "world", (err, responses) => {  
  if (err) {  
    // some clients did not acknowledge the event in the given delay  
  } else {  
    console.log(responses); // one response per client  
  }  
});
```

All broadcasting forms are supported:

- in a room

```
io.to("room123").timeout(5000).emit("hello", "world", (err, responses) => {  
  // ...  
});
```

- from a specific `socket`

```
socket.broadcast.timeout(5000).emit("hello", "world", (err, responses) => {  
  // ...  
});
```

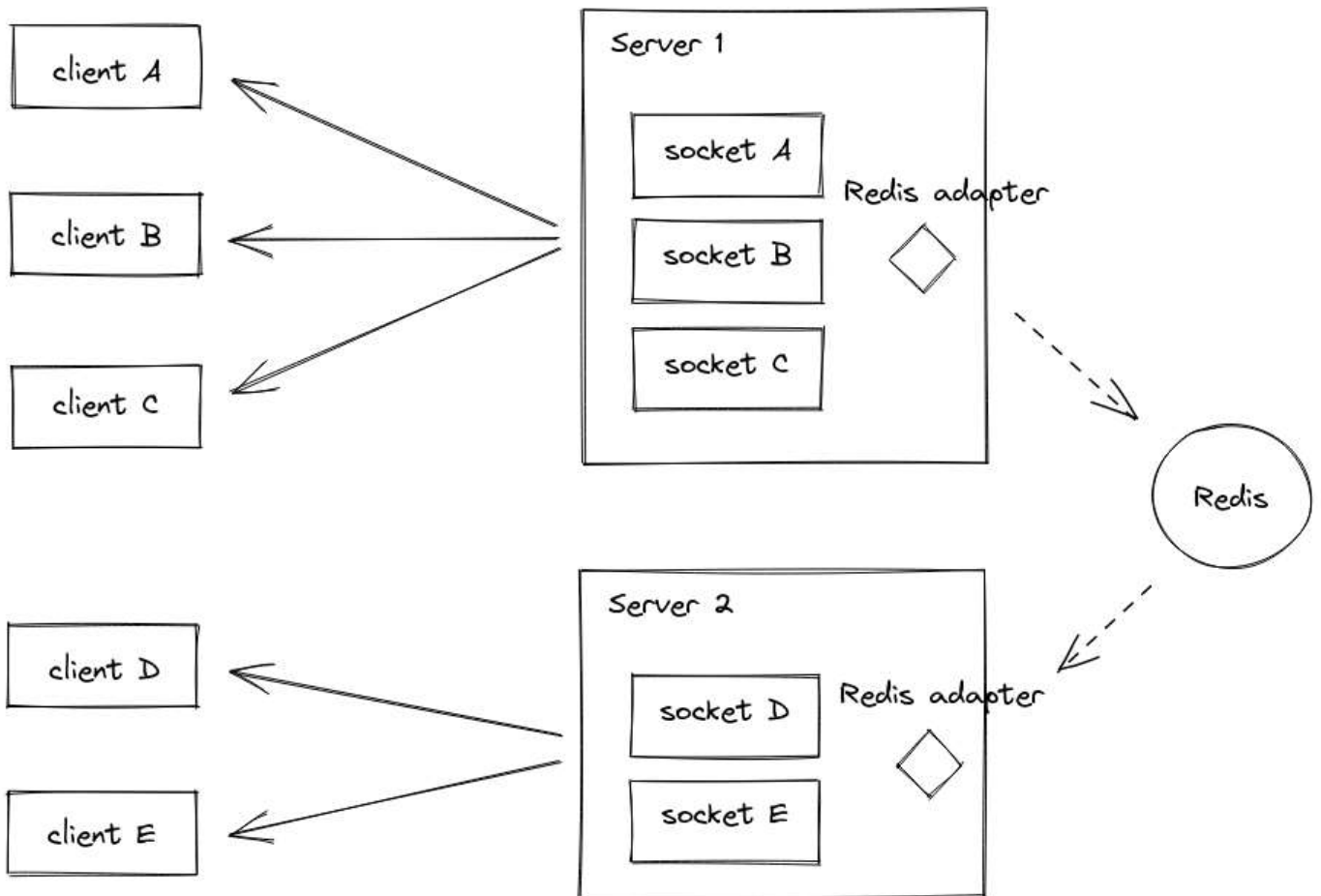
- in a namespace

```
io.of("/the-namespace").timeout(5000).emit("hello", "world", (err, responses)  
=> {  
  // ...  
});
```

## With multiple Socket.IO servers #

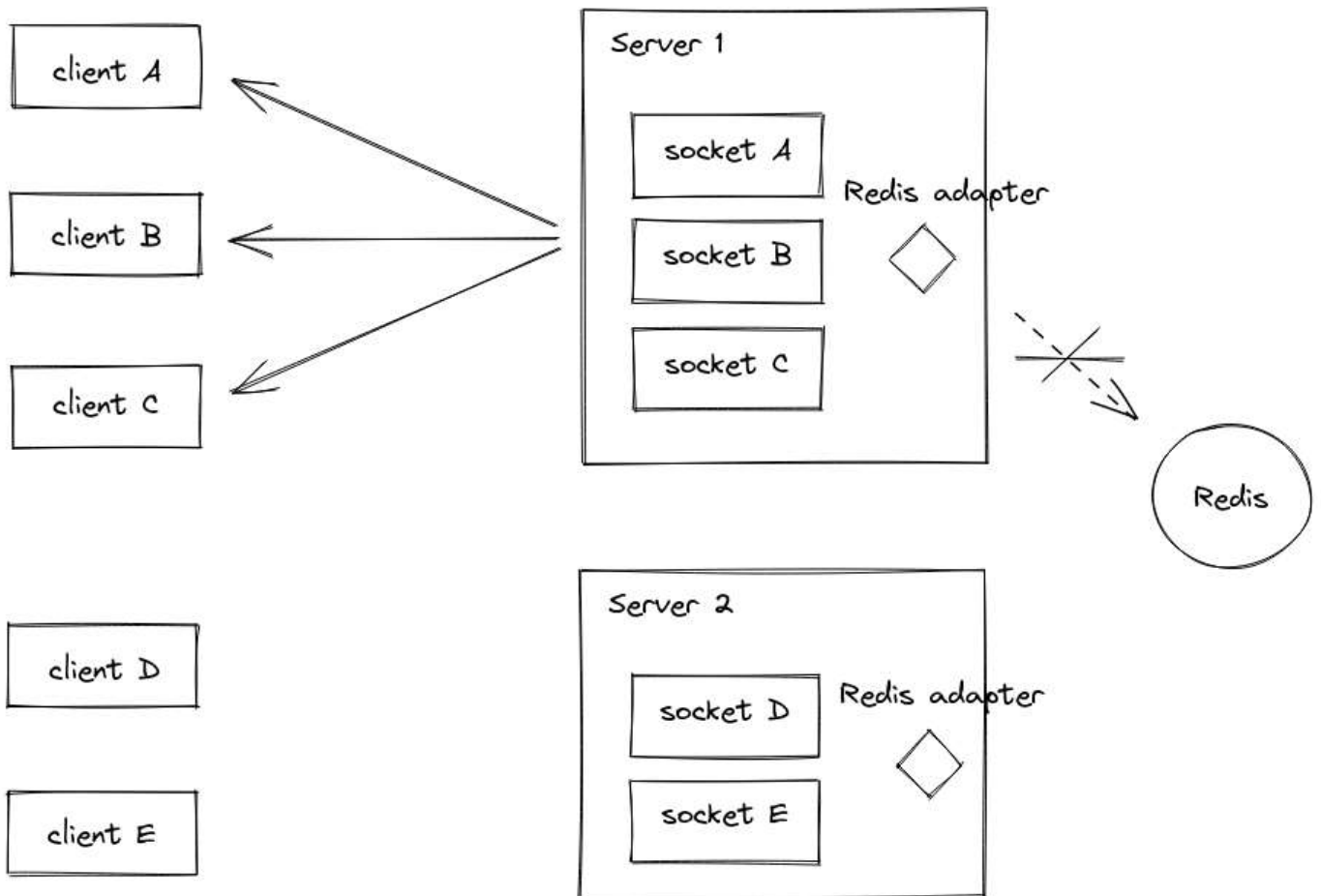
Broadcasting also works with multiple Socket.IO servers.

You just need to replace the default adapter by the [Redis Adapter](#) or another [compatible adapter](#).



In certain cases, you may want to only broadcast to clients that are connected to the current server. You can achieve this with the `local` flag:

```
io.local.emit("hello", "world");
```



In order to target specific clients when broadcasting, please see the documentation about [Rooms](#).

[✎ Edit this page](#)

Last updated on **12/19/2022**