

getConnInfo

August 18, 2026

getConnInfo

Kind: Function

Source: `src/adapters/aws-lambda/conninfo.ts`

Part of: [Adapter](#)

Get connection information from AWS Lambda

Extracts client IP from various Lambda event sources:

- API Gateway v1 (REST API): `requestContext.identity.sourceIp`
- API Gateway v2 (HTTP API/Function URLs): `requestContext.http.sourceIp`
- ALB: Falls back to `x-forwarded-for` header

`getConnInfo` reads AWS Lambda event data to determine client connection information, including the client IP address. It supports API Gateway REST API events, HTTP API or Function URL events, and ALB events through the `x-forwarded-for` header.

Signature

```
function getConnInfo(c: Context<Env>)
```

Parameters

Name	Type
<code>c</code>	<code>Context<Env></code>

Diagram

```
graph LR
  Event[AWS Lambda event] --> Context[requestContext]
  Context --> V1[identity.sourceIp<br/>API Gateway v1]
  Context --> V2[http.sourceIp<br/>API Gateway v2 / Function URL]
  Event --> Headers[headers]
  Headers --> ALB[x-forwarded-for<br/>ALB fallback]
  V1 --> ConnInfo[getConnInfo result]
  V2 --> ConnInfo
  ALB --> ConnInfo
```

Usage

```
import { getConnInfo } from './src/adapters/aws-lambda/conninfo';

export async function handler(event: unknown) {
  const connInfo = getConnInfo(event);

  console.log('Connection information:', connInfo);

  return {
    statusCode: 200,
    body: JSON.stringify({ ok: true }),
  };
}
```

AI Coding Instructions

- Pass the original Lambda event to `getConnInfo` so it can inspect `requestContext` and headers.
- Keep support for both `requestContext.identity.sourceIp` and `requestContext.http.sourceIp` when changing event handling.
- Treat `x-forwarded-for` as the fallback source for ALB events, and account for missing or differently cased headers.
- Avoid assuming every Lambda event has the same `requestContext` shape.