**heena@Clickhouse1**:**~/quickwit-v0.2.1**$ export RUST\_LOG=debug

**heena@Clickhouse1**:**~/quickwit-v0.2.1**$ ./quickwit service run searcher

2022-05-05T04:15:39.150Z INFO quickwit: *version*="0.2.1" *commit*="a857636"

2022-05-05T04:15:39.150Z DEBUG quickwit\_cli::service: run-searcher *args*=RunSearcherArgs { config\_uri: Uri { uri: "file:///home/heena/quickwit-v0.2.1/config/quickwit.yaml", protocol\_idx: 4 }, data\_dir\_path: None }

2022-05-05T04:15:39.151Z DEBUG reqwest::connect: starting new connection: https://telemetry.quickwit.io/

2022-05-05T04:15:39.151Z DEBUG hyper::client::connect::dns: resolving host="telemetry.quickwit.io"

2022-05-05T04:15:39.151Z WARN quickwit\_config::config: Seed list is empty.

2022-05-05T04:15:39.151Z INFO quickwit\_cli: Loaded Quickwit config. *config\_uri*=file:///home/heena/quickwit-v0.2.1/config/quickwit.yaml *config*=QuickwitConfig { version: 0, node\_id: "node-purple-FbZm", listen\_address: "127.0.0.1", rest\_listen\_port: 7280, peer\_seeds: [], data\_dir\_path: "./qwdata", metastore\_uri: "file:///home/heena/quickwit-v0.2.1/qwdata/indexes", default\_index\_root\_uri: "file:///home/heena/quickwit-v0.2.1/qwdata/indexes", indexer\_config: IndexerConfig { split\_store\_max\_num\_bytes: Byte(100000000000), split\_store\_max\_num\_splits: 1000 }, searcher\_config: SearcherConfig { fast\_field\_cache\_capacity: Byte(1000000000), split\_footer\_cache\_capacity: Byte(500000000), max\_num\_concurrent\_split\_streams: 100 }, storage\_config: None }

---------------------------------------------------

Connectivity checklist

✔ metastore

2022-05-05T04:15:39.152Z INFO quickwit\_cluster::cluster: Create new cluster. *node\_id*="node-purple-FbZm" *listen\_addr*=127.0.0.1:7280

2022-05-05T04:15:39.152Z DEBUG quickwit\_swim::cluster: Starting Artillery Cluster

2022-05-05T04:15:39.152Z DEBUG quickwit\_swim::state: Starting Event Loop

2022-05-05T04:15:39.153Z DEBUG quickwit\_search::search\_client\_pool: Add a new client that is connecting to the node that has been joined the cluster. *grpc\_addr*=127.0.0.1:7281

2022-05-05T04:15:39.153Z INFO quickwit\_serve: Searcher ready to accept requests at http://127.0.0.1:7280/

2022-05-05T04:15:39.153Z INFO quickwit\_serve::rest: Starting REST service. *rest\_addr*=127.0.0.1:7280

2022-05-05T04:15:39.154Z INFO **Server::run{***addr*=127.0.0.1:7280**}**: warp::server: listening on http://127.0.0.1:7280

2022-05-05T04:15:39.154Z INFO quickwit\_serve::grpc: Start gRPC service. *grpc\_addr*=127.0.0.1:7281

2022-05-05T04:15:39.246Z DEBUG hyper::client::connect::http: connecting to 34.199.112.236:443

2022-05-05T04:15:39.321Z DEBUG hyper::client::connect::http: connected to 34.199.112.236:443

2022-05-05T04:15:39.321Z DEBUG rustls::client::hs: No cached session for DnsName(DnsName(DnsName("telemetry.quickwit.io")))

2022-05-05T04:15:39.321Z DEBUG rustls::client::hs: Not resuming any session

2022-05-05T04:15:39.402Z DEBUG rustls::client::hs: ALPN protocol is Some(b"h2")

2022-05-05T04:15:39.402Z DEBUG rustls::client::hs: Using ciphersuite Tls12(Tls12CipherSuite { suite: TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256, bulk: Aes128Gcm })

2022-05-05T04:15:39.402Z DEBUG rustls::client::tls12::server\_hello: Server supports tickets

2022-05-05T04:15:39.402Z DEBUG rustls::client::tls12: ECDHE curve is ECParameters { curve\_type: NamedCurve, named\_group: secp256r1 }

2022-05-05T04:15:39.402Z DEBUG rustls::client::tls12: Server DNS name is DnsName(DnsName(DnsName("telemetry.quickwit.io")))

2022-05-05T04:15:39.477Z DEBUG rustls::client::tls12: Session saved

2022-05-05T04:15:39.478Z DEBUG h2::client: binding client connection

2022-05-05T04:15:39.478Z DEBUG h2::client: client connection bound

2022-05-05T04:15:39.478Z DEBUG h2::codec::framed\_write: send *frame*=Settings { flags: (0x0), enable\_push: 0, initial\_window\_size: 2097152, max\_frame\_size: 16384 }

2022-05-05T04:15:39.478Z DEBUG hyper::client::pool: pooling idle connection for ("https", telemetry.quickwit.io)

2022-05-05T04:15:39.478Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Settings { flags: (0x0), max\_concurrent\_streams: 128, initial\_window\_size: 65536, max\_frame\_size: 16777215 }

2022-05-05T04:15:39.478Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Settings { flags: (0x1: ACK) }

2022-05-05T04:15:39.478Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=WindowUpdate { stream\_id: StreamId(0), size\_increment: 2147418112 }

2022-05-05T04:15:39.478Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=WindowUpdate { stream\_id: StreamId(0), size\_increment: 5177345 }

2022-05-05T04:15:39.478Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(1), flags: (0x4: END\_HEADERS) }

2022-05-05T04:15:39.479Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Data { stream\_id: StreamId(1), flags: (0x1: END\_STREAM) }

2022-05-05T04:15:39.669Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Settings { flags: (0x1: ACK) }

2022-05-05T04:15:39.669Z DEBUG **Connection{***peer*=Client**}**: h2::proto::settings: received settings ACK; applying Settings { flags: (0x0), enable\_push: 0, initial\_window\_size: 2097152, max\_frame\_size: 16384 }

2022-05-05T04:15:39.671Z DEBUG **Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(1), flags: (0x5: END\_HEADERS | END\_STREAM) }

2022-05-05T04:15:39.671Z DEBUG reqwest::async\_impl::client: response '200 OK' for https://telemetry.quickwit.io/

2022-05-05T04:16:09.944Z DEBUG hyper::proto::h1::io: parsed 3 headers

2022-05-05T04:16:09.944Z DEBUG hyper::proto::h1::conn: incoming body is empty

2022-05-05T04:16:09.944Z INFO quickwit\_serve::rest: search\_stream *index\_id*=hackernews\_5 *request*=SearchStreamRequestQueryString { query: "Ambulance", search\_fields: None, start\_timestamp: None, end\_timestamp: None, fast\_field: "id", output\_format: Csv, partition\_by\_field: None }

2022-05-05T04:16:09.947Z DEBUG **root\_search\_stream{***search\_stream\_request*=SearchStreamRequest { index\_id: "hackernews\_5", query: "Ambulance", search\_fields: [], start\_timestamp: None, end\_timestamp: None, fast\_field: "id", output\_format: Csv, partition\_by\_field: None }**}**: quickwit\_search::search\_stream::root: Assigned leaf search jobs. *assigned\_leaf\_search\_jobs*=[(Grpc(127.0.0.1:7281), [SearchJob { cost: 1, offsets: SplitIdAndFooterOffsets { split\_id: "01G26NHEB10T2DX37288EKX0SJ", split\_footer\_start: 270323695, split\_footer\_end: 278910648 } }, SearchJob { cost: 1, offsets: SplitIdAndFooterOffsets { split\_id: "01G26NHMCV1BAP61AS006H7A75", split\_footer\_start: 2678183120, split\_footer\_end: 2678792526 } }, SearchJob { cost: 1, offsets: SplitIdAndFooterOffsets { split\_id: "01G26NK8YX0DM4YSVH6J9YD1GN", split\_footer\_start: 349970236, split\_footer\_end: 350048435 } }])]

2022-05-05T04:16:09.948Z DEBUG hyper::proto::h1::io: flushed 108 bytes

2022-05-05T04:16:09.948Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: hyper::client::connect::http: connecting to 127.0.0.1:7281

2022-05-05T04:16:09.948Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: hyper::client::connect::http: connected to 127.0.0.1:7281

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: h2::client: binding client connection

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: h2::client: client connection bound

2022-05-05T04:16:09.949Z DEBUG h2::codec::framed\_write: send *frame*=Settings { flags: (0x0), initial\_window\_size: 1048576, max\_frame\_size: 16384 }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: h2::codec::framed\_write: send *frame*=Settings { flags: (0x0), enable\_push: 0, initial\_window\_size: 2097152, max\_frame\_size: 16384 }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: tower::buffer::worker: *service.ready*=true processing request

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_write: send *frame*=WindowUpdate { stream\_id: StreamId(0), size\_increment: 983041 }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=WindowUpdate { stream\_id: StreamId(0), size\_increment: 5177345 }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Settings { flags: (0x0), initial\_window\_size: 1048576, max\_frame\_size: 16384 }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Settings { flags: (0x0), enable\_push: 0, initial\_window\_size: 2097152, max\_frame\_size: 16384 }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_write: send *frame*=Settings { flags: (0x1: ACK) }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Settings { flags: (0x1: ACK) }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=WindowUpdate { stream\_id: StreamId(0), size\_increment: 983041 }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=WindowUpdate { stream\_id: StreamId(0), size\_increment: 5177345 }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Settings { flags: (0x1: ACK) }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::proto::settings: received settings ACK; applying Settings { flags: (0x0), initial\_window\_size: 1048576, max\_frame\_size: 16384 }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Settings { flags: (0x1: ACK) }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::proto::settings: received settings ACK; applying Settings { flags: (0x0), enable\_push: 0, initial\_window\_size: 2097152, max\_frame\_size: 16384 }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(1), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Data { stream\_id: StreamId(1) }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(1), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.949Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Data { stream\_id: StreamId(1), flags: (0x1: END\_STREAM) }

2022-05-05T04:16:09.949Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Data { stream\_id: StreamId(1) }

2022-05-05T04:16:09.950Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Data { stream\_id: StreamId(1), flags: (0x1: END\_STREAM) }

2022-05-05T04:16:09.950Z INFO **search\_adapter:leaf\_search\_stream**: quickwit\_search::service: leaf\_search *index*="hackernews\_5" *splits*=[SplitIdAndFooterOffsets { split\_id: "01G26NHEB10T2DX37288EKX0SJ", split\_footer\_start: 270323695, split\_footer\_end: 278910648 }, SplitIdAndFooterOffsets { split\_id: "01G26NHMCV1BAP61AS006H7A75", split\_footer\_start: 2678183120, split\_footer\_end: 2678792526 }, SplitIdAndFooterOffsets { split\_id: "01G26NK8YX0DM4YSVH6J9YD1GN", split\_footer\_start: 349970236, split\_footer\_end: 350048435 }]

2022-05-05T04:16:09.950Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(1), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.950Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(1), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.952Z ERROR **search\_adapter:leaf\_search\_stream**:**leaf\_search\_stream**:**leaf\_search\_stream\_single\_split{***split\_id*=01G26NK8YX0DM4YSVH6J9YD1GN**}**:**warmup**:**warm\_up\_terms**: quickwit\_directories::storage\_directory: *path*="6dc68fd1122c44a985ccf5348907c5f8.term" *msg*="Unsupported operation. StorageDirectory only supports async reads"

2022-05-05T04:16:09.956Z ERROR **search\_adapter:leaf\_search\_stream**:**leaf\_search\_stream**:**leaf\_search\_stream\_single\_split{***split\_id*=01G26NHMCV1BAP61AS006H7A75**}**:**warmup**:**warm\_up\_terms**: quickwit\_directories::storage\_directory: *path*="ccf34dbac4614904b1124b751756dab8.term" *msg*="Unsupported operation. StorageDirectory only supports async reads"

2022-05-05T04:16:09.957Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(1), flags: (0x5: END\_HEADERS | END\_STREAM) }

2022-05-05T04:16:09.958Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(1), flags: (0x5: END\_HEADERS | END\_STREAM) }

2022-05-05T04:16:09.958Z DEBUG quickwit\_search::cluster\_client: Leaf search stream response error. Retry once to execute LeafSearchStreamRequest { request: Some(SearchStreamRequest { index\_id: "hackernews\_5", query: "Ambulance", search\_fields: [], start\_timestamp: None, end\_timestamp: None, fast\_field: "id", output\_format: Csv, partition\_by\_field: None }), split\_offsets: [SplitIdAndFooterOffsets { split\_id: "01G26NHEB10T2DX37288EKX0SJ", split\_footer\_start: 270323695, split\_footer\_end: 278910648 }, SplitIdAndFooterOffsets { split\_id: "01G26NHMCV1BAP61AS006H7A75", split\_footer\_start: 2678183120, split\_footer\_end: 2678792526 }, SplitIdAndFooterOffsets { split\_id: "01G26NK8YX0DM4YSVH6J9YD1GN", split\_footer\_start: 349970236, split\_footer\_end: 350048435 }], doc\_mapper: "{\"type\":\"default\",\"store\_source\":false,\"default\_search\_fields\":[\"title\",\"by\",\"text\",\"url\"],\"timestamp\_field\":\"time\",\"field\_mappings\":[{\"name\":\"id\",\"type\":\"u64\",\"stored\":true,\"fast\":true,\"indexed\":true},{\"name\":\"type\",\"type\":\"text\",\"stored\":true,\"fast\":false,\"tokenizer\":\"raw\",\"record\":\"basic\"},{\"name\":\"by\",\"type\":\"text\",\"stored\":true,\"fast\":false,\"tokenizer\":\"default\",\"record\":\"position\"},{\"name\":\"time\",\"type\":\"i64\",\"stored\":true,\"fast\":true,\"indexed\":true},{\"name\":\"text\",\"type\":\"text\",\"stored\":true,\"fast\":false,\"tokenizer\":\"raw\",\"record\":\"position\"},{\"name\":\"kids\",\"type\":\"array<i64>\",\"stored\":true,\"fast\":true,\"indexed\":true},{\"name\":\"url\",\"type\":\"text\",\"stored\":true,\"fast\":false,\"tokenizer\":\"default\",\"record\":\"position\"},{\"name\":\"score\",\"type\":\"i64\",\"stored\":true,\"fast\":true,\"indexed\":true},{\"name\":\"title\",\"type\":\"text\",\"stored\":true,\"fast\":false,\"tokenizer\":\"default\",\"record\":\"position\"}],\"tag\_fields\":[]}", index\_uri: "file:///home/heena/quickwit-v0.2.1/qwdata/indexes/hackernews\_5" } with Grpc(127.0.0.1:7281)

2022-05-05T04:16:09.958Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**: tower::buffer::worker: *service.ready*=true processing request

2022-05-05T04:16:09.958Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(3), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.958Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Data { stream\_id: StreamId(3) }

2022-05-05T04:16:09.958Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_write: send *frame*=Data { stream\_id: StreamId(3), flags: (0x1: END\_STREAM) }

2022-05-05T04:16:09.958Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(3), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.958Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Data { stream\_id: StreamId(3) }

2022-05-05T04:16:09.958Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_read: received *frame*=Data { stream\_id: StreamId(3), flags: (0x1: END\_STREAM) }

2022-05-05T04:16:09.958Z INFO **search\_adapter:leaf\_search\_stream**: quickwit\_search::service: leaf\_search *index*="hackernews\_5" *splits*=[SplitIdAndFooterOffsets { split\_id: "01G26NHEB10T2DX37288EKX0SJ", split\_footer\_start: 270323695, split\_footer\_end: 278910648 }, SplitIdAndFooterOffsets { split\_id: "01G26NHMCV1BAP61AS006H7A75", split\_footer\_start: 2678183120, split\_footer\_end: 2678792526 }, SplitIdAndFooterOffsets { split\_id: "01G26NK8YX0DM4YSVH6J9YD1GN", split\_footer\_start: 349970236, split\_footer\_end: 350048435 }]

2022-05-05T04:16:09.958Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(3), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.959Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(3), flags: (0x4: END\_HEADERS) }

2022-05-05T04:16:09.959Z ERROR **search\_adapter:leaf\_search\_stream**:**leaf\_search\_stream**:**leaf\_search\_stream\_single\_split{***split\_id*=01G26NHMCV1BAP61AS006H7A75**}**:**warmup**:**warm\_up\_terms**: quickwit\_directories::storage\_directory: *path*="ccf34dbac4614904b1124b751756dab8.term" *msg*="Unsupported operation. StorageDirectory only supports async reads"

2022-05-05T04:16:09.959Z ERROR **search\_adapter:leaf\_search\_stream**:**leaf\_search\_stream**:**leaf\_search\_stream\_single\_split{***split\_id*=01G26NK8YX0DM4YSVH6J9YD1GN**}**:**warmup**:**warm\_up\_terms**: quickwit\_directories::storage\_directory: *path*="6dc68fd1122c44a985ccf5348907c5f8.term" *msg*="Unsupported operation. StorageDirectory only supports async reads"

2022-05-05T04:16:09.960Z DEBUG **Connection{***peer*=Server**}**: h2::codec::framed\_write: send *frame*=Headers { stream\_id: StreamId(3), flags: (0x5: END\_HEADERS | END\_STREAM) }

2022-05-05T04:16:09.960Z DEBUG **client:leaf\_search\_stream{***grpc\_addr*=127.0.0.1:7281**}**:**Connection{***peer*=Client**}**: h2::codec::framed\_read: received *frame*=Headers { stream\_id: StreamId(3), flags: (0x5: END\_HEADERS | END\_STREAM) }

2022-05-05T04:16:09.960Z ERROR quickwit\_serve::rest: Error when streaming search results. *error*=Internal error: `Internal error: `An IO error occurred: 'Unsupported operation. StorageDirectory only supports async reads: "6dc68fd1122c44a985ccf5348907c5f8.term"'`.`.

2022-05-05T04:16:09.960Z DEBUG hyper::server::conn::spawn\_all: connection error: error from user's HttpBody stream: body write aborted

2022-05-05T04:16:09.992Z ERROR **search\_adapter:leaf\_search\_stream**:**leaf\_search\_stream**: quickwit\_search::search\_stream::leaf: Failed to send leaf search stream result. Stop sending. Cause: channel closed

2022-05-05T04:16:09.993Z ERROR **search\_adapter:leaf\_search\_stream**:**leaf\_search\_stream**: quickwit\_search::search\_stream::leaf: Failed to send leaf search stream result. Stop sending. Cause: channel closed

^C

**heena@Clickhouse1**:**~/quickwit-v0.2.1**$