Microsoft Defender

Microsoft Defender → Coralogix Via Azure Event Hubs

At a glance (architecture)

- Microsoft Defender (Defender for Cloud and/or Defender XDR / Endpoint)
- → **Azure Event Hubs** (central streaming bus)
- → **Azure Function (Event Hub trigger)** using Coralogix template
- → **Coralogix** (logs/alerts analytics)

Prerequisites

- Azure subscription with permission to:
  - Create & configure **Event Hubs** namespace/event hub
  - Configure **Defender** export (Defender for Cloud and/or Defender XDR)
  - Deploy **Azure Functions** from template

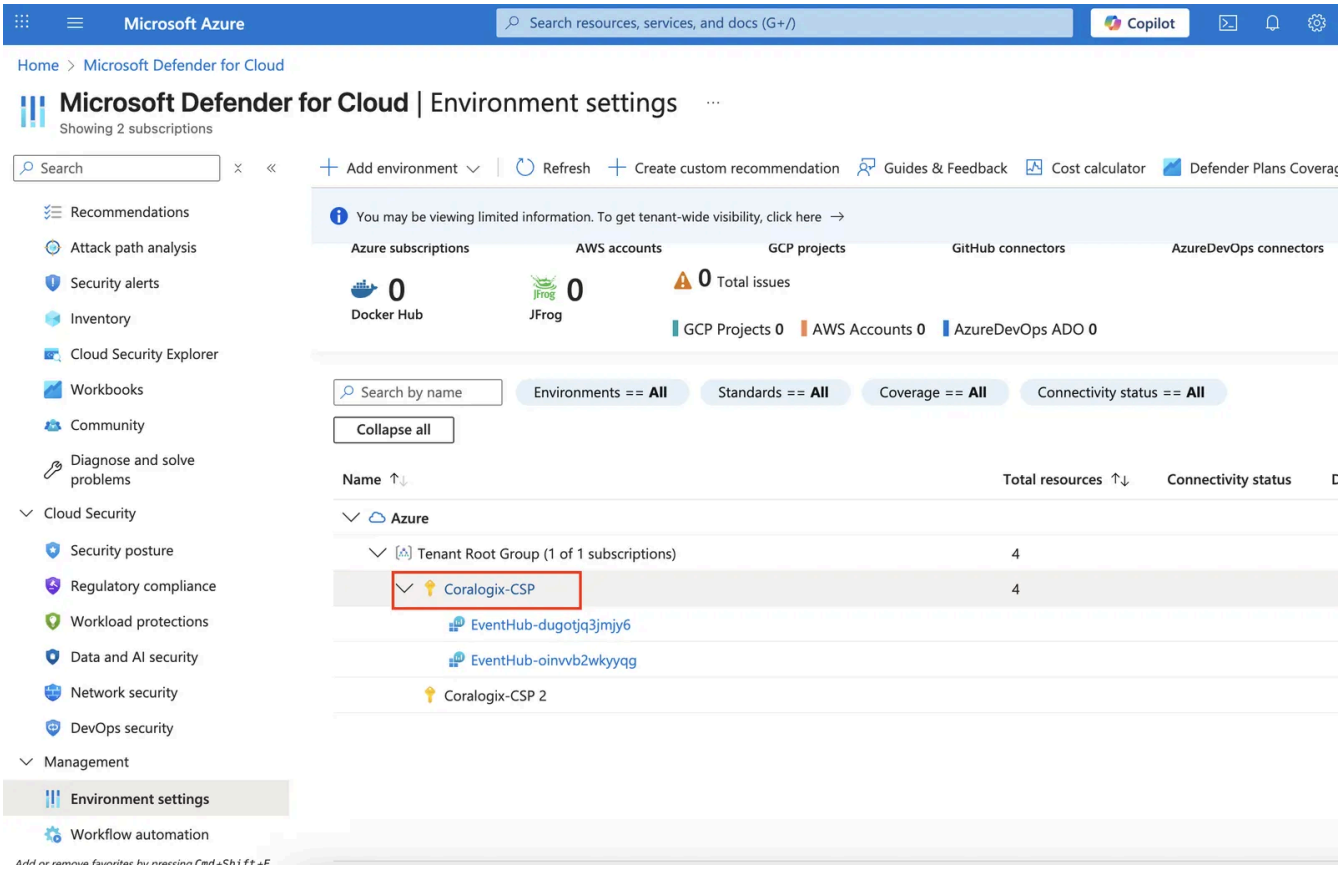
Step 1 — Create Event Hubs namespace & event hub

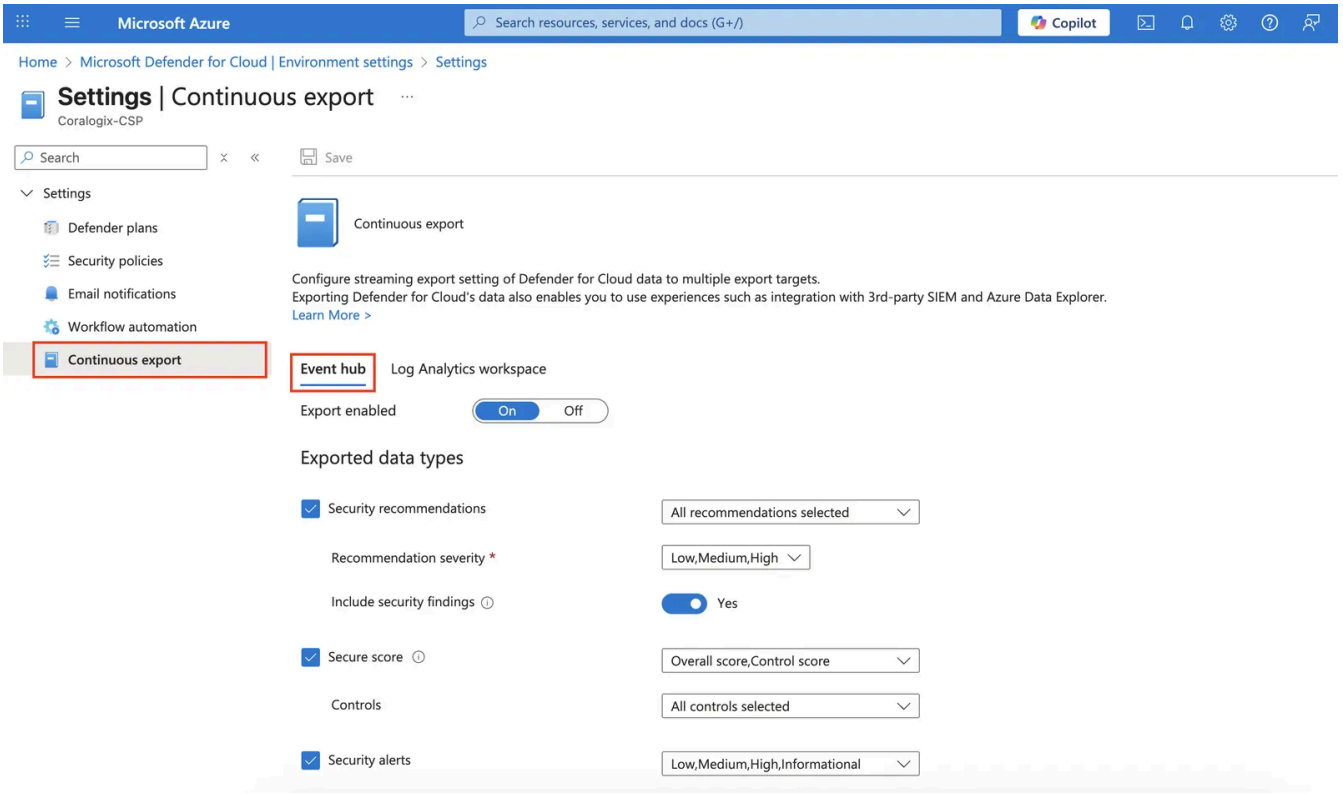
1. In Azure Portal → **Event Hubs** → **+ Create** namespace.
   - Pricing tier: **Standard** (recommended)
   - Throughput units: start with **1** (scale later)
   - Networking: public or private as per policy
2. Inside the namespace → **+ Event Hub**
   - Name: `defender-stream`
   - Partitions: **2–4** (scale with volume)
   - Message retention: 1–7 days (align to recovery needs)
3. Create a **Shared access policy** with **Listen** rights (for the Function) and copy the connection string.

Step 2 — Export from Defender for Cloud → Event Hubs

Use **Continuous export** to stream alerts and recommendations.

1. Azure Portal → **Defender for Cloud** → *Environment settings* → choose subscription.





Continuous export → + Add export

1. **Export target**: **Azure Event Hubs**
2. **Data types**: select **Security alerts** and/or **Security recommendations** (add resource state changes if needed)
3. **Event Hub**: choose your namespace and the `defender-stream` hub
4. Save.

Resulting payloads: JSON per alert/recommendation with keys such as `alertId`, `severity`, `resourceId`, `description`, `compromisedEntity`, `tactics`, `firstSeen`, `lastSeen`.

Step 3 — Deploy the Coralogix Azure Function (Event Hub → Coralogix)

We'll use Coralogix's ARM template to create a Function App that triggers on your Event Hub and forwards events to Coralogix.

1. Open the Coralogix **Azure Functions** deployment link (ARM template).

Learn more about Coralogix ARM in our documentation.

1. Fill parameters:
   - **EventHub connection string** (with `Endpoint=...;EntityPath=defender-stream` or supply hub name separately if template asks)
   - **Consumer group**: `$Default` or a dedicated `coralogix-cg`
   - **CORALOGIX_PRIVATE_KEY**: from Coralogix UI → Data Ingestion
   - **CORALOGIX_REGION**: e.g., `EU1`, `AP1`, `US1` (exact value per account)
   - **APPLICATION_NAME**: e.g., `microsoft-defender`
   - **SUBSYSTEM**: e.g., `alerts` or `xdr-events`
   - (optional) **COMPANY_ID** if required by your region
2. **Plan**: Consumption or Premium (P1 for high throughput). Enable **Always On** if using Premium.
3. **Networking**: if Event Hub is private, integrate the Function with the VNet and grant access.
4. Deploy. Confirm the Function is bound to the Event Hub trigger.

Step 4 — Validate ingestion in Coralogix

- In Coralogix → **Logs** → filter by `application:microsoft-defender`.
- You should see JSON logs arriving within a minute of new alerts/events.
- Example search (Lucene):
  - `severity:High AND (tactics:CredentialAccess OR tactics:DefenseEvasion)`

Support

Need help?

Our world-class customer success team is available 24/7 to walk you through your setup and answer any questions that may come up.

Feel free to reach out to us **via our in-app chat** or by sending us an email to support@coralogix.com.

Last updated: October 9, 2025

Was this helpful?

Leave your feedback here.

◯ **Yes** ◯ **No**

Send