

AMAZON EVENTBRIDGE

BUILD EVENT-DRIVEN APPLICATIONS AT SCALE



INTRODUCTION 🙌

EventBridge makes it easier to build Event-Driven architectures by integrating AWS services without code.

The general idea of EventBridge is that your event bus receives an event from an event producer. You can add rules to your event bus. The rule routes the incoming event to the defined consumer, e.g. Lambda, SQS, Step Functions, and many more.

SOURCES & TARGETS ⚡

Your **source** string describes where an event is coming from. For internal AWS events, this always starts with `aws..`.

Targets are the consumers of our events. There are many AWS services supported to integrate with. Among them are **API Destinations**, **API Gateway**, **CloudWatch**, **EventBus**, **Lambda**, **Kinesis**, **SQS**, or **SNS**.

SCHEMA BINDINGS 📄

When working with events we recommend sharing a schema of your events.

A schema is a description of your event properties. It describes the structure of your event. Your customers who use EventBridge also benefit from these schemas a lot. They can see which attributes they need to send to the event bus. Or if they want to create a consumer they know which attributes are available.

Schemas are available in both OpenAPI 3 and JSONSchema Draft 4 format. Schemas are versioned. Each time the schema changes you can create a new version.

EVENT-DRIVEN ARCHITECTURES 📈

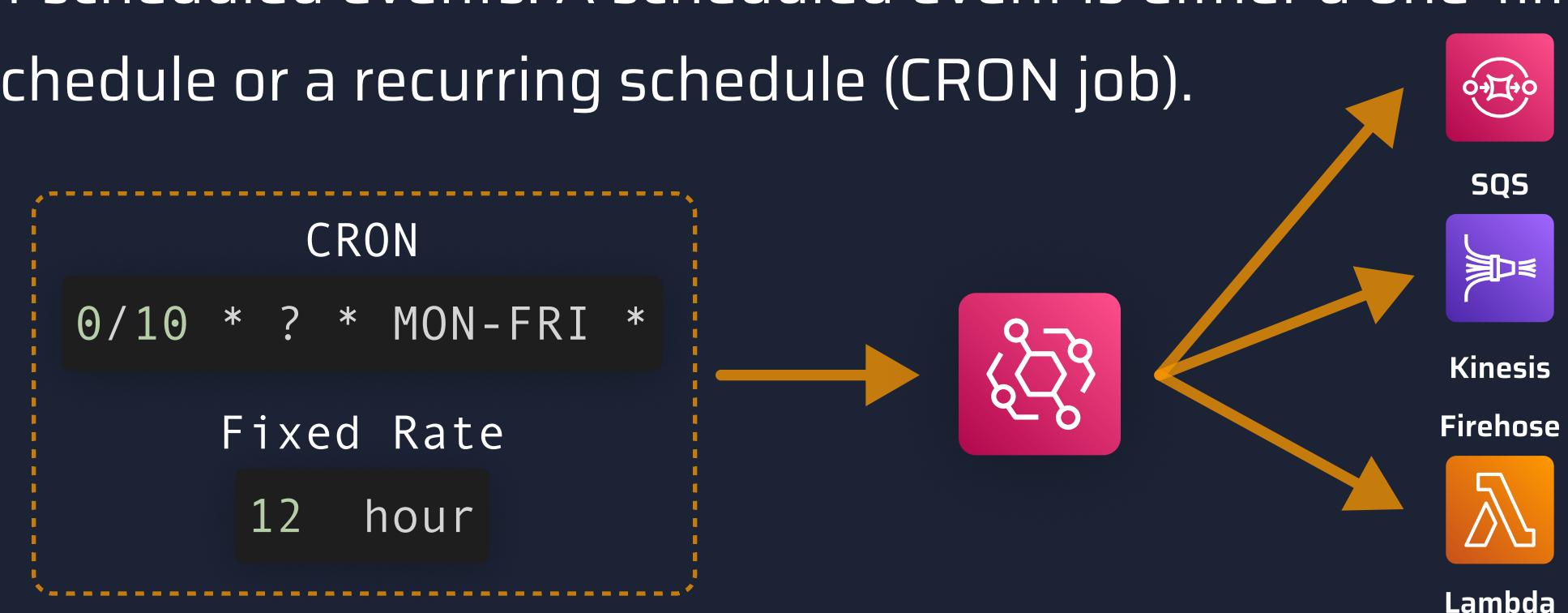
EventBridge is the ultimate service for implementing an event-driven architecture (EDA).

The main goal is to decouple the producer and consumer of events. In a synchronous architecture, the API tells the consumer what to do. In an event-driven way, the endpoint tells the router that something happened, without caring what happens next.

This helps to decouple services

SCHEDULE TASKS ⏳

One of the first functionalities in EventBridge was the usage of scheduled events. A scheduled event is either a one-time schedule or a recurring schedule (CRON job).



A recurring schedule can be created with a **CRON syntax** or a **human-readable fixed rate**. The human-readable syntax simply defines a rate.

SCHEMA REGISTRY 📄

For internal AWS events, there are many schema bindings available. You can find them in the EventBridge console under [Schema Registry > AWS event schema registry](#).



The custom schema registry allows you to create schemas by yourself. This is very helpful if you are using a custom event bus and want to share the schema. This schema can then be shared inside your organization.

EVENT BUS ➔

The event bus is the receiver of all events. You always need to define the **event bus name** if you want to send **events** to EventBridge.

There are three different main types of event buses:

- **Default Bus** - each AWS account has a default event bus, where AWS sends internal events. For example, if you launch or remove an EC2 instance.
- **Custom Bus** - For custom events
- **Partner Event Bus** - EventBridge allows you to include Partner events such as MongoDB, Auth0, or Zendesk.

SCHEMA DISCOVERY 🤖

Schema discovery creates your schema automatically. EventBridge discovers all fields of the incoming event and adds them to a schema. The name of the created schema will be `source@detail-type`.

If you start adding and removing attributes your Schema Discovery will be aware of that. It then creates a new schema. It also emits its own event that you can get notified once the schema changes.

AD-HOC TASKS ⚡

The latest innovation in the EventBridge Scheduler was the creation of one-time tasks.

Define a date, time, and flexible time window. This task will be executed at least once in this time period. EventBridge has a one-minute buffer to execute these tasks. Keep that in mind in case you need a more exact execution.

RULES 📝

Event rules define how to route events to its consumers. Once **an event arrives** the rule sends it to the consumer if the **event pattern** matches.



AWS describes this as filter patterns. You don't filter anything, but you try to **match on fields** in the event.

HANDLING FAILURES WITH DLQs 💀

Your service will experience some kind of failure.

1. **EventBridge cannot deliver the event to the target =>** attaching a DLQ to your event rule. You can define retries and the event age before it can be moved to a DLQ.
2. **The Consumer fails** - EventBridge won't get any feedback. Retry handling is up to the consumer. For native services like Lambda, Destinations with `onError` DLQs can be used.

THE SUBSCRIPTION PATTERN ⚙️

We don't want to increase coupling by coupling targets to the rules. The consumers should be able to flexible subscribe and unsubscribe. Also, we remember that there's a limit of 5 targets per rule.

Instead of attaching multiple consumers to a single rule, we create a rule per consumer. This way, we decouple the consumers from the rule and we're flexible about changing.

The consumers themselves decide if they want to change the pattern or remove it. Not the rule. There shouldn't be any side effects when a consumer wants to change the rule.

EVENTS ⚡

The event itself is a JSON message that always includes the properties **detail**, **detail-type**, and **source**.

- **detail** - your event message
- **detail-type** - a description of your event type
- **source** - the source of your event

There are many more fields in the event like **version**, **account**, and **time**. The three stated above are the most important ones.

ARCHIVE & REPLAY 🗃

Archives save all incoming events in an S3 bucket (not accessible by you).

The replay functionality uses all events saved in the archive and sends them back to the event bus. You can define a start and an end time for the replay. It is also possible to define the rule the archive should replay the events to.

QUOTAS & PRICING 💰

There are quotas around event buses, publishing events per second, and many more.

EventBridge Quotas		
Description	Quota	
Event Buses		100 per account
Publishing Events per second		Between 10,000 (us-east-1) & 400 (eu-south-1)
API Destinations		3,000
Rules Targets		5 per rule
Event Pattern		2,048 characters max
Invocations per second		Between 18,750 (US) and 750 (Asia, Africa)
Schema Discovery		Nested up to 255 levels

The default service events are **free**. You only pay for custom and partner events. Current rate is **\$1.00 per one million events**.

