AWS Eventbridge

AWS EventBridge is a serverless event bus service that makes it easier to build event-driven applications by using events generated from your applications, integrated SaaS applications, and AWS services.

Main reason would be

Event-Driven Architecture:

- **Decoupling:** EventBridge allows you to build applications that react to events in a decoupled manner. This means that the components of your application can work independently and only communicate through events.
- **Scalability:** It helps in building scalable systems where components can be scaled independently based on the events they need to handle.

There are few terms we need to know before we jump into the EventBridge...!!

- **Event:**

  A signal that something has happened in your system. Events are represented as JSON objects and can include details about the event source and the event itself.

- **Event Bus:**

  A channel through which events flow. EventBridge provides a default event bus and allows you to create custom event buses for your applications or to integrate with SaaS partners.

- **Event Source:**

  The origin of an event. Event sources can be AWS services, custom applications, or third-party SaaS applications integrated with EventBridge.

- **Rule:**

  Defines how to process an event. Rules can filter events based on event patterns and route them to one or more targets for processing.

AWS
Eventbridge

- **Target:**

  The destination for an event. Targets can include AWS services like Lambda, SQS, SNS, Kinesis Data Streams, Step Functions, and more. Each rule can route events to multiple targets.

- **Event Pattern:**

  A JSON structure used in rules to filter events. Event patterns specify which events match the criteria defined in the rule, allowing selective routing of events to targets.

- **Schema:**

  The structure of an event, including its fields and data types. Schemas help in understanding the format of events and can be stored in the EventBridge Schema Registry.

- **Schema Registry:**

  A repository for storing and managing event schemas. It enables schema discovery and versioning and allows you to generate code bindings for events in different programming languages.

- **Event Archive:**

  A feature that allows you to store and replay events. Archived events can be replayed for troubleshooting, testing, or reprocessing.

- **Event Bus Policy:**

  Defines permissions for event buses, allowing you to control who can publish events to or subscribe to events from your event buses.

- **Partner Event Source:**

  An integration with a third-party SaaS application that publishes events to EventBridge. Partner event sources need to be activated and linked to your account before you can use them.

- **Custom Event Bus:**

  An event bus you create to handle events specific to your application or to segregate events from different sources. Custom event buses provide better organization and management of events.

AWS
Eventbridge

- **Event Replay**:

  The capability to reprocess past events stored in an archive. This is useful for debugging, testing, or reprocessing events after code changes.

AWS
Eventbridge

Types of EventBridge

Default Event Bus:

- **Description:** The default event bus is provided by AWS EventBridge and is automatically available in your AWS account. It receives events from AWS services (e.g., EC2, S3, RDS, etc.) and supports custom events from your applications.
- **Use Case:** Ideal for handling standard AWS service events and custom events without needing additional configuration.

Custom Event Buses:

- **Description:** These are user-created event buses that allow you to organize events from your applications and different sources. Custom event buses provide more control over event routing and isolation.
- **Use Case:** Suitable for applications that need separate event buses for different environments (e.g., dev, staging, production) or for isolating events from different parts of a large application.

Partner Event Buses:

- **Description:** Partner event buses are used to integrate events from third-party SaaS applications. SaaS providers can send events directly to these partner event buses.
- **Use Case:** Useful for integrating and reacting to events from SaaS applications like Zendesk, Shopify, or Datadog.

AWS Eventbridge

On high level

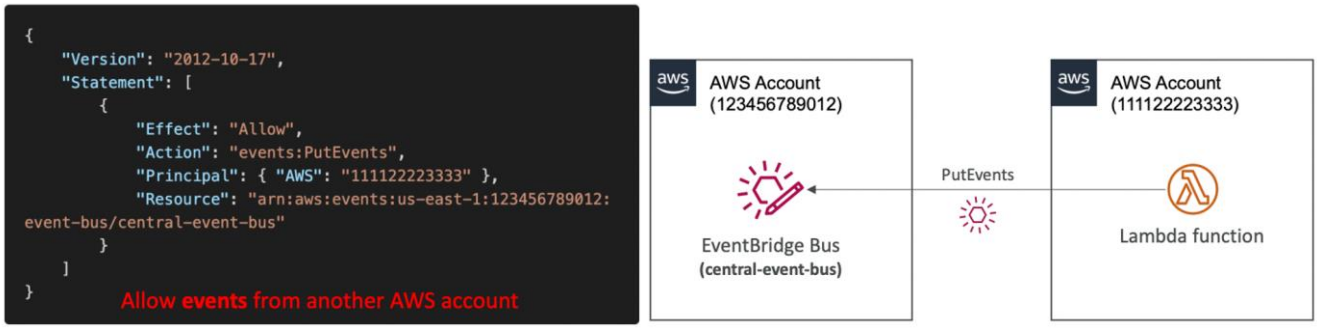
- Extension of CloudWatch Events
- Event buses types:
    - **Default event bus:** events from AWS services are sent to this
    - **Custom Event bus:** for your own applications
    - **Partner event bus:** receive events from external SaaS applications
- Event Rules: how to process the events
- **Event buses support cross-account access using Event Bus Policy**
- **Cron Jobs**: when creating an EB rule, we can select "Schedule" instead of event pattern to trigger an event based on a cron expression.
- Can archive events (all or based on a filter) sent to an event bus to replay later
- EventBridge delivers a near real-time stream of system events that describe changes in AWS resources. Using simple rules that you can quickly set up, you can match events and route them to one or more target functions or streams.

EventBridge is recommended for decoupling applications that reacts to events from third-party SaaS applications.

AWS
Eventbridge

Event Bus Policy

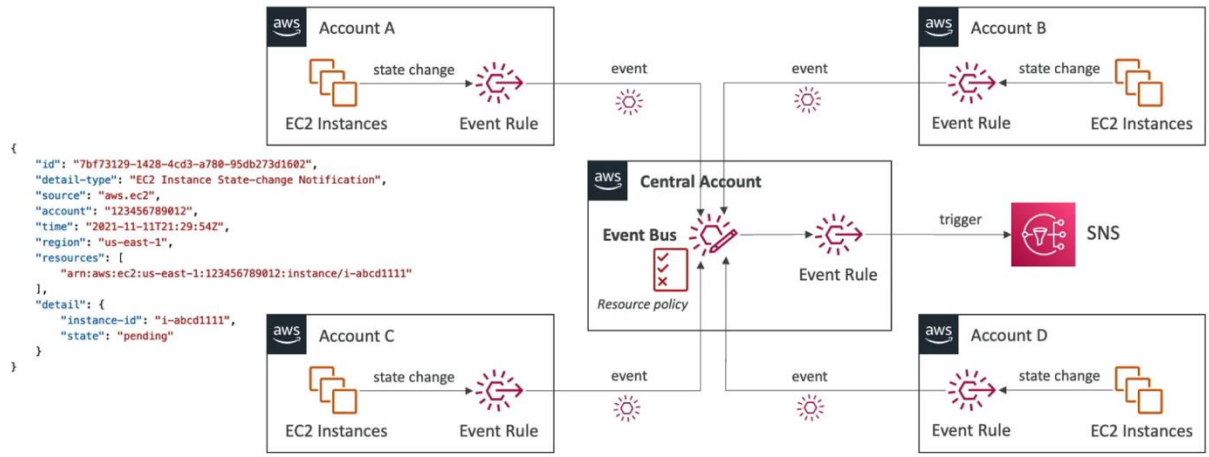
- Manage permissions for an event bus
- Useful to allow or deny events from another AWS account or region



AWS
Eventbridge

Multi-account Aggregation

The target for an event rule in an account can be an event bus in another account. The target event bus needs to have an event bus policy, allowing other accounts to send events into it. This way, a central event bus can be used to aggregate events from multiple accounts.



AWS
Eventbridge

Questions:

https://lisireddy.medium.com/aws-eventbridge-scenario-based-questions-598301d31e9b

Wish you the best …! Happy Learning ..!

Yours' Love (@lisireddy across all the platforms)