Event Driven Service Architecture with Amazon EventBridge

Matt Fitzgerald

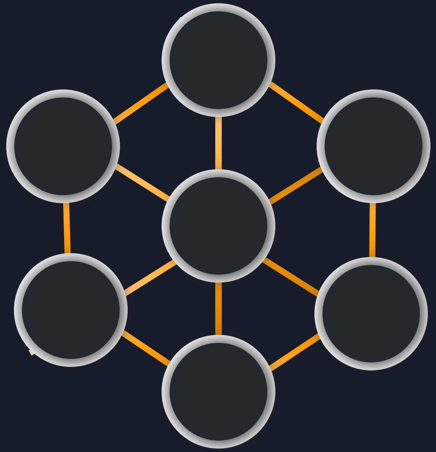
Principal Technical Evangelist, AWS

Event-driven Architectures

Changes to Architectural Patterns

Monolith

Does everything



Microservices

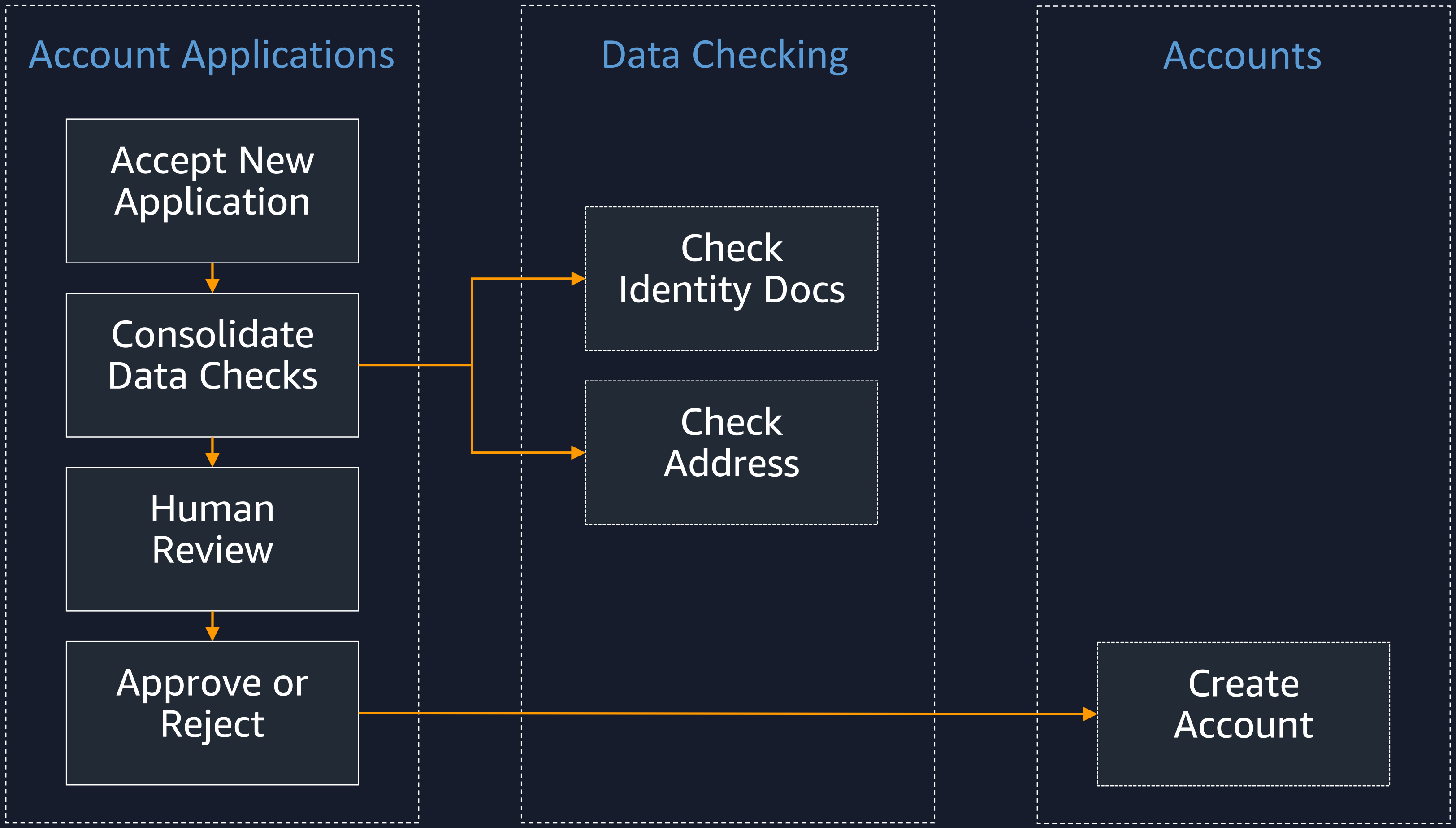
Do one thing

The End Goal

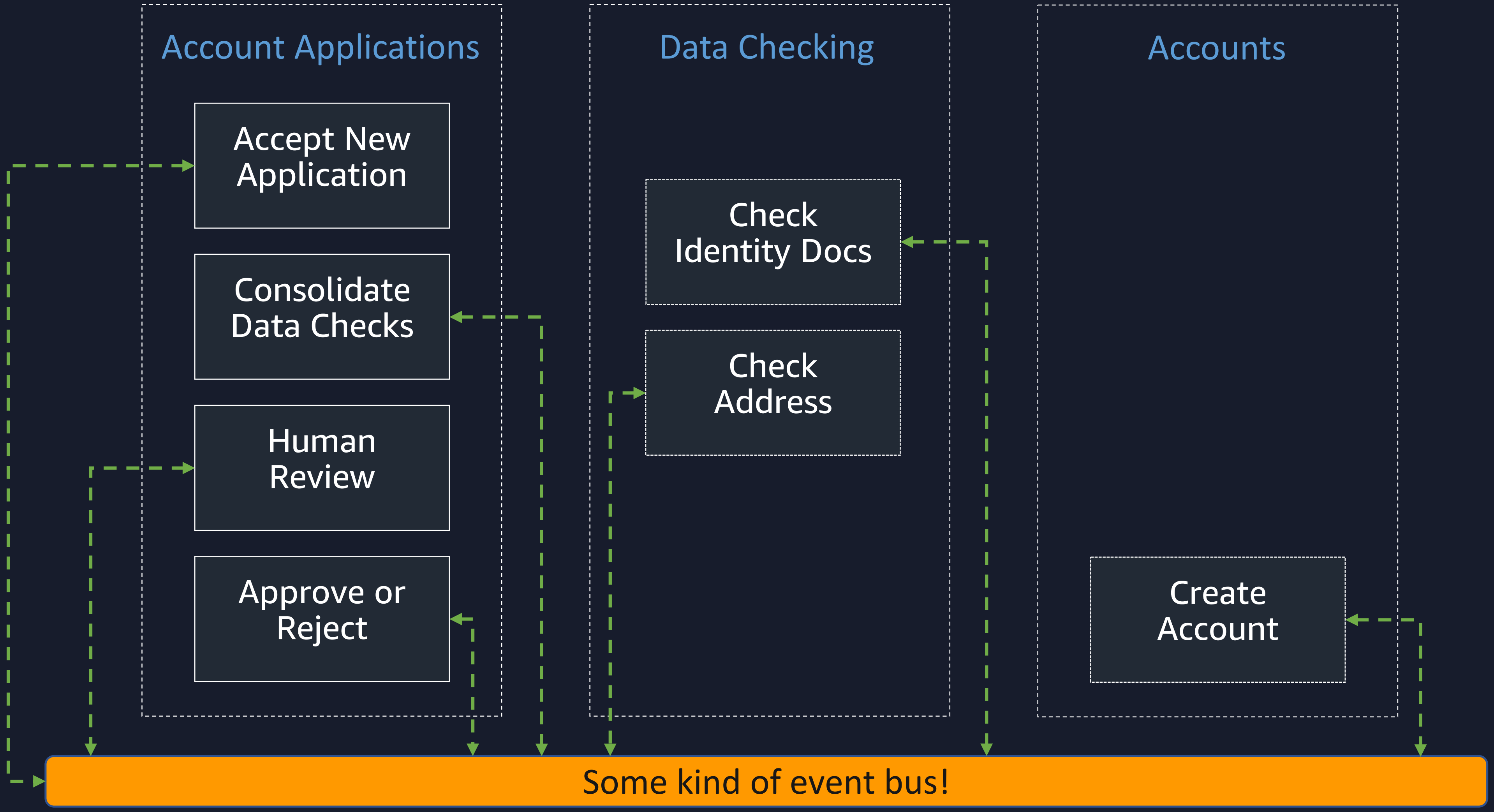
- Customer value
- Reliability
- Resilience
- Scalability

And do it faster

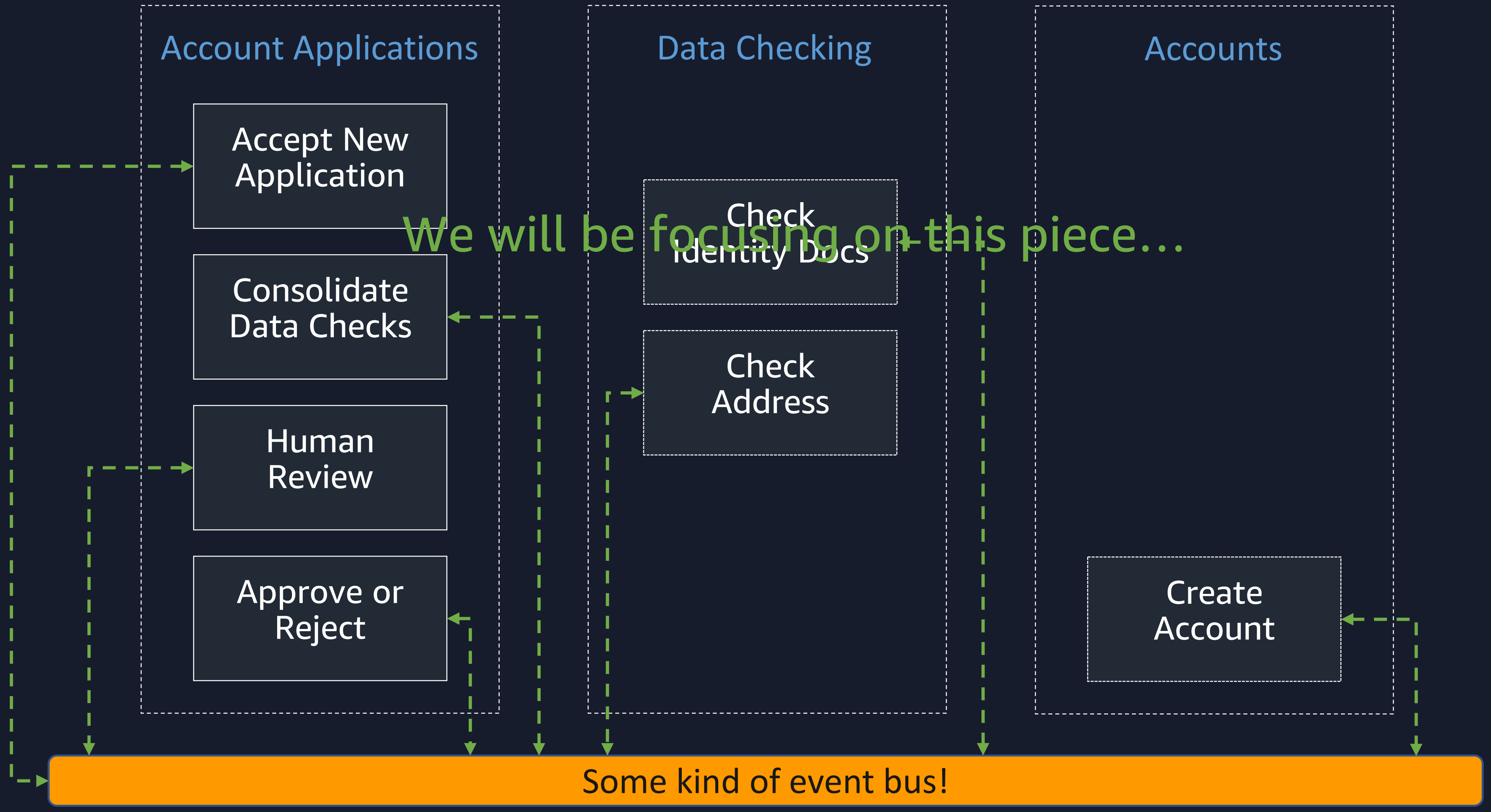
Orchestration vs Choreography



Orchestration vs Choreography

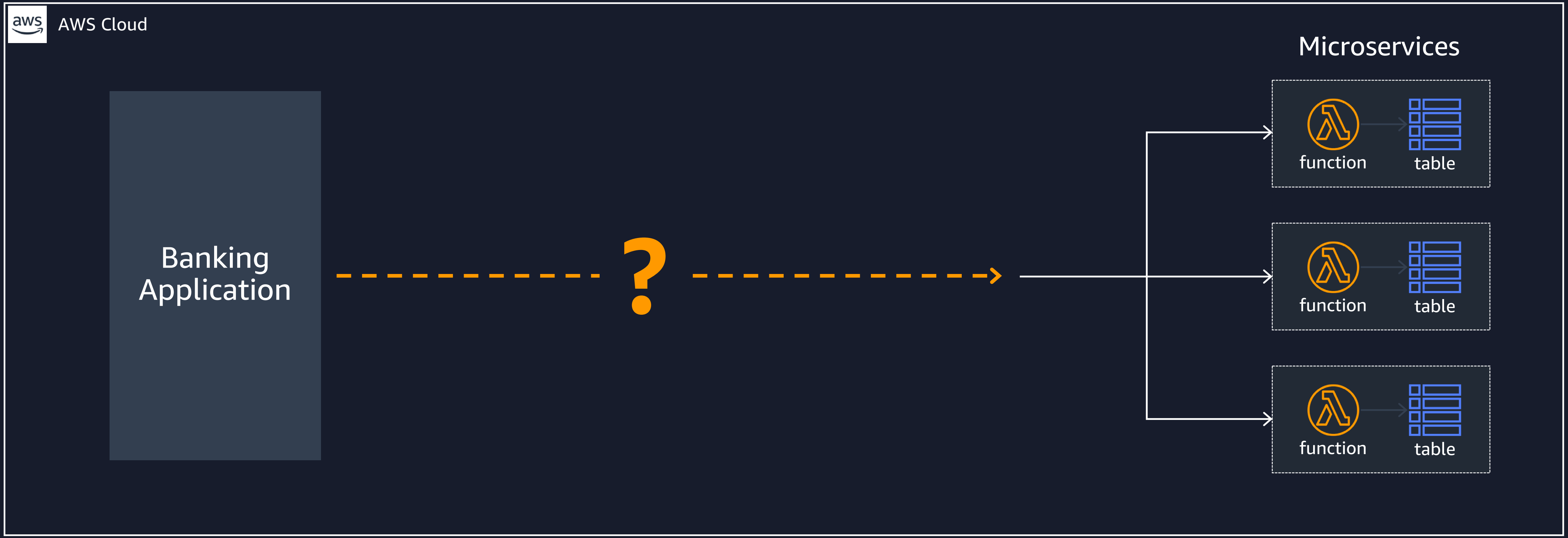


Orchestration vs Choreography



We will be focusing on this piece…

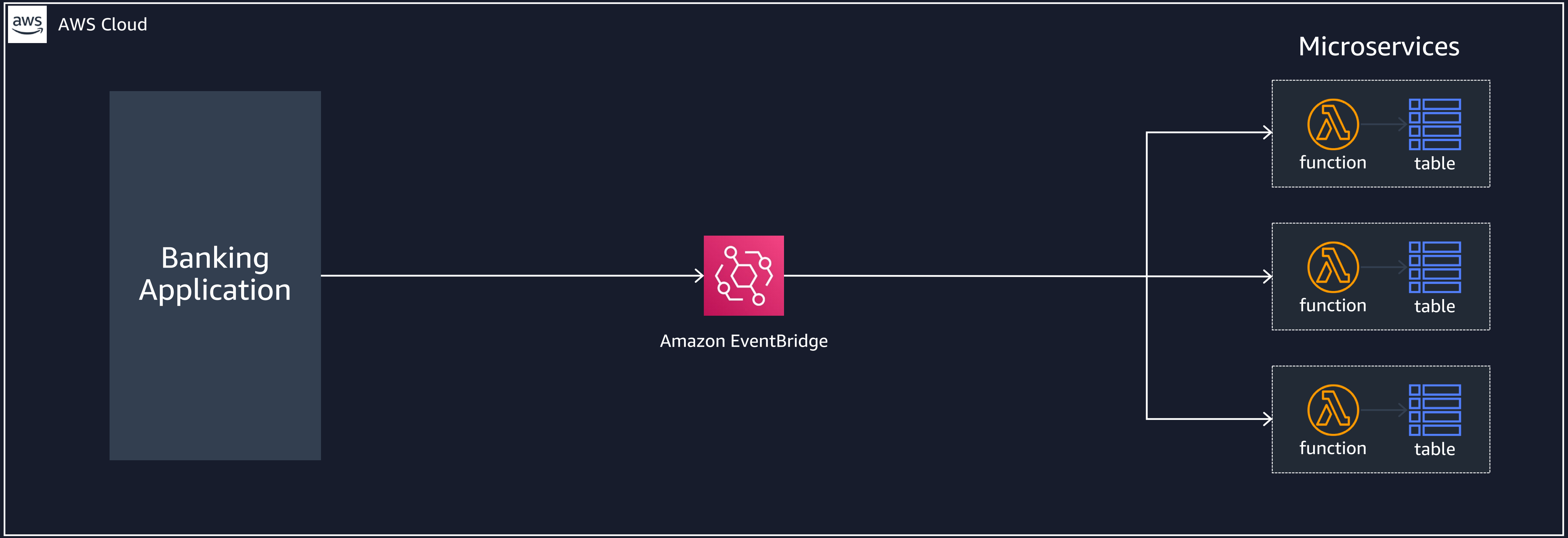
Bridging the Gap - Integration



Banking App Team

e.g. Auditing/Marketing Teams

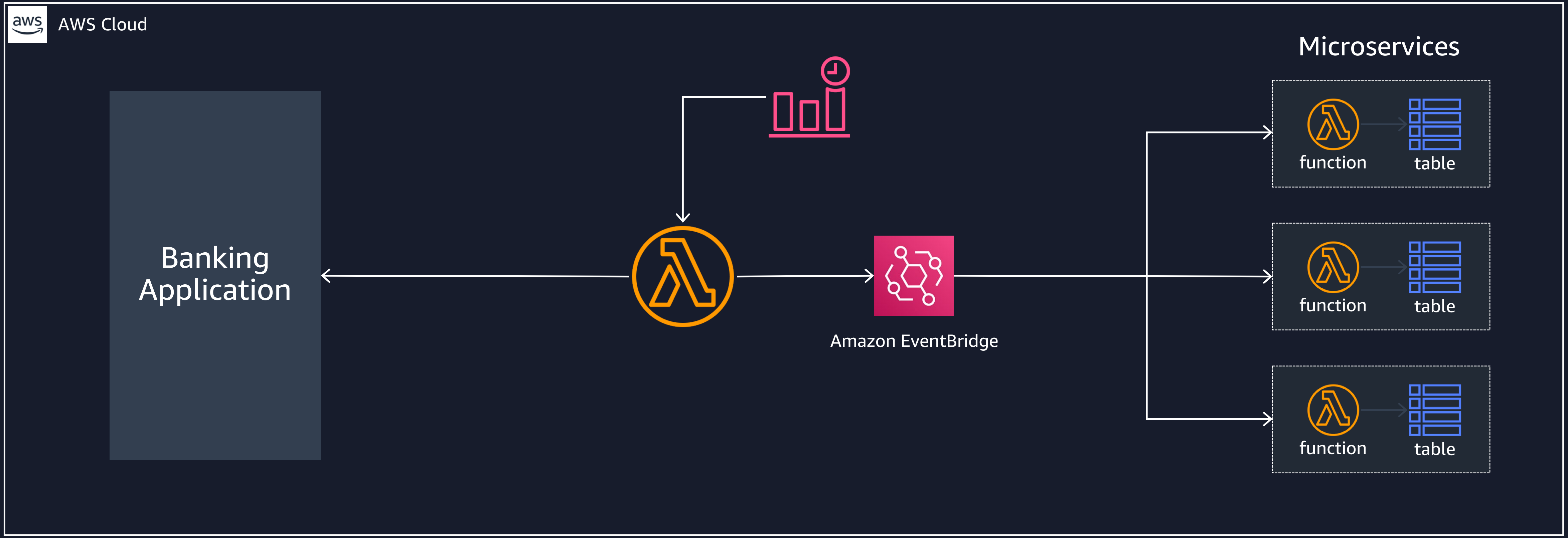
Bridging the Gap - Integration



Banking App Team

e.g. Auditing/Marketing Teams

Bridging the Gap - Polling



Banking App Team

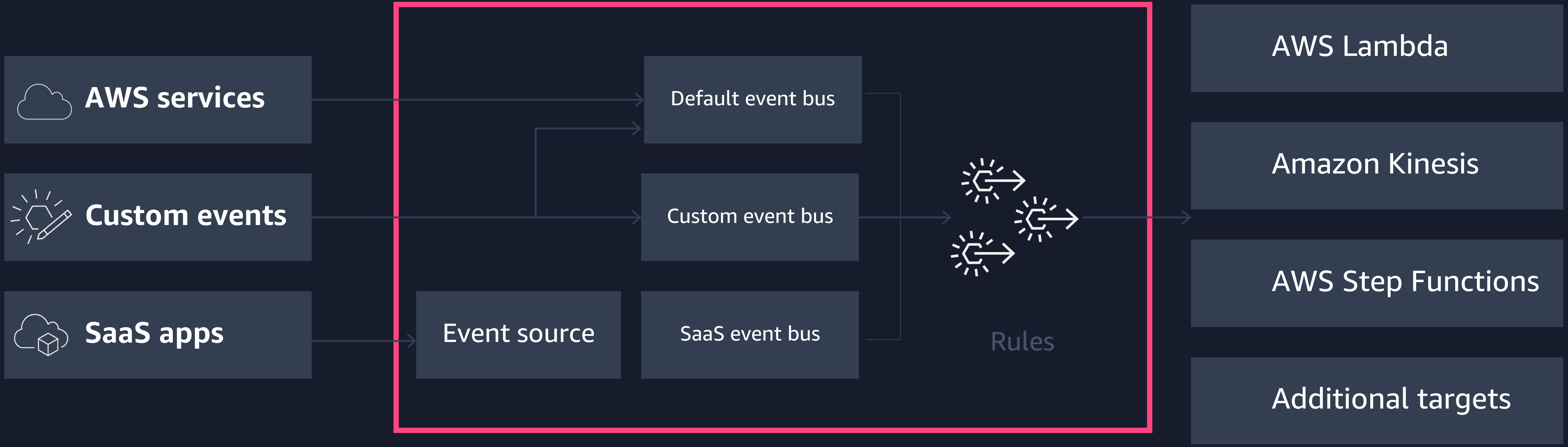
e.g. Auditing/Marketing Teams

Amazon EventBridge - Nomenclature

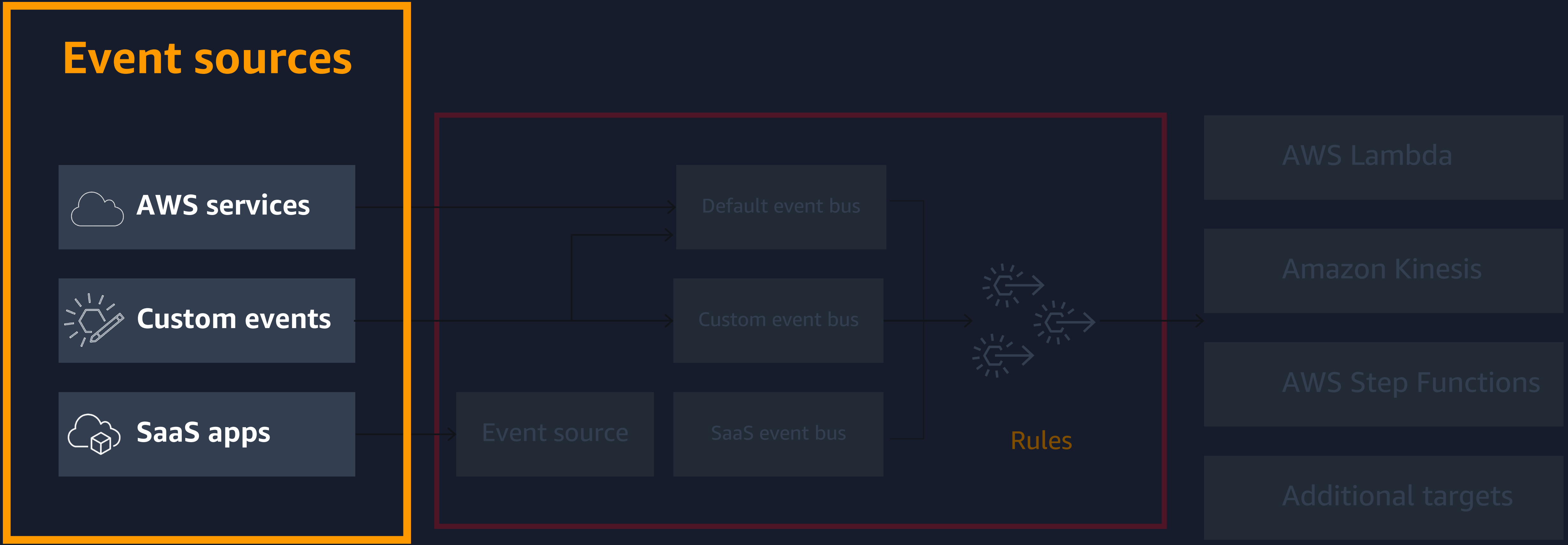
Emitters, agents	Source
Channel	Event Bus
Consumer, sink	Target

Amazon EventBridge
The Basics

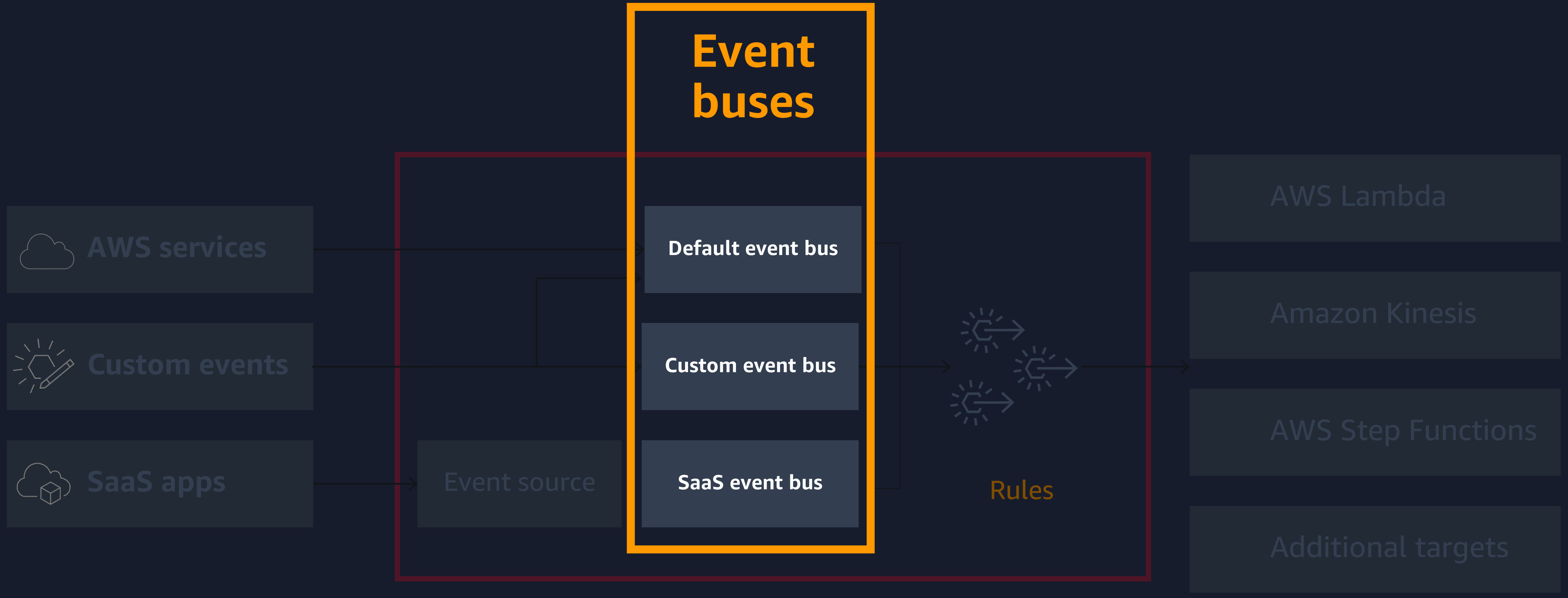
Amazon EventBridge



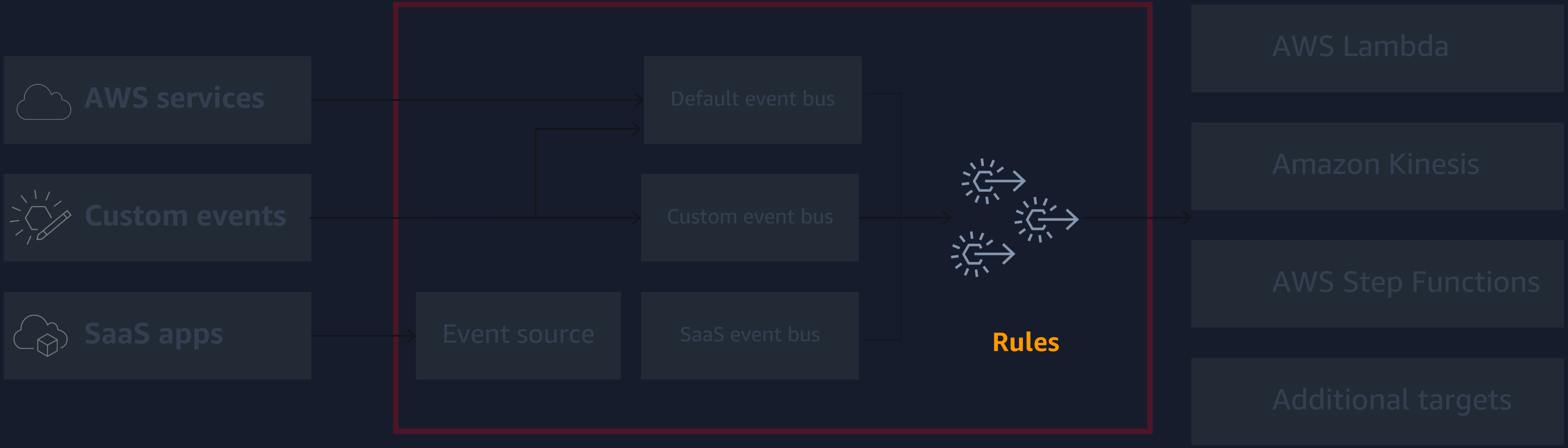
Amazon EventBridge



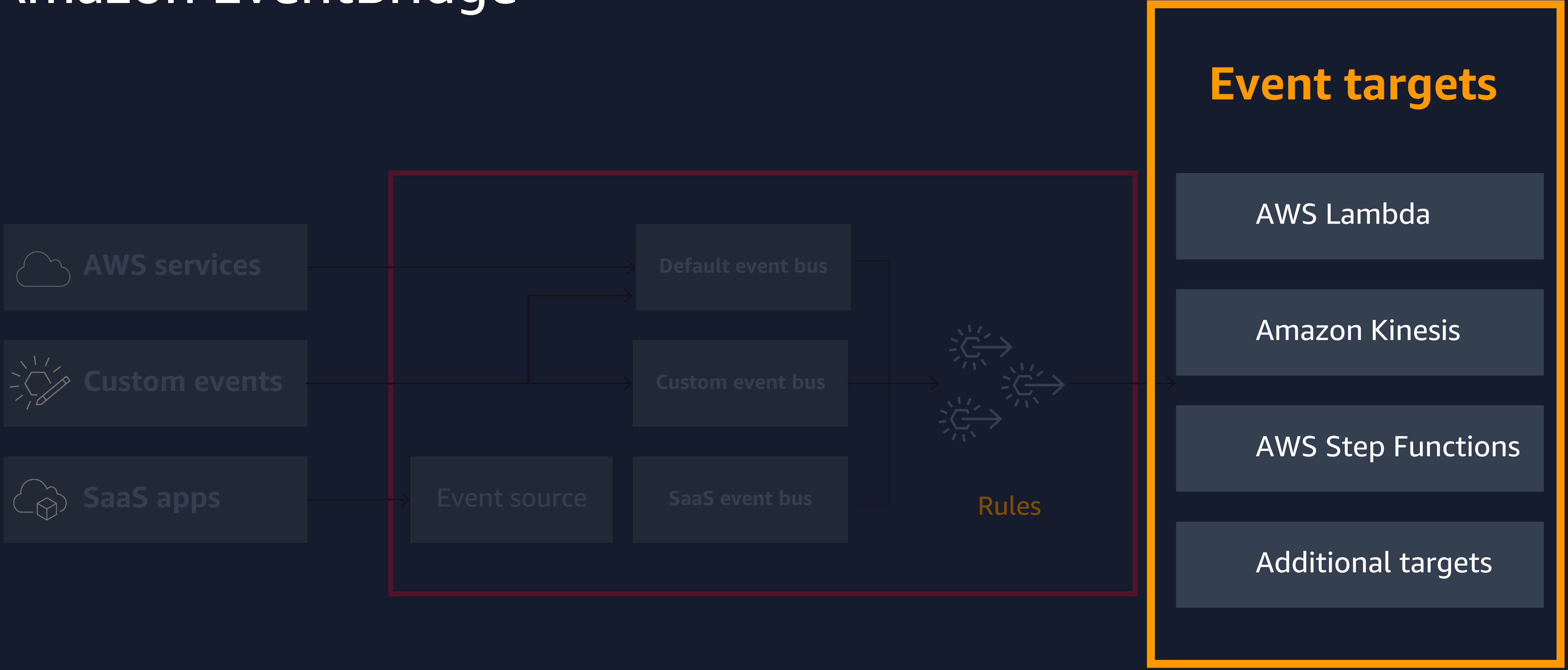
Amazon EventBridge



Amazon EventBridge



Amazon EventBridge

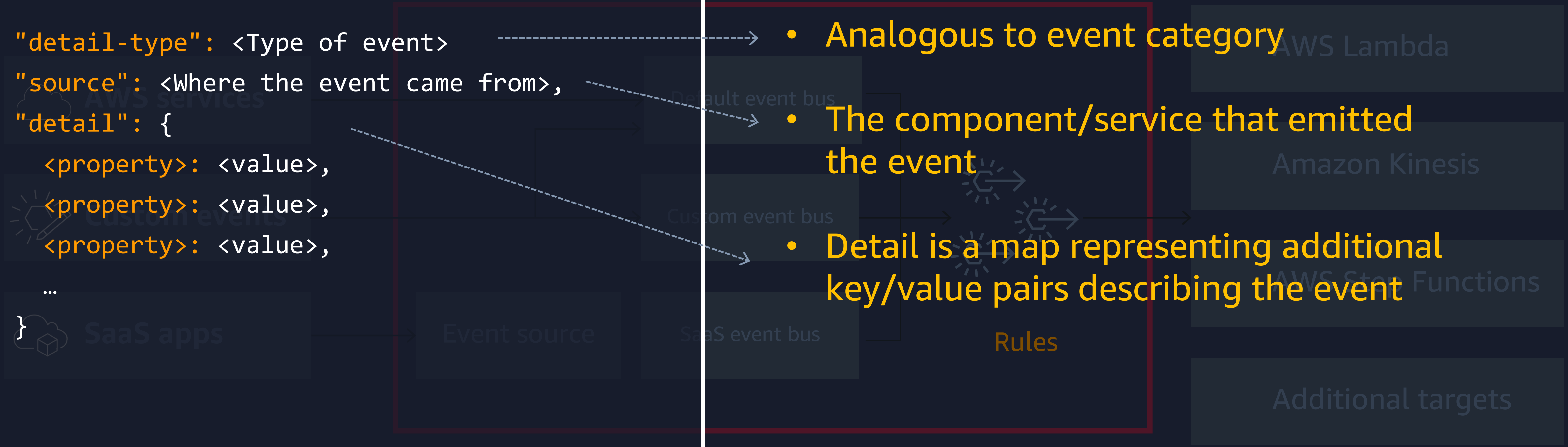


Amazon EventBridge

Event structure:

```
{
  "detail-type": <Type of event>
  "source": <Where the event came from>,
  "detail": {
    <property>: <value>,
    <property>: <value>,
    <property>: <value>,
    …
  }
}
```

- Analogous to event category
- The component/service that emitted the event
- Detail is a map representing additional key/value pairs describing the event



Amazon EventBridge

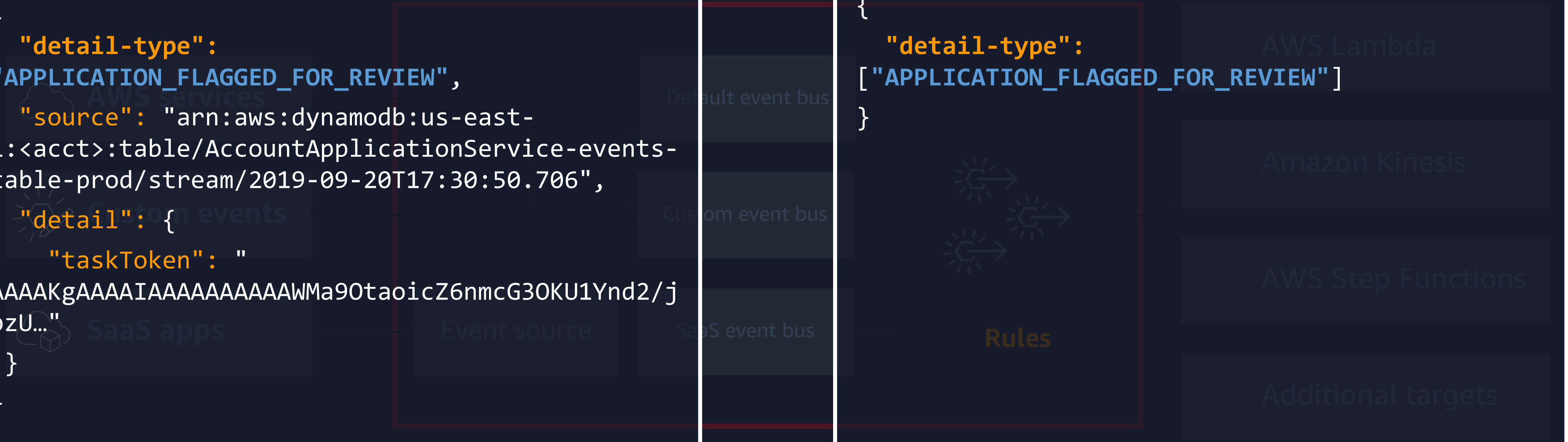
Example event:

```
{
  "detail-type": "APPLICATION_APPROVED",
  "source": "arn:aws:dynamodb:us-east-
1:<acct>:table/AccountApplicationService-events-
table-prod/stream/2019-09-20T17:30:50.706",
  "detail": {
    "name": "Gabe",
    "applicationId": "429901a4-4dc1-4d4f-8cc3-
5f82b3dc13a3",
    "state": "APPROVED"
    "approvalType": "AUTO"
  }
}
```

Example rule:

```
{
  "detail-type": [”APPLICATION_APPROVED"]
  "detail":
  {
    "state": ["APPROVED"],
    "approvalType": ["AUTO", "HUMAN"]
  }
}
```

Amazon EventBridge



Example event:

```
{
  "detail-type":
"APPLICATION_FLAGGED_FOR_REVIEW",
  "source": "arn:aws:dynamodb:us-east-
1:<acct>:table/AccountApplicationService-events-
table-prod/stream/2019-09-20T17:30:50.706",
  "detail": {
    "taskToken": "
AAAAKgAAAAIAAAAAAAAAAWMa9OtaoicZ6nmcG3OKU1Ynd2/j
pzU…"
 }
}
```

Example rule:

```
{
  "detail-type":
["APPLICATION_FLAGGED_FOR_REVIEW"]
}
```

Amazon EventBridge

Example event:

```
{
  "detail-type": "APPLICATION_REJECTED",
  "source": "arn:aws:dynamodb:us-east-
1:<acct>:table/AccountApplicationService-events-
table-prod/stream/2019-09-20T17:30:50.706",
  "detail": {
    "name": "Gabe-evil",
    "applicationId": "2a39eb8-2d69-4859-8168-
68d00cef15fa",
    "state": "SUBMITTED"
  }
}
```

Example rule:

```
{
  "detail-type": ["APPLICATION_APPROVED"]
}
```

Amazon EventBridge: What Does This Mean?

Connect data from other apps

Use data from AWS services, your own custom components and supported SaaS apps to trigger workflows

Write Less Code

Ingest, filter, and deliver events without writing custom code

Reduce Operational Overhead

No servers to manage or software to operate. Scale automatically and only pay for events published

Easily build event-driven architectures

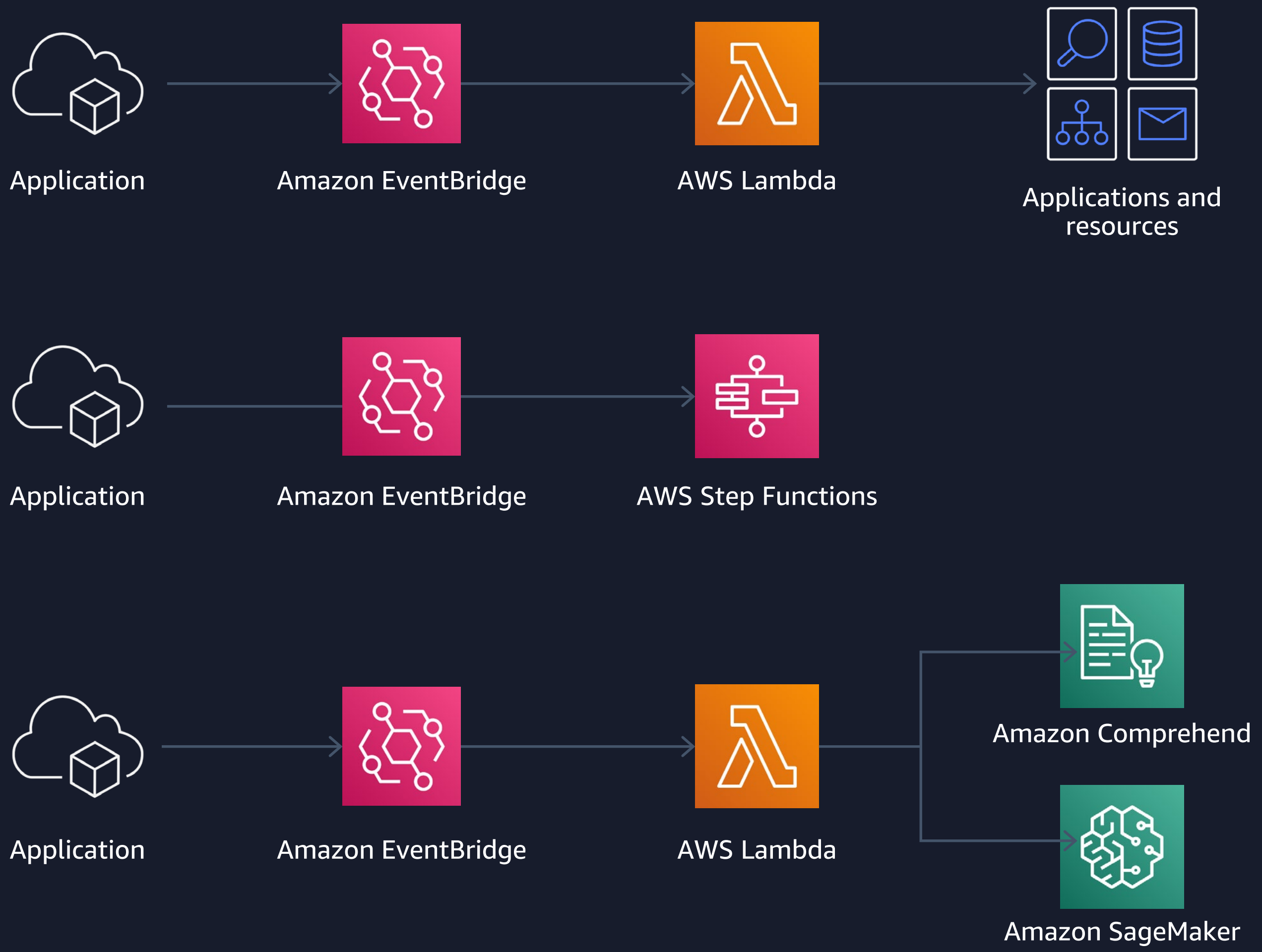
Simplify the process as your event targets don't need to be aware of event sources

Amazon EventBridge: Common Use-Cases

Take action

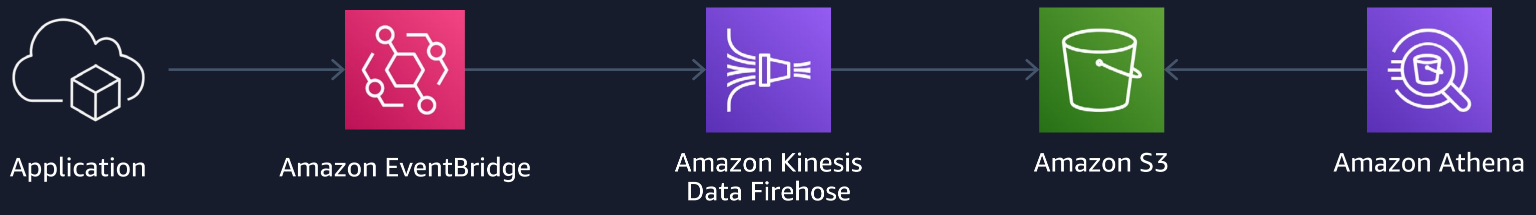
Run workflows

Apply intelligence

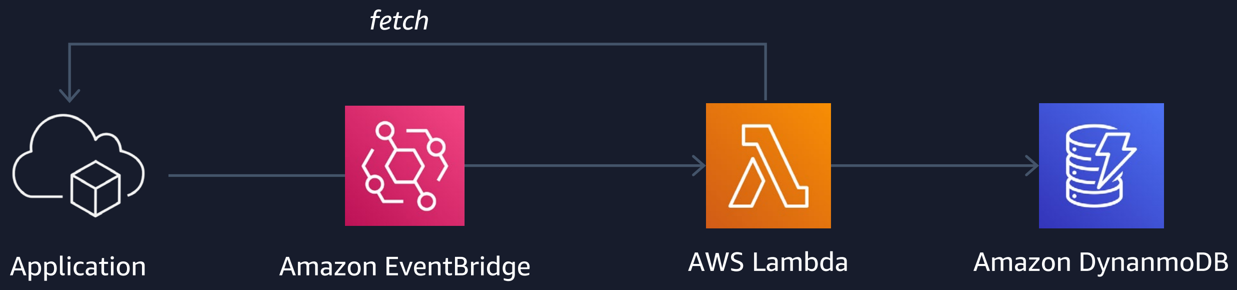


Amazon EventBridge: Common Use-Cases

Audit and analyze

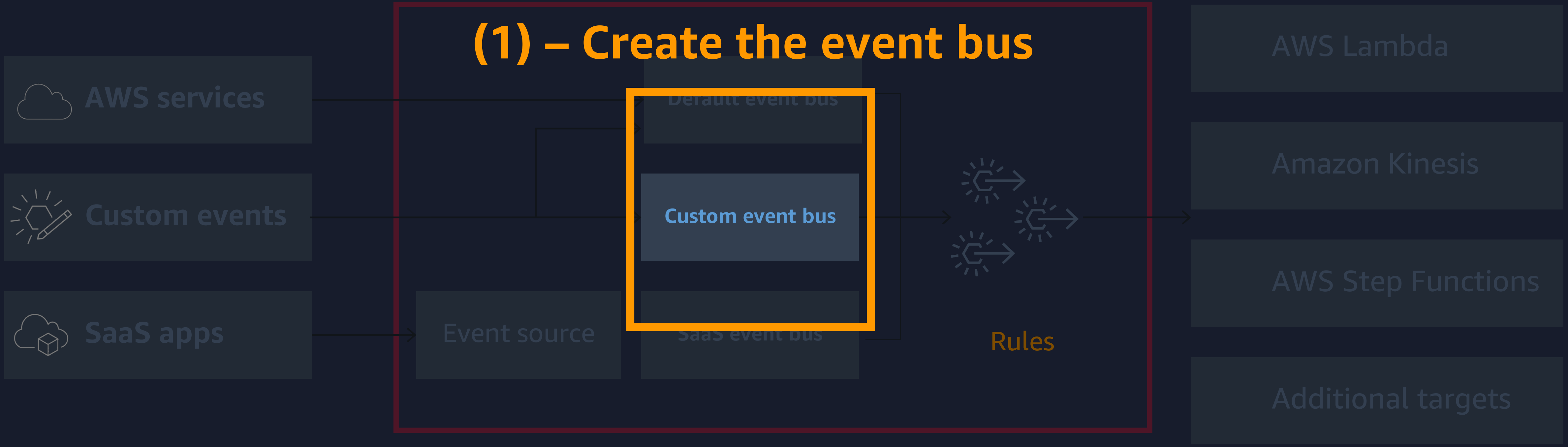


Synchronize data



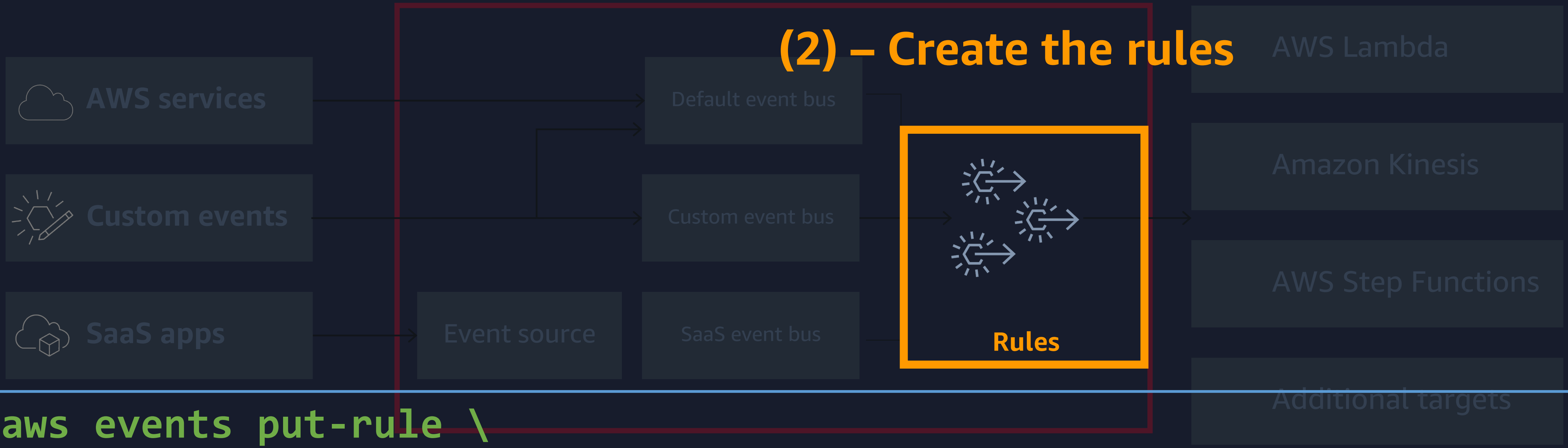
Amazon EventBridge
Diving Deeper

Amazon EventBridge – Create Event Bus



```
> aws events create-event-bus --name "banking-demo-events-dev"
```

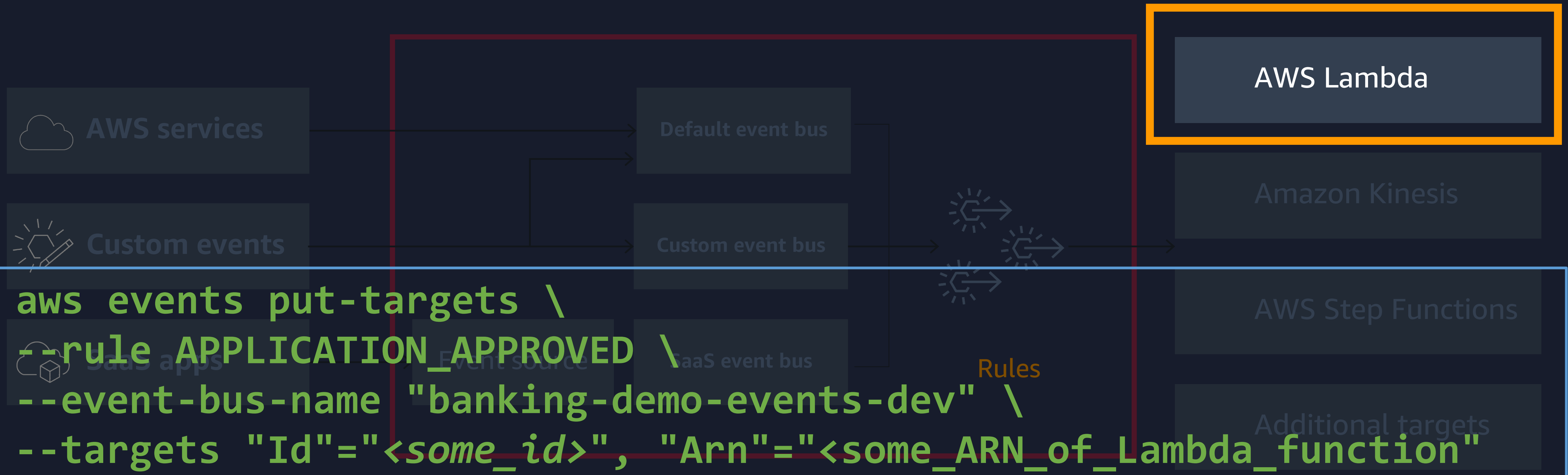
Amazon EventBridge – Create Rules



```
➢ aws events put-rule \
  --name APPLICATION_SUBMITTED \
  --event-bus-name "banking-demo-events" \
  --event-pattern "{ \"detail-type\": [\"APPLICATION_SUBMITTED\"] }" \
  --state ENABLED
```

Amazon EventBridge – Create Targets

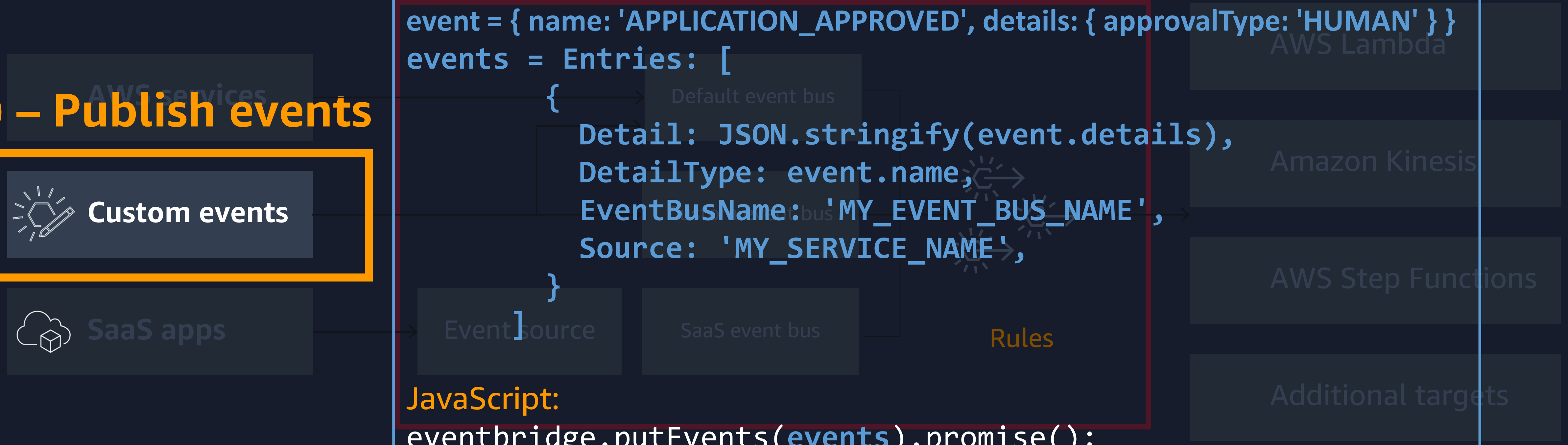
(3) – Create the targets



```
➢ aws events put-targets \
  --rule APPLICATION_APPROVED \
  --event-bus-name "banking-demo-events-dev" \
  --targets "Id"="<some_id>", "Arn"="<some_ARN_of_Lambda_function"
```

Amazon EventBridge – Publish Events

(4) – Publish events



Events Structure:

```
event = { name: 'APPLICATION_APPROVED', details: { approvalType: 'HUMAN' } }
events = Entries: [
          {
            Detail: JSON.stringify(event.details),
            DetailType: event.name,
            EventBusName: 'MY_EVENT_BUS_NAME',
            Source: 'MY_SERVICE_NAME',
          }
        ]
```

JavaScript:

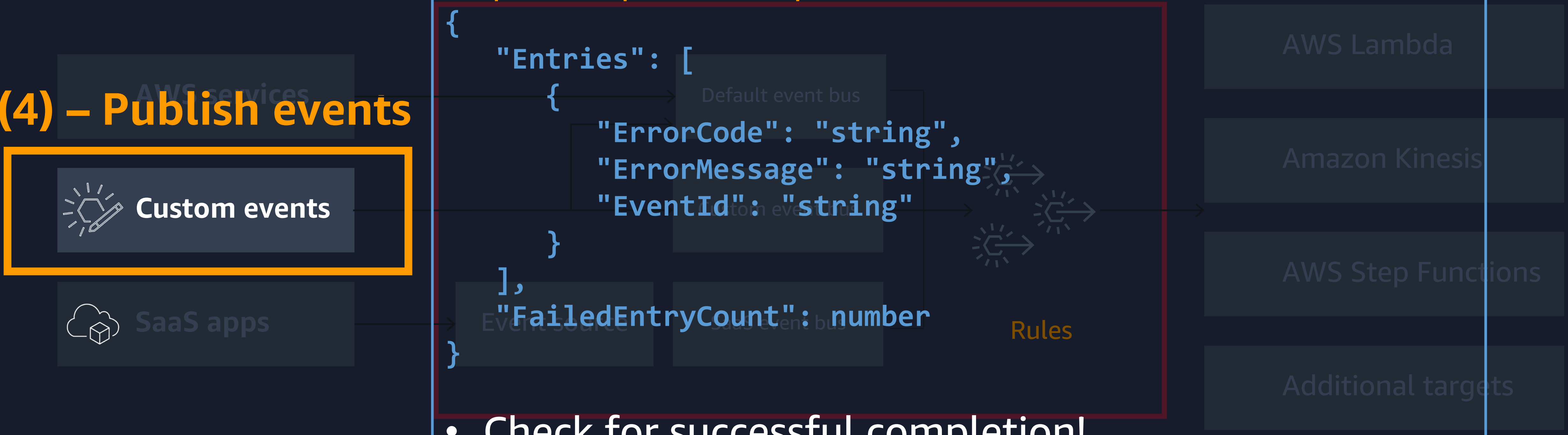
```
eventbridge.putEvents(events).promise();
```

Python:

```
eventbridge.put_events(Entries=events)
```

Amazon EventBridge – Publish Events

(4) – Publish events

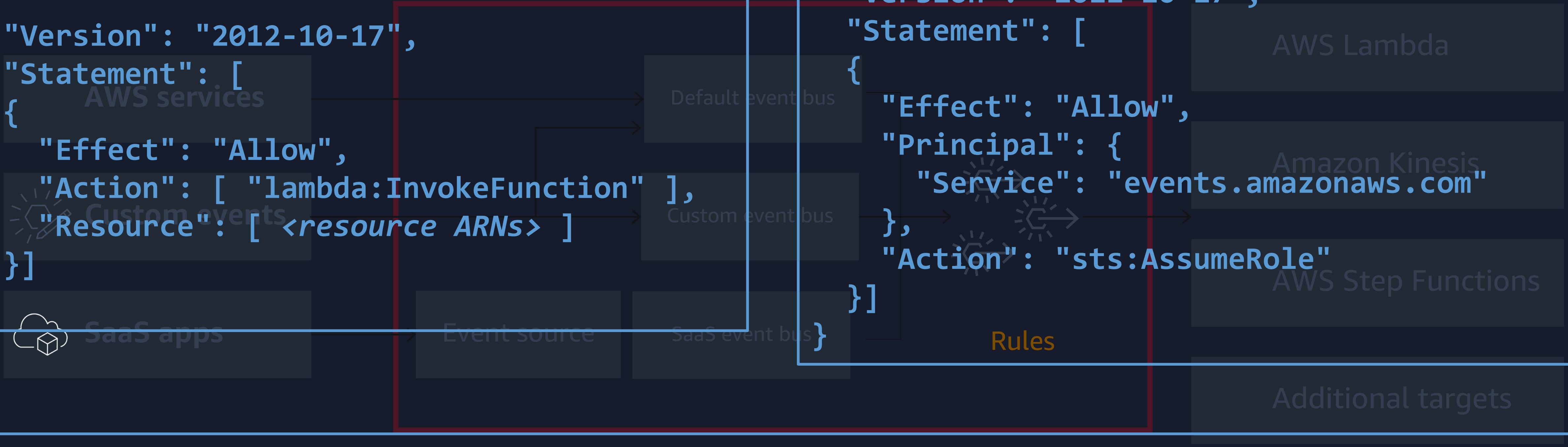


Response of putEvents()/put_events():

```
{
   "Entries": [
      {
         "ErrorCode": "string",
         "ErrorMessage": "string",
         "EventId": "string"
      }
   ],
   "FailedEntryCount": number
}
```

- Check for successful completion!
- "At least once delivery": Deal with idempotency!

Amazon EventBridge – Security



Policy Statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [ "lambda:InvokeFunction" ],
    "Resource": [ <resource ARNs> ]
  }]
}
```

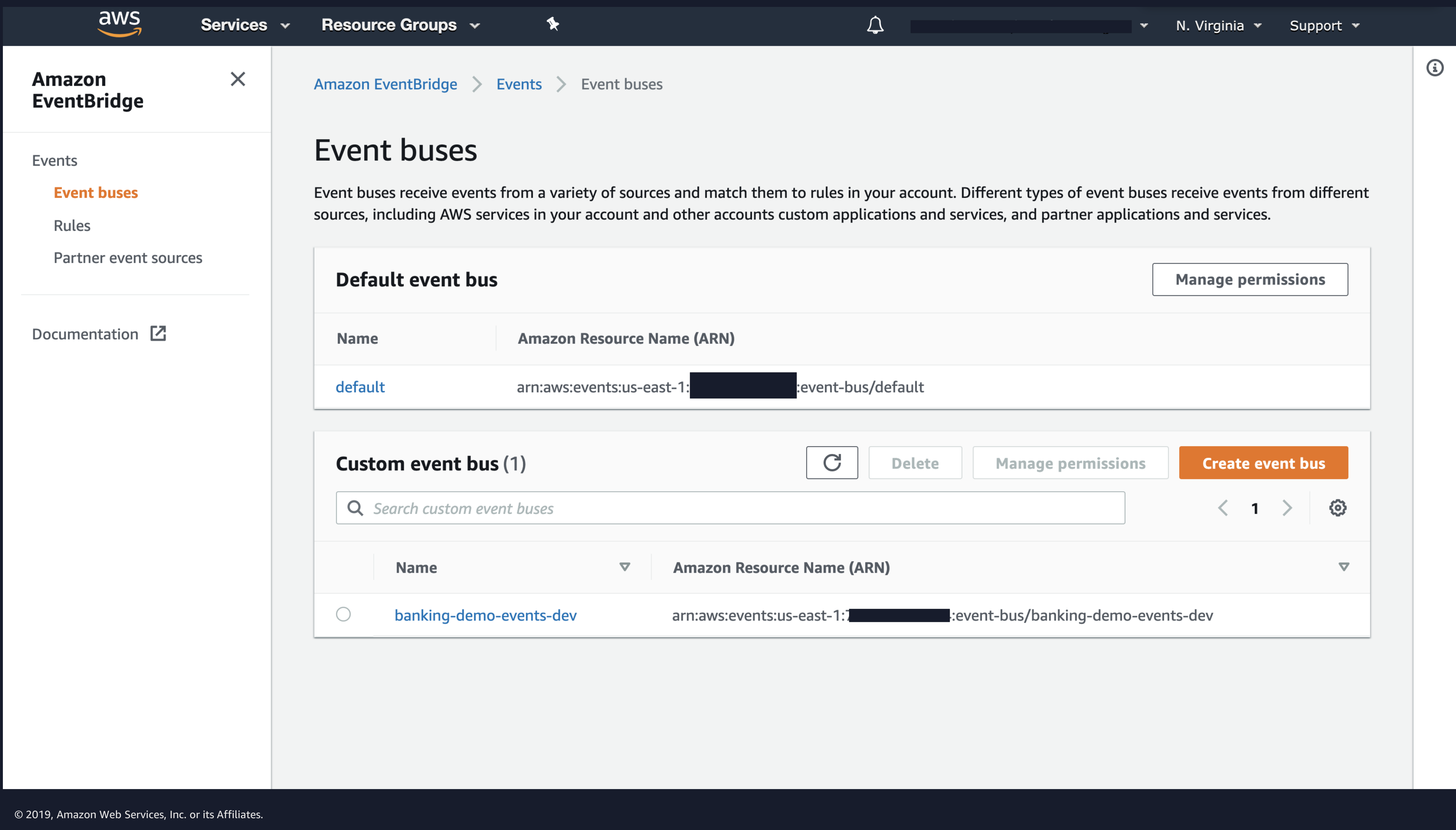
Assume Role Policy Statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

```
➢ aws iam create-role --role-name <name> \
  --assume-role-policy-document <assume_role_policy_doc>
➢ aws iam put_role_policy --role-name <name> \
  --policy-name <policy_name> --policy-document <policy_doc>
```

Amazon EventBridge – Key Limits

# event buses per account	100
# put requests per second	400
# events per put request	10
Max event pattern size	2,048 characters
Target innovations per second	750
# rules per event bus	100
# targets per rule	5



[redacted]

Amazon EventBridge

Events

Event buses

Rules

Partner event sources

Documentation

Amazon EventBridge > Events > Event buses

Event buses

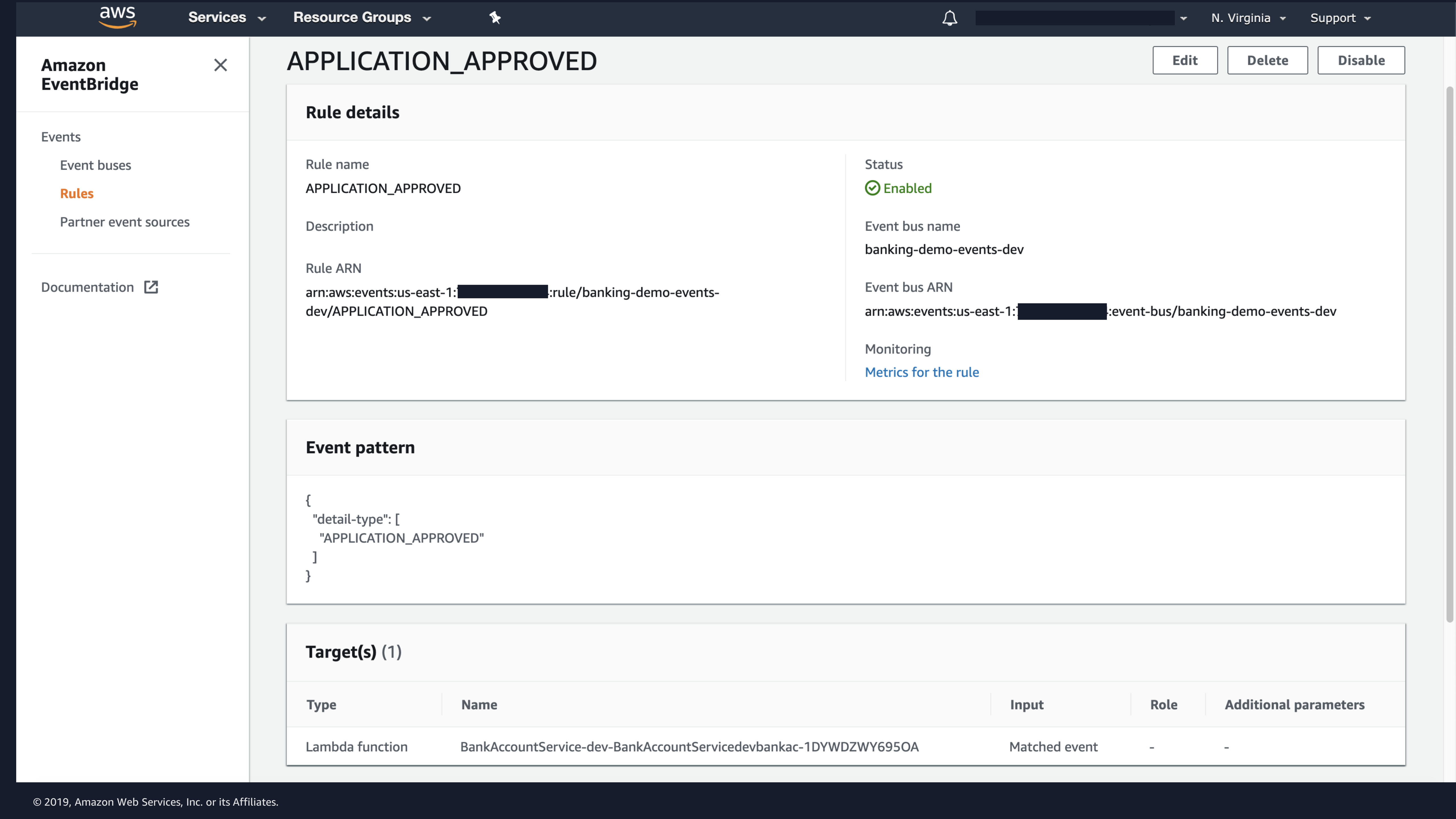
Event buses receive events from a variety of sources and match them to rules in your account. Different types of event buses receive events from different sources, including AWS services in your account and other accounts custom applications and services, and partner applications and services.

Default event bus

Name	Amazon Resource Name (ARN)
default	arn:aws:events:us-east-1:[redacted]:event-bus/default

Custom event bus (1)

	Name	Amazon Resource Name (ARN)
○	banking-demo-events-dev	arn:aws:events:us-east-1:[redacted]:event-bus/banking-demo-events-dev



Amazon EventBridge

Events

Event buses

Rules

Partner event sources

Documentation

APPLICATION_APPROVED

Rule details

Rule name
APPLICATION_APPROVED

Description

Rule ARN
arn:aws:events:us-east-1:[redacted]:rule/banking-demo-events-dev/APPLICATION_APPROVED

Status
Enabled

Event bus name
banking-demo-events-dev

Event bus ARN
arn:aws:events:us-east-1:[redacted]:event-bus/banking-demo-events-dev

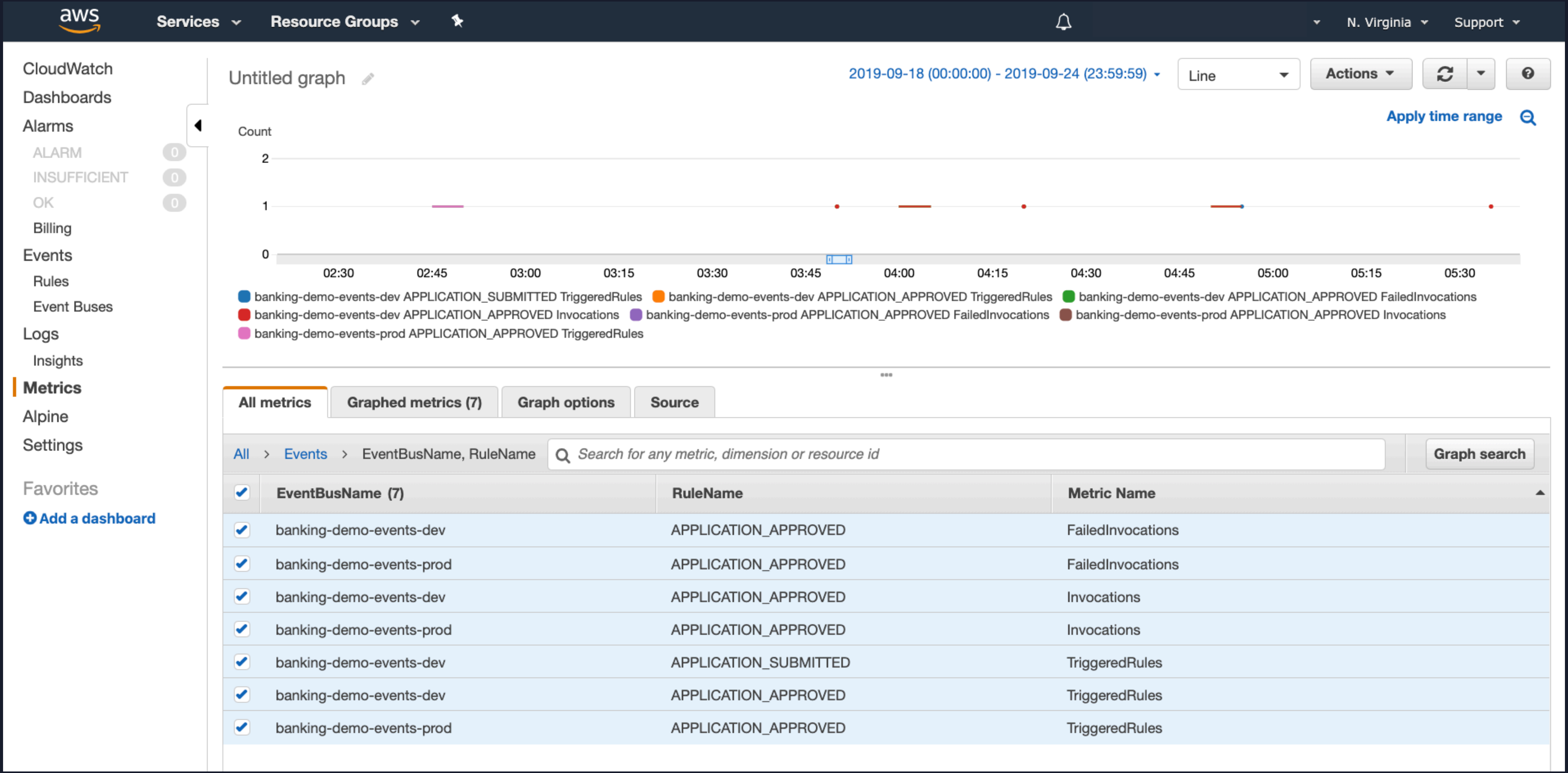
Monitoring
Metrics for the rule

Event pattern

```
{
  "detail-type": [
    "APPLICATION_APPROVED"
  ]
}
```

Target(s) (1)

Type	Name	Input	Role	Additional parameters
Lambda function	BankAccountService-dev-BankAccountServicedevbankac-1DYWDZWY695OA	Matched event	-	-



2019-09-18 (00:00:00) - 2019-09-24 (23:59:59)

🟦 banking-demo-events-dev APPLICATION_SUBMITTED TriggeredRules 🟧 banking-demo-events-dev APPLICATION_APPROVED TriggeredRules 🟩 banking-demo-events-dev APPLICATION_APPROVED FailedInvocations
🟥 banking-demo-events-dev APPLICATION_APPROVED Invocations 🟪 banking-demo-events-prod APPLICATION_APPROVED FailedInvocations 🟫 banking-demo-events-prod APPLICATION_APPROVED Invocations
🟪 banking-demo-events-prod APPLICATION_APPROVED TriggeredRules

All > Events > EventBusName, RuleName

✓	EventBusName (7)	RuleName	Metric Name
✓	banking-demo-events-dev	APPLICATION_APPROVED	FailedInvocations
✓	banking-demo-events-prod	APPLICATION_APPROVED	FailedInvocations
✓	banking-demo-events-dev	APPLICATION_APPROVED	Invocations
✓	banking-demo-events-prod	APPLICATION_APPROVED	Invocations
✓	banking-demo-events-dev	APPLICATION_SUBMITTED	TriggeredRules
✓	banking-demo-events-dev	APPLICATION_APPROVED	TriggeredRules
✓	banking-demo-events-prod	APPLICATION_APPROVED	TriggeredRules

Amazon EventBridge
In Action

Datadog Enables Ticket Prioritization and Routing with Zendesk EventBridge Integration

Challenge

To provide timely response to their customers by leveraging Zendesk's Customer Experience platform, Datadog needed information across various sources and smarter routing for faster case resolution. Their 100+ support and solution engineers dealt with fluctuating capacity, often handling very high volumes of support tickets per day.

Solution

Datadog uses EventBridge for real-time routing and prioritization of Zendesk Support tickets. Events are sent from Zendesk to EventBridge when a new customer support ticket is opened with Datadog, and those events are then sent to AWS Lambda to process the case complexity and to route to the best support team to acress.

Benefits

- More timely response to event activity, leading to higher customer satisfaction
- Seamless integration into Datadog for advanced analytics
- Engineers spend time solving issues, as opposed to managing APIs
- Zendesk's systems are not hit as hard

Zendesk's Events Connector for AWS enables streaming of event data from Zendesk's support suite into Amazon EventBridge. The native connector will allow any organization to easily utilize AWS services while reducing dependencies and overhead from API based integrations and API management. With Zendesk + AWS, organizations will be able to leverage their customer data for BI and Analytics, Machine Learning, Cloud Access Security Brokerage (CASB), and Security and Audit needs. AWS services that can be leveraged include: S3, Kinesis, Lambda, SNS, SQS and more.

Cox Automotive Has Faster Incident MTTR with PagerDuty EventBridge Integration

Challenge

Cox Automotive cannot afford to have any of their infrastructure down, which would lead to dissatisfied or lost customers. As they are using different tools and systems, including PagerDuty's incident reporting, they need to improve efficiency for when unexpected downtime occurs.

Solution

Cox Automotive uses EventBridge to build a runbook automation script. Anytime a new incident is created in PagerDuty and the event is sent to EventBridge, a AWS Lambda function is triggered to determine which systems are impacted, pull down an existing runbook from Amazon S3, and attach that runbook in the notes field of the incident report.

Benefits

- Operators can immediate look at the initial troubleshooting steps taken directly from the incident report
- Decrease time dealing with complex webhook or other manual configurations
- Direct integration of PagerDuty events into event-driven workflows

"AWS EventBridge, combined with PagerDuty, helps us generate event-driven workflows in real time. When we detect an issue, PagerDuty can generate an alert that triggers an AWS Lambda function to grab runbooks and post details back into PagerDuty, helping us resolve issues faster and create the best experience for our customers." - Ed Kozlowski, Lead Software Engineer at Cox Automotive.

PagerDuty's platform for real-time IT operations integrates with Amazon EventBridge to enable teams to take action on issues, not just alert on them. More specifically, PagerDuty's integration for Amazon EventBridge helps teams utilize PagerDuty event data to trigger event-driven workflows across the AWS ecosystem. You can monitor and proactively act on security, compliance, resource deployment, customer service, and many other datasets to automatically take action within your AWS environment, freeing teams to spend less time fixing issues and more time innovating. With Amazon EventBridge, PagerDuty customers can easily add PagerDuty events from the AWS Management Console.

Where to Learn More

Get Started with Amazon EventBridge

Intro to Event-driven Architectures and Amazon EventBridge (video, 1hr)
https://youtu.be/tvELVa9D9qU

Developer Guide
https://docs.aws.amazon.com/eventbridge/index.html

Frequently Asked Questions
https://aws.amazon.com/eventbridge/faqs/

Your modern application development journey starts with AWS Training and Certification

Training

Developing on AWS is where you will learn how to use the AWS SDK to develop secure and scalable cloud applications. We will explore how to interact with AWS using code and discuss key concepts, best practices, and troubleshooting tips.

AWS Certified Developer – Associate

Developers with one or more years of hands-on experience on AWS

This exam validates an understanding of core AWS services, uses, and basic AWS architecture best practices. Examinees must demonstrate proficiency in developing, deploying, and debugging cloud-based applications using AWS.

Visit https://www.aws.training/

AWS Certified DevOps Engineer – Professional

DevOps engineers with two or more years of experience on AWS

This exam tests an engineer's experience provisioning, operating, and managing AWS environments. Examinees will show an understanding of how to build highly scalable, available, and self-healing systems on the AWS platform and to design, manage, and maintain tools to automate operational processes.

Thank You for Attending
AWS Online Event: Modern Application Development

We hope you found it interesting! A kind reminder to **complete the survey.**
Let us know what you thought of today's event and how we can improve the event experience for you in the future.

- aws-apac-marketing@amazon.com
- twitter.com/AWSCloud
- facebook.com/AmazonWebServices
- youtube.com/user/AmazonWebServices
- slideshare.net/AmazonWebServices
- twitch.tv/aws