



serverlesspresso

What is Serverlesspresso?

Serverlesspresso is a coffee ordering app built with serverless architecture:

1. Place an order on your mobile device.
2. The order appears on the monitor and the barista's tablet app.
3. You get a notification when the drink is ready.



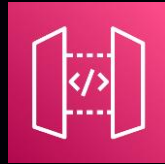
serverlesspresso

Try it out! Go to:
<https://s12d.com/coffee>

AWS Serverless services used



AWS Amplify
Console



Amazon
API Gateway



Amazon
DynamoDB



Amazon
EventBridge



AWS Step
Functions



AWS
IoT Core



AWS
Lambda



serverlesspresso

What does it cost to run this workload?

We can serve up to 960 drinks to 960 customers every day

Service	Daily cost	With Free Tier
AWS Amplify Console	\$0.28	Free
Amazon API Gateway	\$0.01	Free
Amazon Cognito	\$0.00	Free
Amazon DynamoDB	\$0.01	Free
Amazon EventBridge	\$0.01	Free
AWS IoT Core	\$0.01	Free
AWS Lambda	\$0.01	Free
Amazon SNS	\$7.98	\$7.98
AWS Step Functions	\$0.29	Free

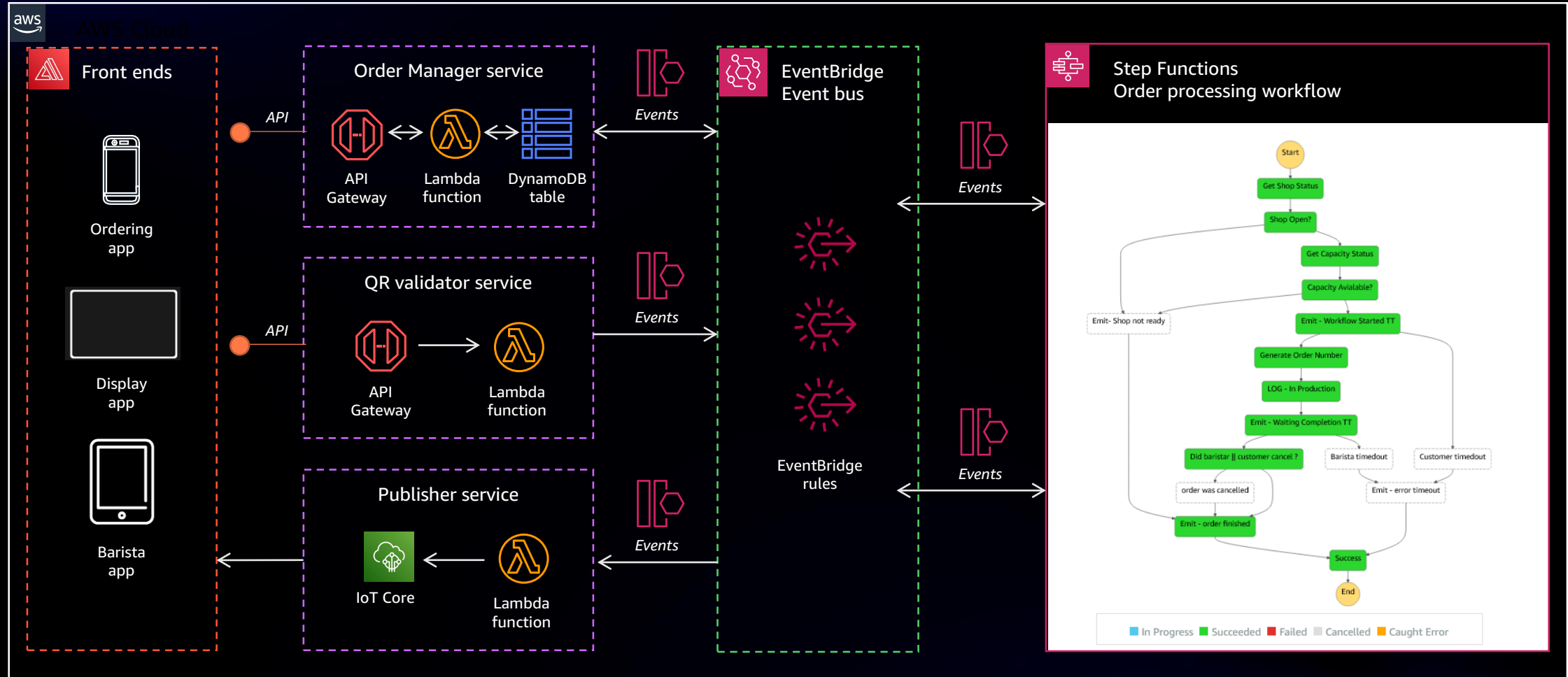


serverlesspresso

Learn about the AWS Free Tier:
<https://aws.amazon.com/free>

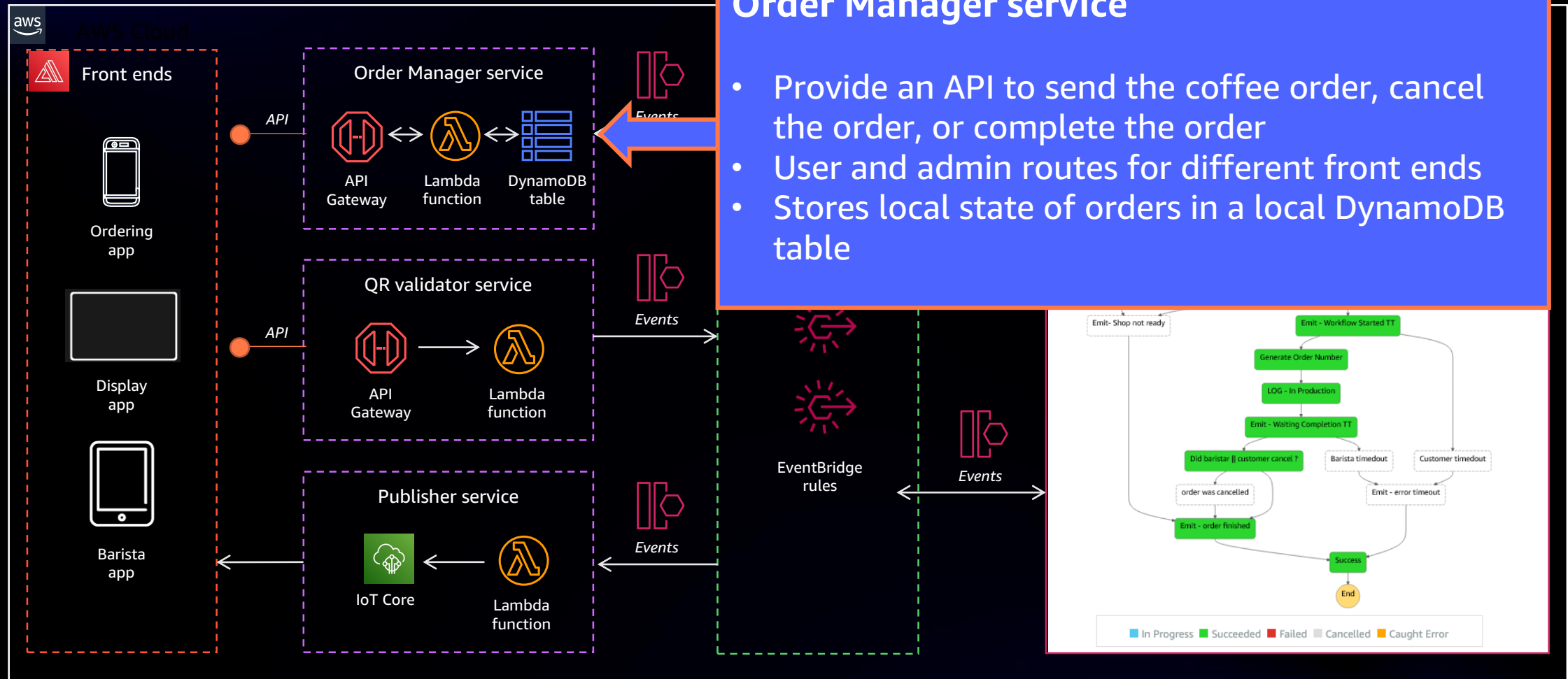
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How it works



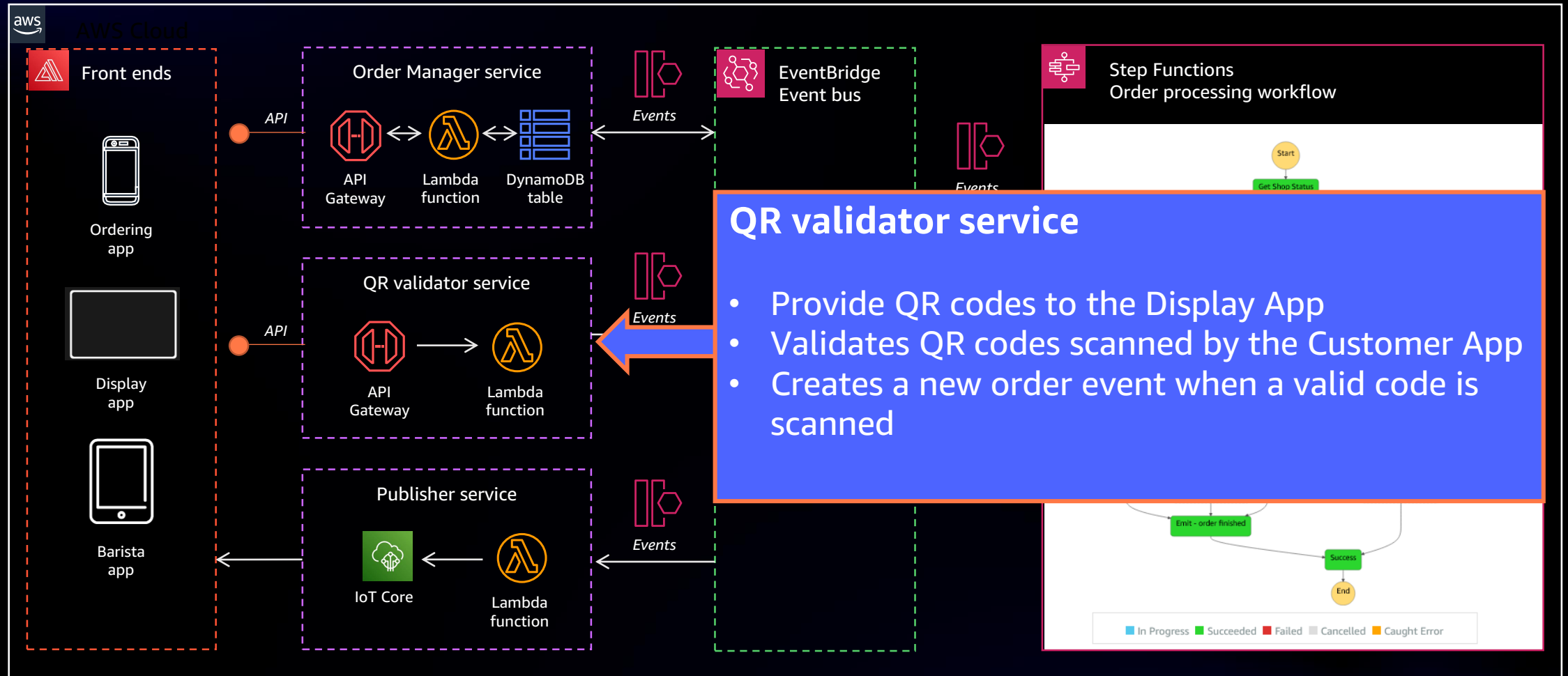
serverlesspresso

How it works



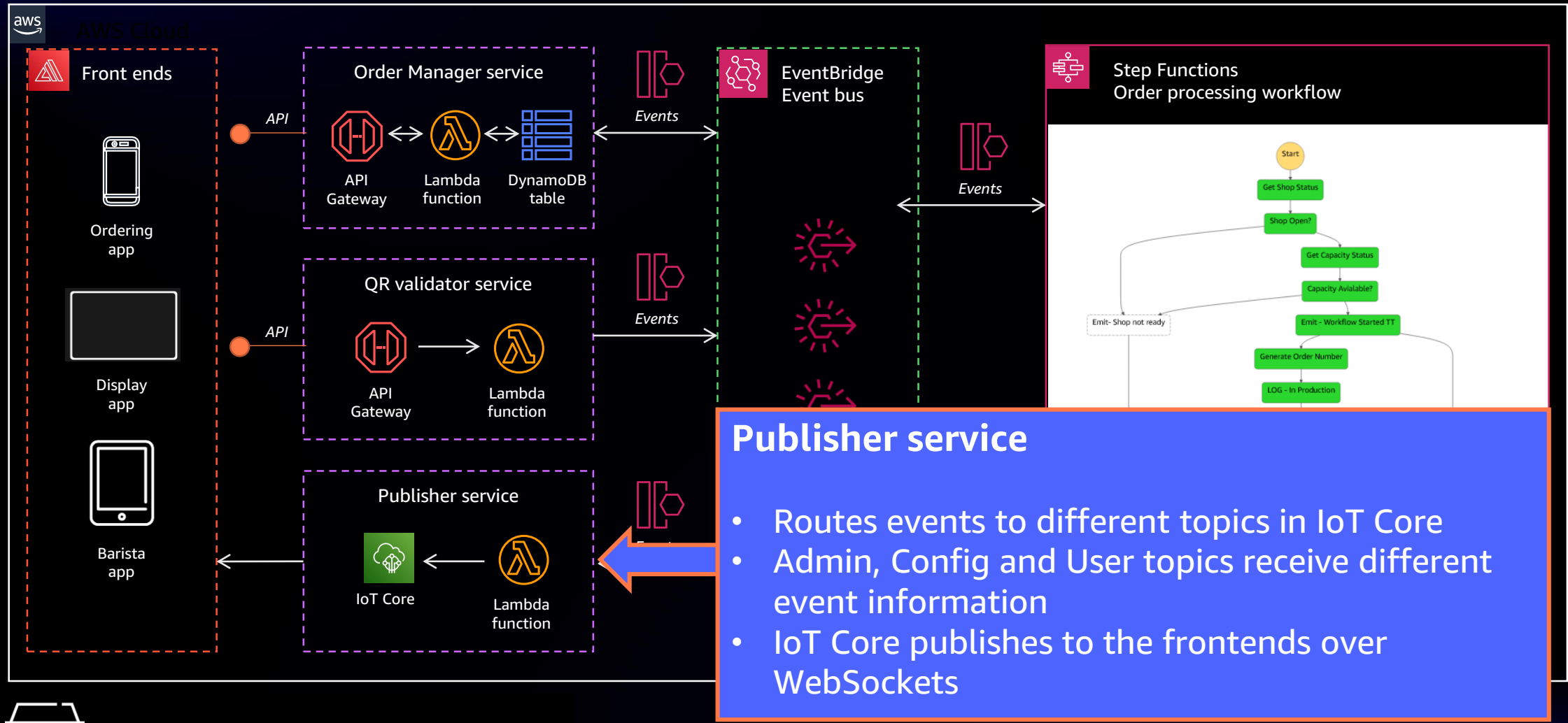
serverlesspresso

How it works



serverlesspresso

How it works



serverlesspresso

The frontend applications

VUE.JS APPS HOSTED WITH AWS AMPLIFY CONSOLE

1. Ordering app:

- Loads menu
- Scans barcode to get order token
- Sends order
- Displays updates

2. TV display app:

- Shows current barcode
- Receives order status updates
- Listens for store open/close events

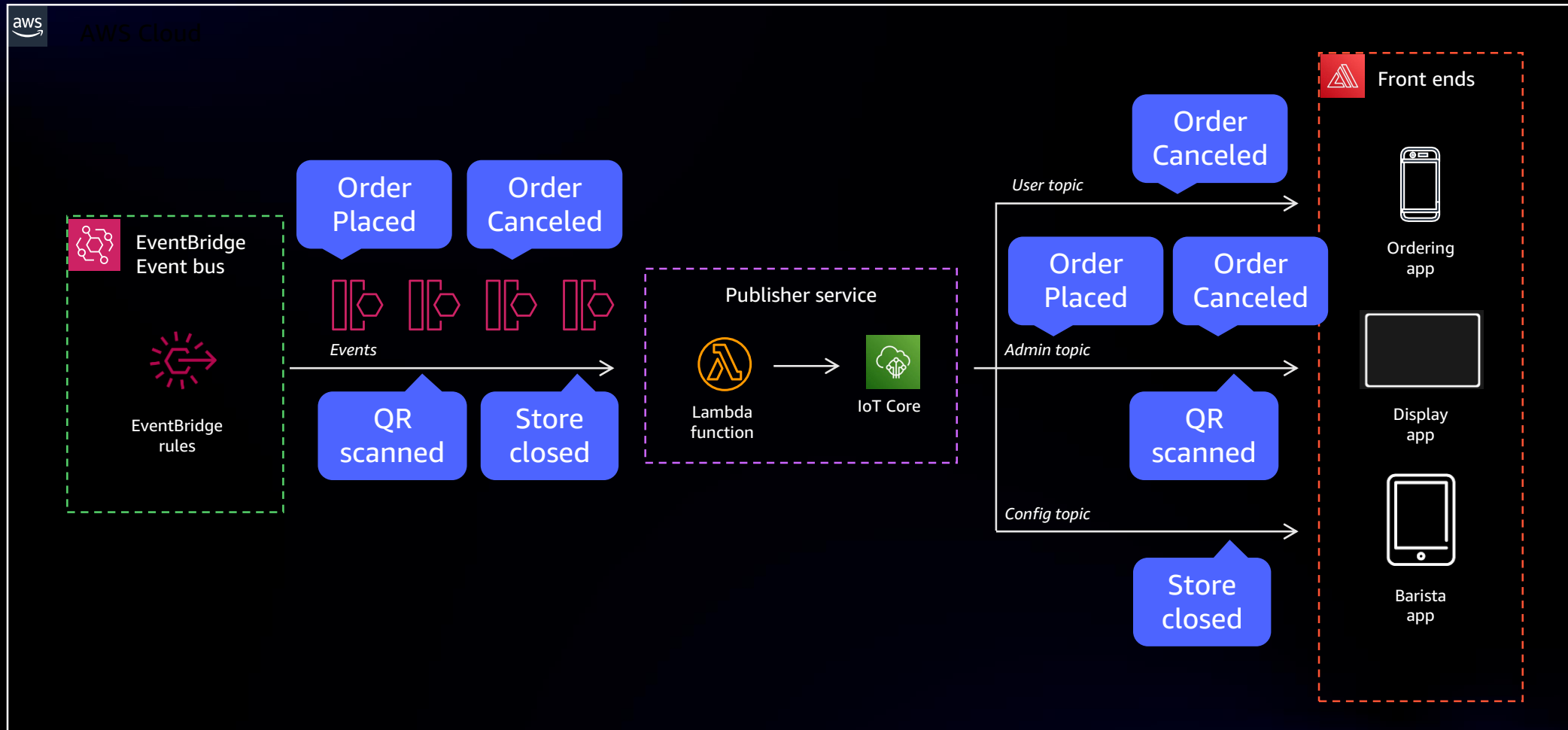
3. Barista app:

- Shows incoming orders
- Enables order completion or cancellation
- Enables store open/close events



serverlesspresso

Realtime updates with AWS IoT Core

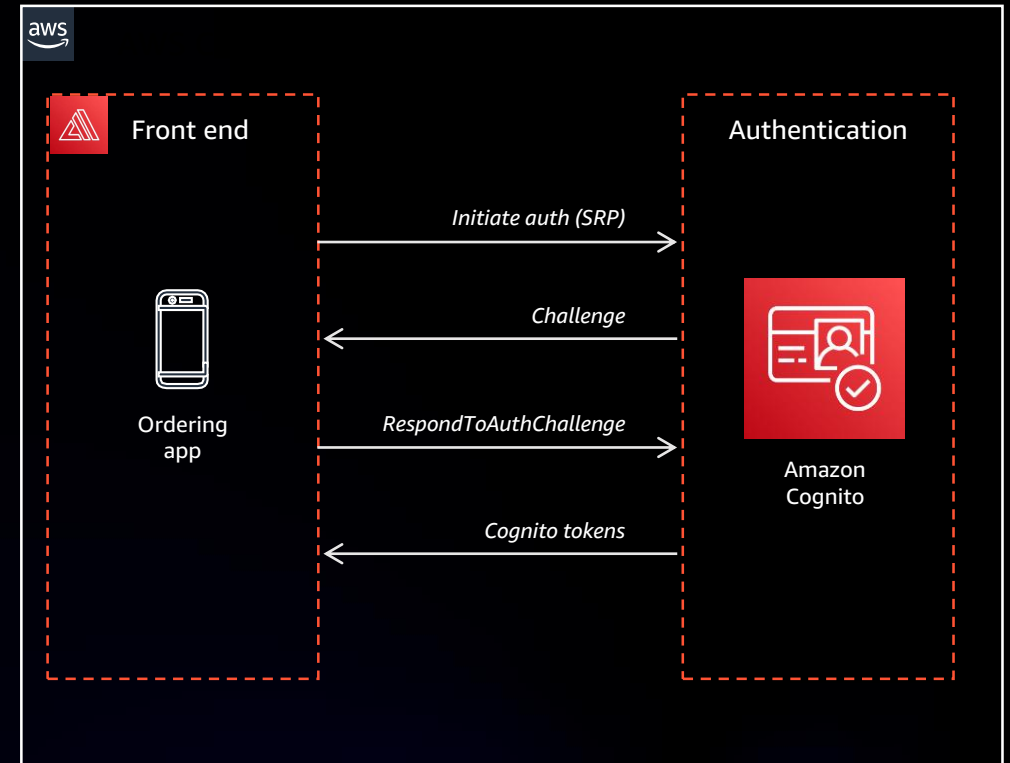


serverlesspresso

Authorization and authentication

SMS-BASED ONE-TIME PASSWORDS WITH AMAZON COGNITO

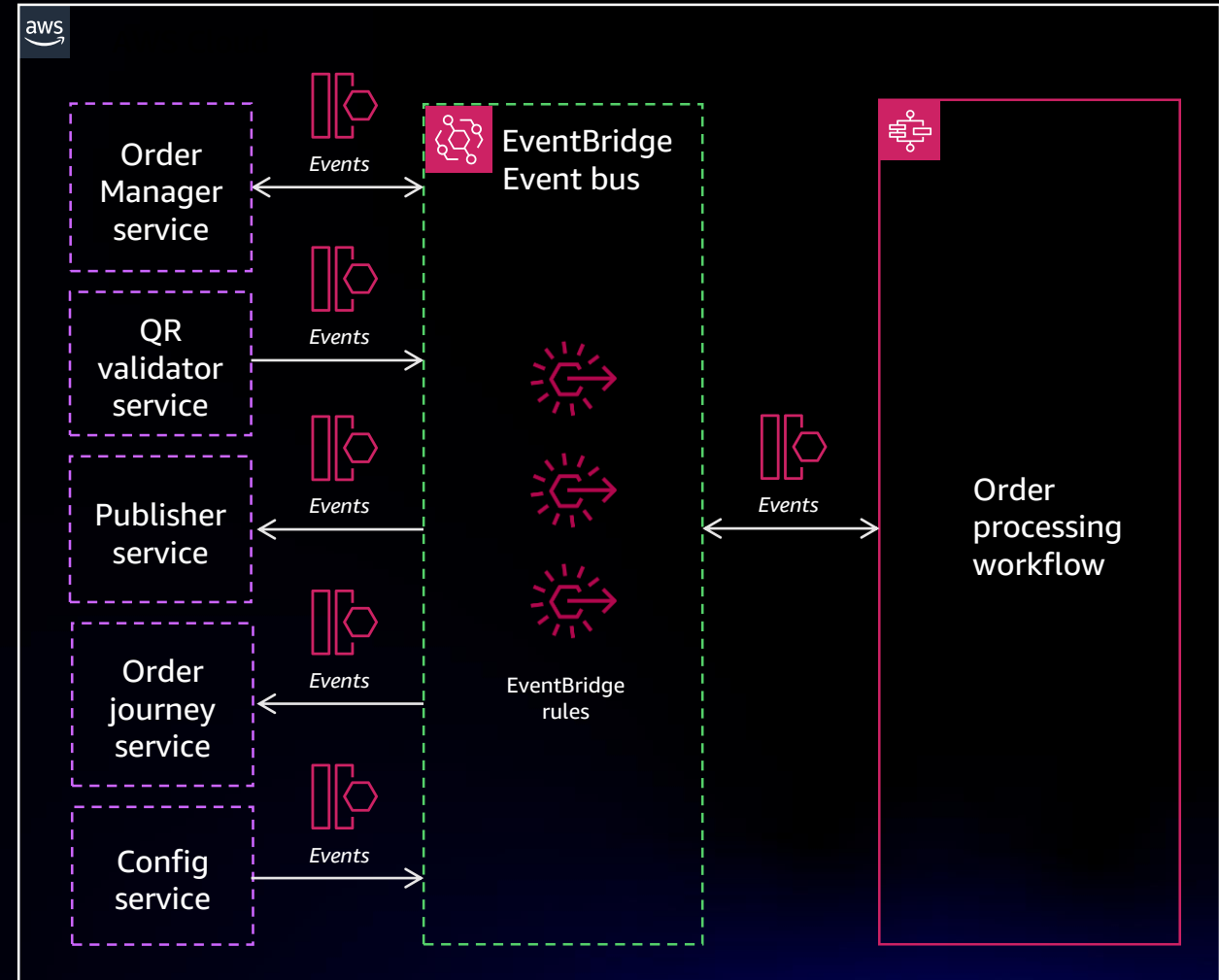
- Uses Lambda triggers in Cognito to create a custom authentication flow
- SMS messages with random codes are sent via Amazon SNS
- Frontends use Amplify auth component to simplify JWT token refresh process
- Frontends present JWT to backend APIs and services



Event-driven architectures

USING AMAZON EVENTBRIDGE FOR CHOREOGRAPHING MICROSERVICES

- Event flow drives the application
- Events choreograph the services, while Step Functions orchestrates the transactions
- Add new microservices as event consumers without changing existing code
- Microservices emit events independently of consumers



serverlesspresso

What is an event?

USING AMAZON EVENTBRIDGE FOR CHOREOGRAPHING MICROSERVICES

- An event is defined in JSON
- "Detail" is application specific
- Envelope attributes are provided by Amazon EventBridge
- Producers create events
- Consumers choose which events to listen to by using rules

```
{
  "version": "0",
  "id": "6ac4e27b-1234-1234-1234-5fb02c880319",
  "detail-type": "OrderProcessor.OrderStarted",
  "source": "awsserverlessda.serverlesspresso",
  "account": "123456789012",
  "time": "2021-11-28T13:12:30Z",
  "region": "us-west-2",
  "detail": {
    "userId": "jbesw",
    "orderId": "eYmAfqLD67vlbdUViLe_D",
    "drinkOrder": {
      "icon": "barista-icons_cafe-latte",
      "modifiers": [],
      "drink": "Latte"
    }
  }
}
```



serverlesspresso

Design the workflow...

DEFINE ALL THE STEPS IN MAKING A DRINK

Process for making a drink

- 1) Check the store is open
- 2) Get barista capacity
- 3) Wait for the customer order – cancel if > 5 mins
- 4) Generate an order number
- 5) Wait for barista to make drink – cancel if > 15 mins
- 6) Also handle cancelation by customer or barista



... then design visually with Workflow Studio

REPLACE SPAGHETTI CODE WITH STATE MACHINES


Step Functions Workflow Studio [Info](#)


Search


Actions


Flow


MOST POPULAR

 AWS Lambda Invoke


 Amazon SNS Publish


 Amazon ECS RunTask


 AWS Step Functions StartExecution


 AWS Glue StartJobRun


COMPUTE

 Amazon Data Lifecycle ...

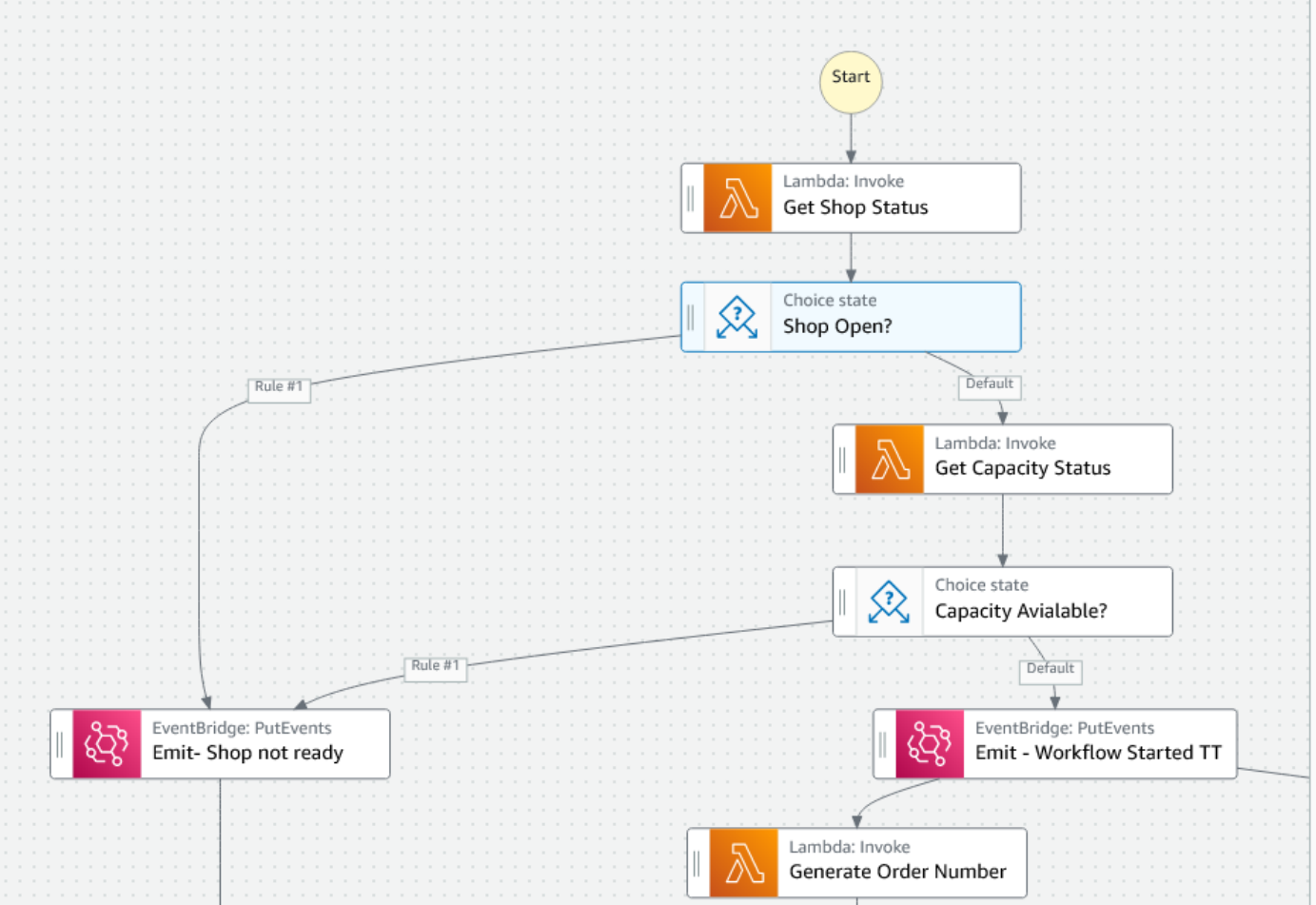
 Amazon EBS

 Amazon EC2

 AWS EC2 Instance Conn...

 Elastic Inference

Undo Redo Zoom in Zoom out Center



```
graph TD; Start((Start)) --> GetStatus[Lambda: Invoke Get Shop Status]; GetStatus --> ChoiceOpen[Choice state Shop Open?]; ChoiceOpen -- Rule #1 --> PutEventsNotReady[EventBridge: PutEvents Emit- Shop not ready]; ChoiceOpen -- Default --> GetCapacity[Lambda: Invoke Get Capacity Status]; GetCapacity --> ChoiceCapacity[Choice state Capacity Available?]; ChoiceCapacity -- Rule #1 --> PutEventsNotReady; ChoiceCapacity -- Default --> PutEventsTT[EventBridge: PutEvents Emit - Workflow Started TT]; PutEventsTT --> GenerateOrder[Lambda: Invoke Generate Order Number];
```

Cancel **Apply and exit**

Export

Form

Definition

Shop Open?

Configuration

Input

Output

State name

Shop Open?

State type

Choice

Choice Rules

Choice rules let you create if-then logic to determine which state the workflow should transition to next.

Rule #1

not(\$.StoreOpen.modified.storeOpen == true)

Default rule

Defines default state when no rule evaluates to true

+ Add new choice rule

Comment - optional

check if Capacity is available

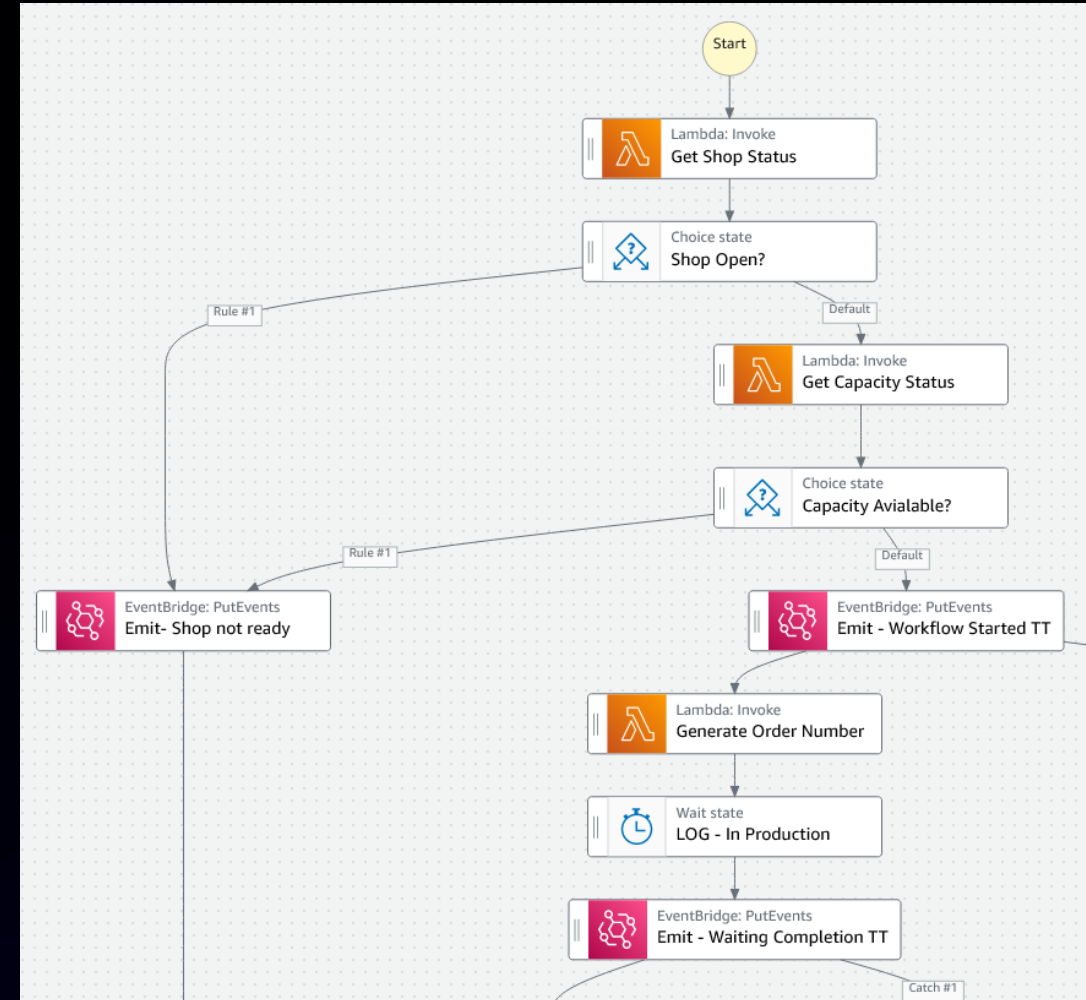
Managing each coffee's journey

USING AWS STEP FUNCTIONS TO MANAGE EACH WORKFLOW EXECUTION

- Workflow ensures store is open and baristas have capacity
- Allows customer 5 minutes to order before timing out / cancelling
- Allows barista 15 minutes to make before timing out / cancelling
- Uses Lambda functions for custom logic; uses direct service integrations otherwise



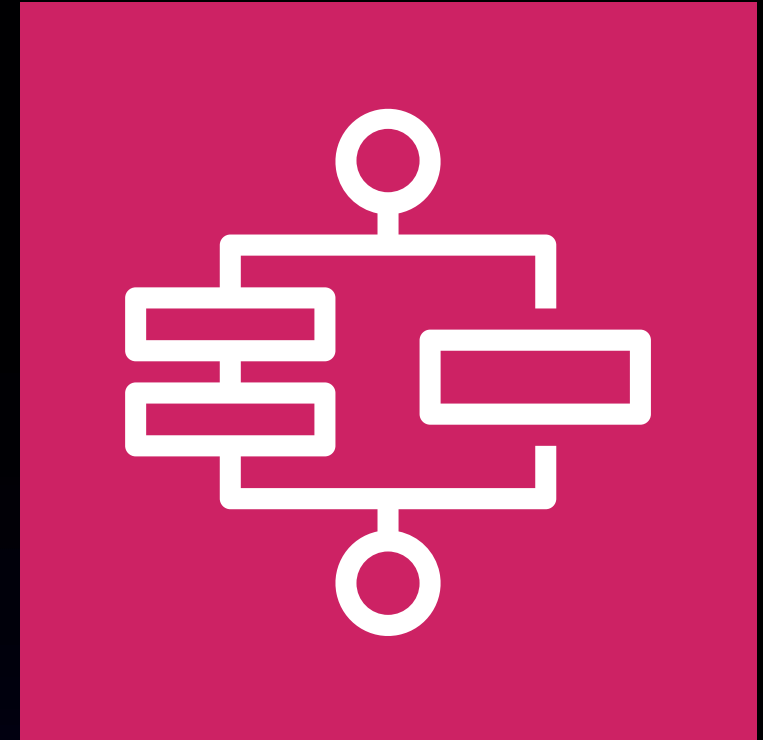
serverlesspresso



AWS Step Functions

FULLY MANAGED STATE MACHINES ON AWS

- Resilient workflow automation
- Built-in error handling
- Powerful AWS service integration
- First-class support for integrating with your own services
- Auditable execution history and visual monitoring





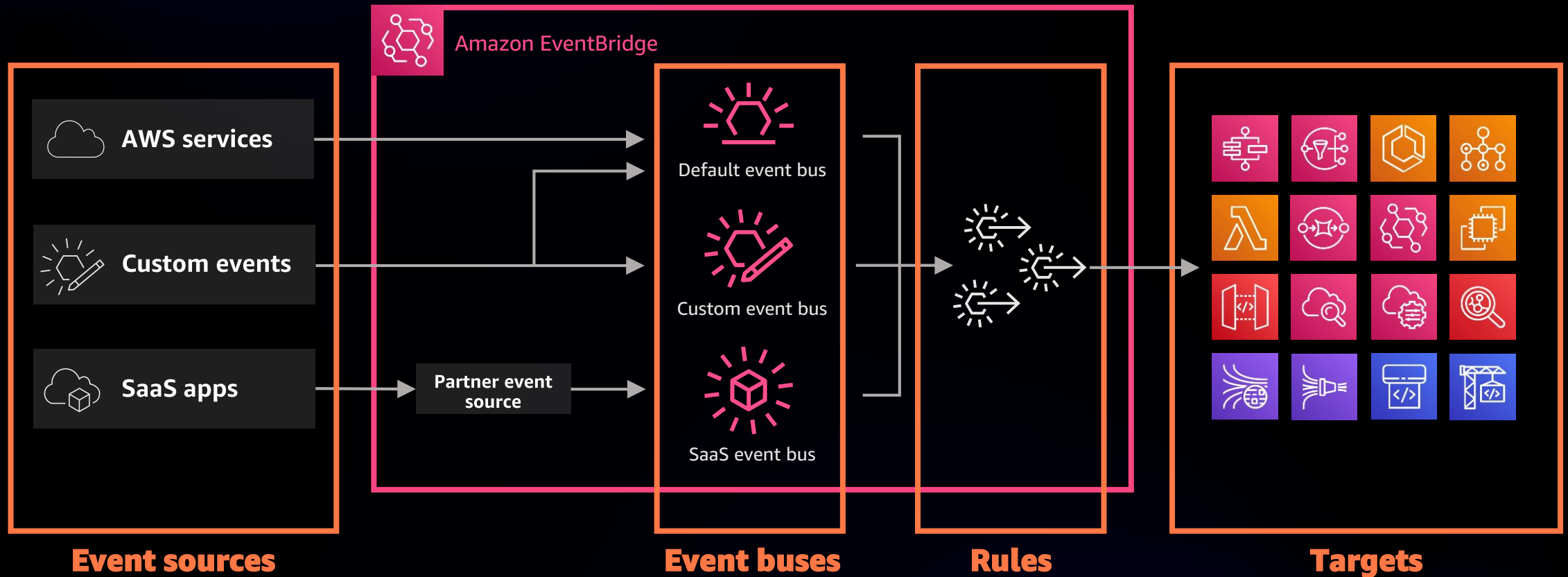
Amazon EventBridge

A serverless event bus service for AWS services, your own applications, and SaaS providers

- Serverless—pay only for the events you process
- Simplified scaling avoids increasing costs to sustain and manage resources
- No upfront investments, ongoing licensing, or maintenance costs
- No specialist knowledge needed

Amazon EventBridge architecture

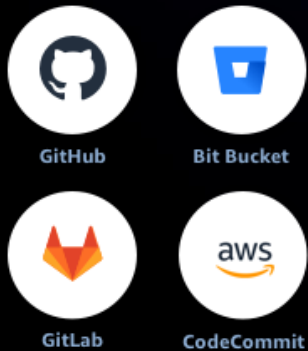
ROUTING EVENTS BETWEEN MICROSERVICES



AWS Amplify Console

HOSTING WITH AN INTEGRATED CI/CD WORKFLOW

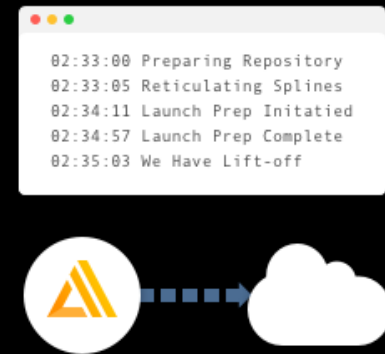
Build, deploy, and host cloud-powered modern web apps



1. Connect your repository



2. Configure build settings



3. Deploy your app

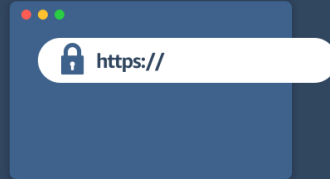
Optional deployment of backend resources +
fully managed front-end hosting

AWS Amplify Console



Globally available

Your app is served via Amazon's reliable content delivery network with 144 points of presence globally.



Easy custom domain setup

Set up custom domains managed in Amazon Route 53 with a single click, plus get a free HTTPS certificate.



Simplified continuous workflows

Connect your repository to 'git push' changes to your front end and backend in a single workflow.



Feature branch deployments

Work on new features without impacting production. Create branch deployments linked to each feature branch.



Atomic deployments

All deployments either rollout successfully or fail without requiring maintenance windows.



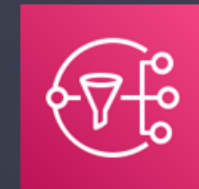
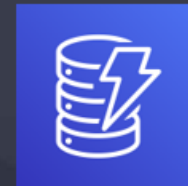
Password protection

Share yet-to-be released features with internal stakeholders by setting a username and password.

[New](#)[Blogs](#)[Videos](#)[Learn](#)[Events ▾](#)[Patterns](#)[About](#)

Welcome to Serverless Land

This site brings together all the latest blogs, videos, and training for AWS Serverless. Learn to use and build apps that scale automatically on low-cost, fully-managed serverless architecture.

[Learn More](#)

For more serverless learning resources, visit:

<https://serverlessland.com>