

The background features a dark blue gradient on the left side, transitioning into a vibrant, multi-colored geometric design on the right. This design consists of overlapping triangular and quadrilateral shapes in shades of purple, magenta, and red, creating a dynamic, abstract pattern. A thin, light-colored vertical line runs through the center of the image, and a horizontal line is positioned near the top.

# AWS re:Invent

DECEMBER 1 - 5, 2025 | LAS VEGAS, NV

CNS381

# Secure Multi-tenant SaaS with AWS Lambda: A Tenant Isolation Deep Dive



**Anton Aleksandrov**  
Pr. Solutions Architect, Serverless  
AWS



**Bill Tarr**  
Pr. Partner Solutions Architect,  
AWS



**Joe**

COMICS

1

APR

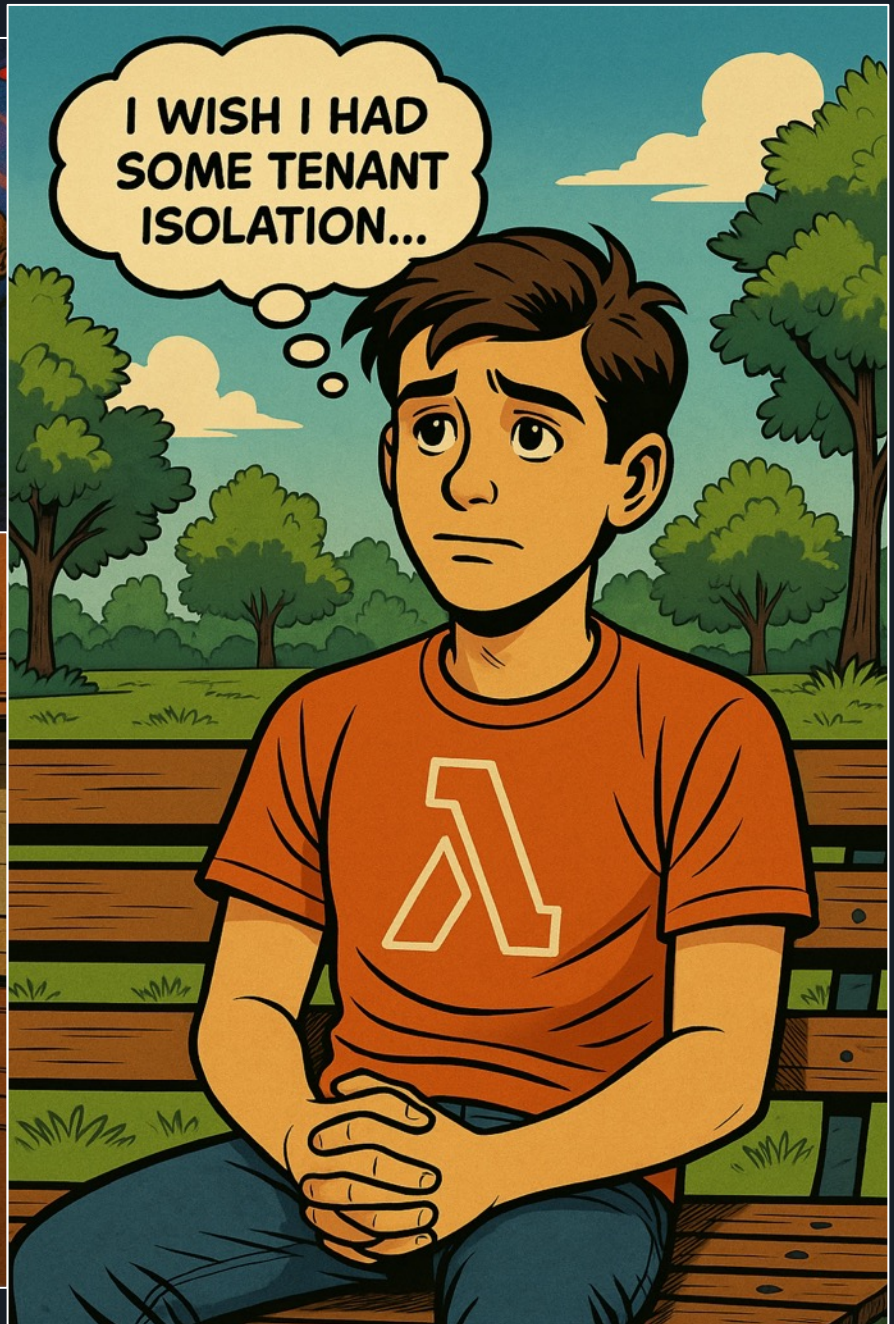
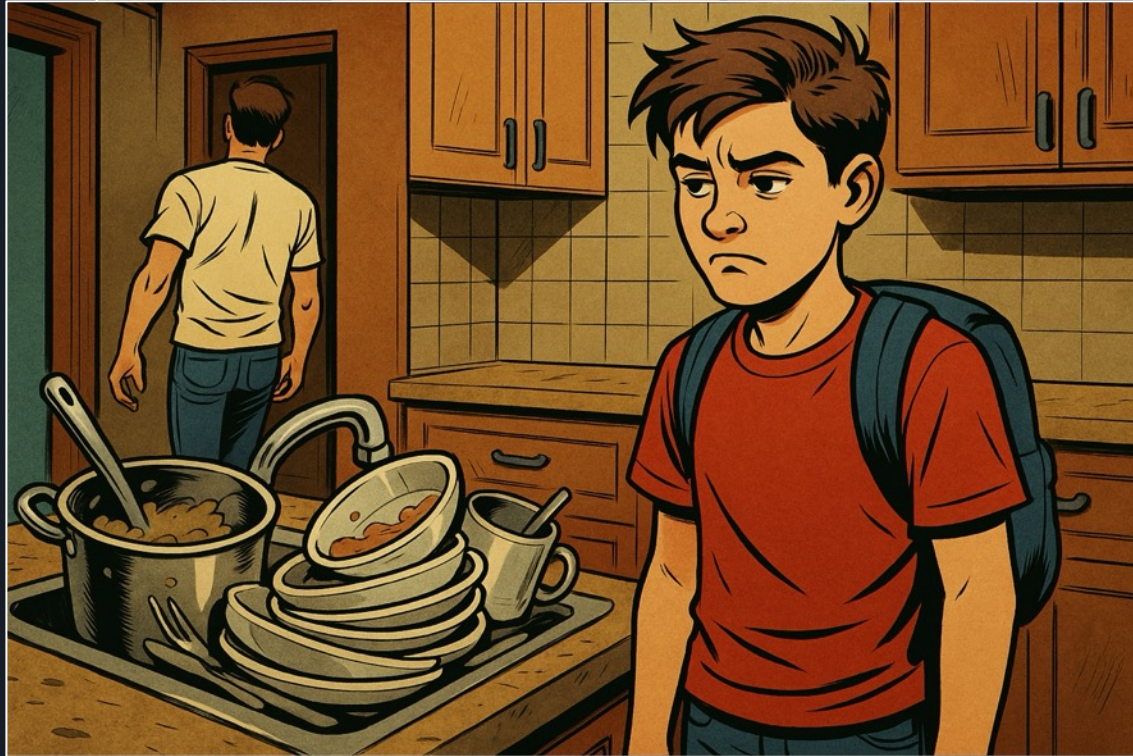
35¢

# MEET JOE



INTRODUCING  
A NEW HERO!







**Single-Tenant**  
(aka dedicated or siloed)



**Multi-Tenant**  
(aka shared or pooled)



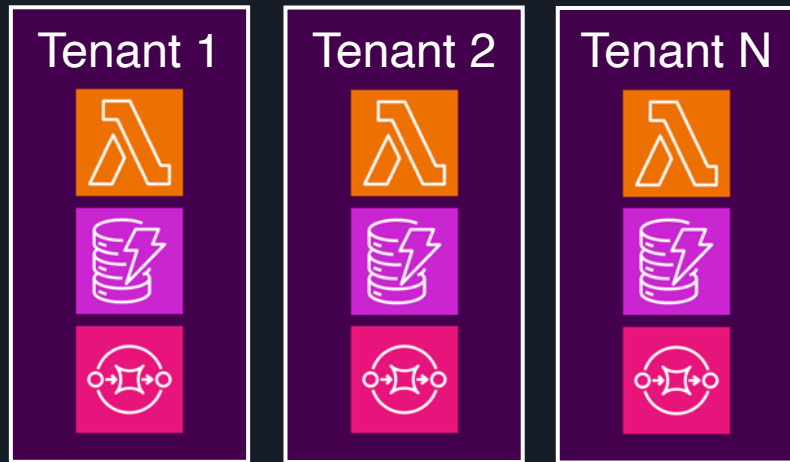
## Single-Tenant (aka dedicated or siloed)



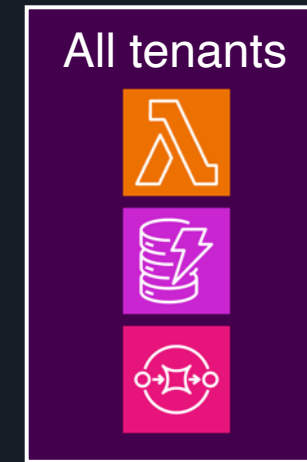
## Multi-Tenant (aka shared or pooled)

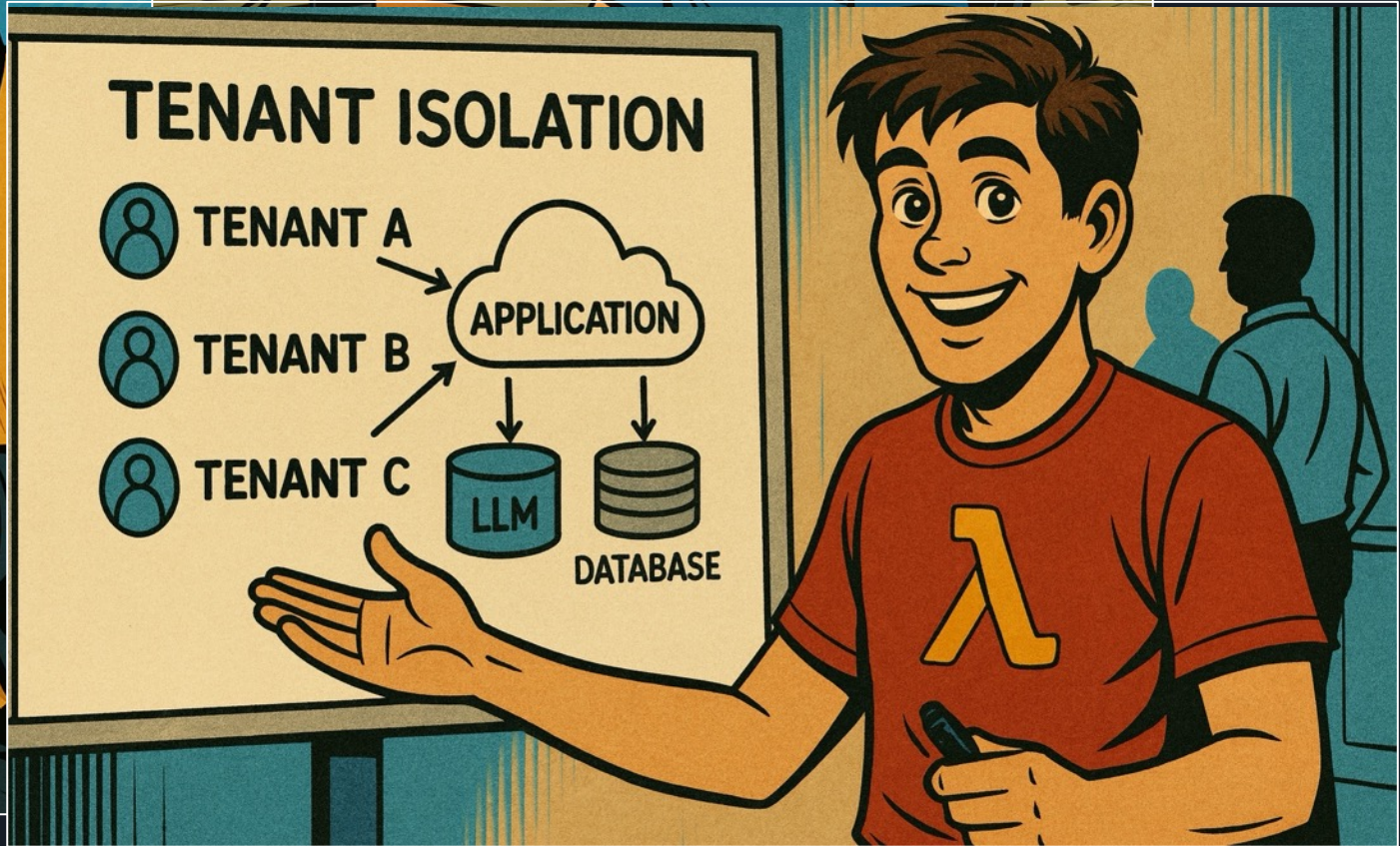
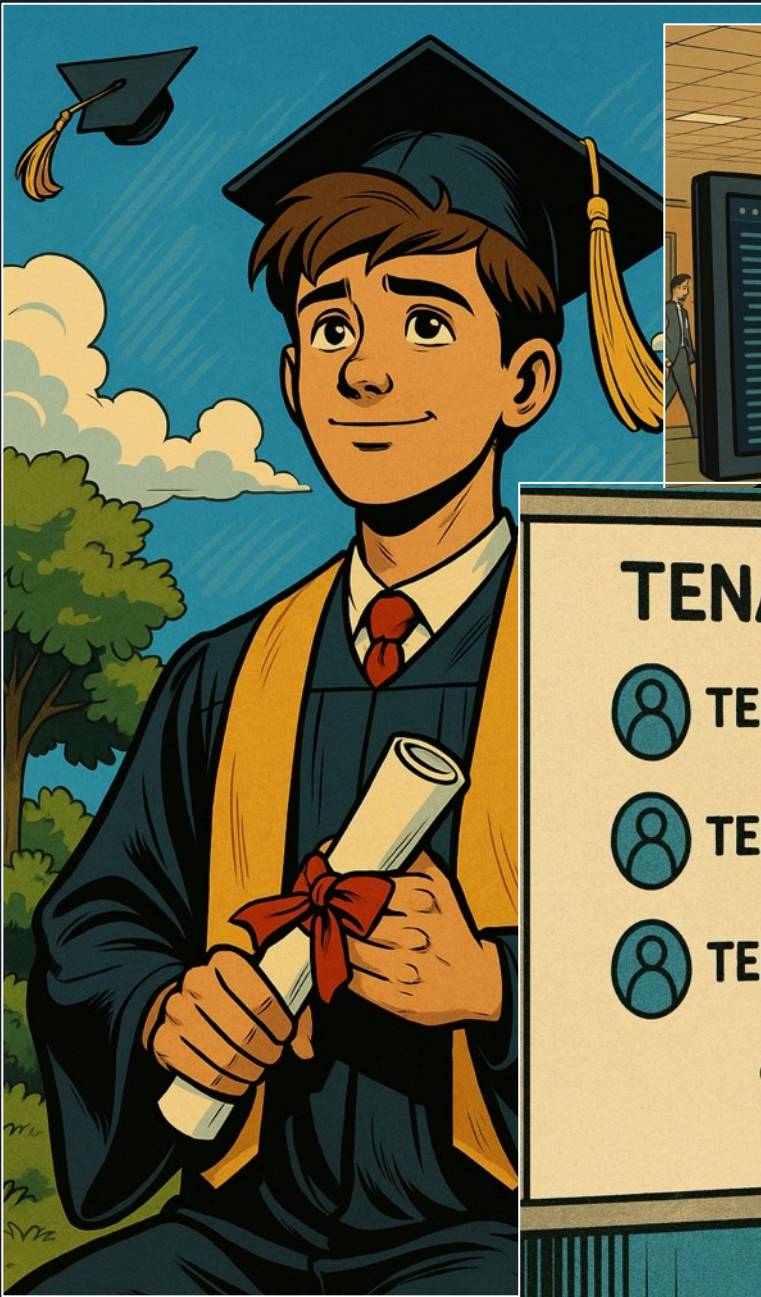


## Single-Tenant (aka dedicated or siloed)



## Multi-Tenant (aka shared or pooled)





Building multi-tenant SaaS apps can be tricky



# Common SaaS challenges

Technology requirements



Business requirements



# Common SaaS challenges

## Technology requirements

- Isolate tenant compute
- Avoid noisy neighbors

## Business requirements

- Innovate faster
- Stay ahead of competitors



# Serverless enables you to focus on business value



**Reduced operational overhead**



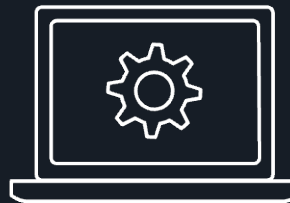
**Improved availability and scalability**



**Improved security posture**



**Native integrations**



**Flexible compute choice**



**Accelerated application delivery**

# Why Serverless is a great fit for SaaS

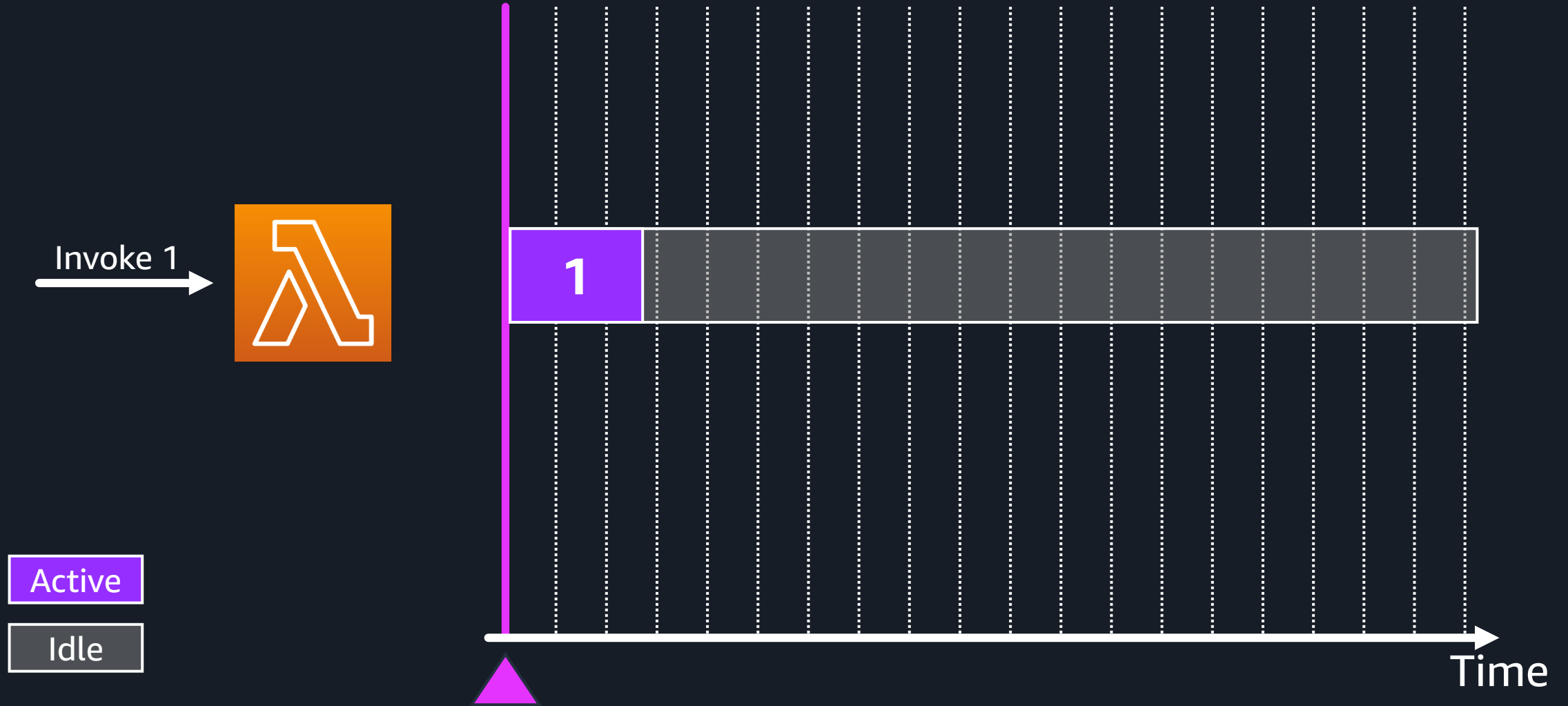


**Using serverless allows you to have the best capacity cost to capacity utilization ratio.**

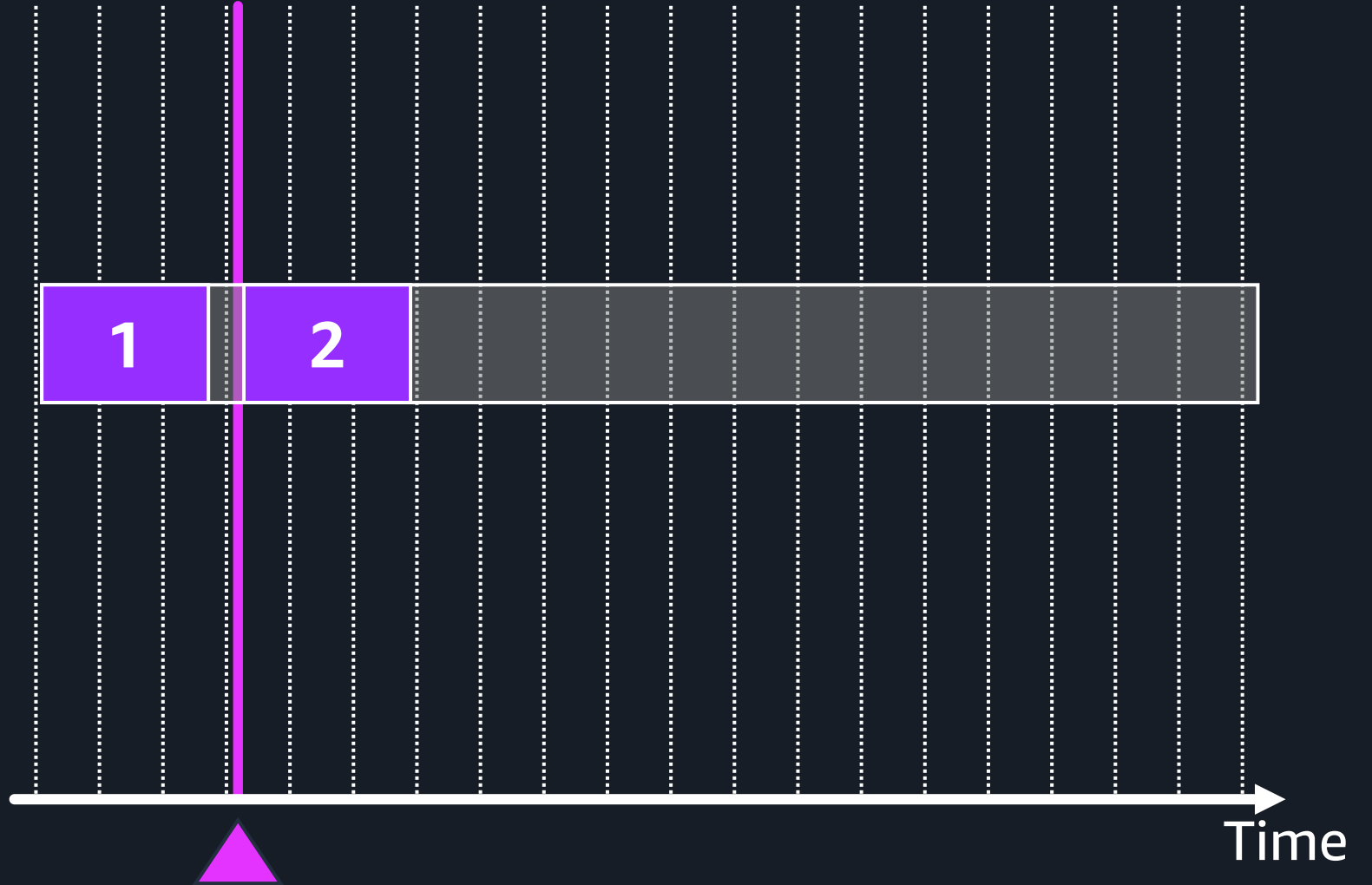
How does Lambda  
scale execution  
environments?



# Understanding Lambda concurrency



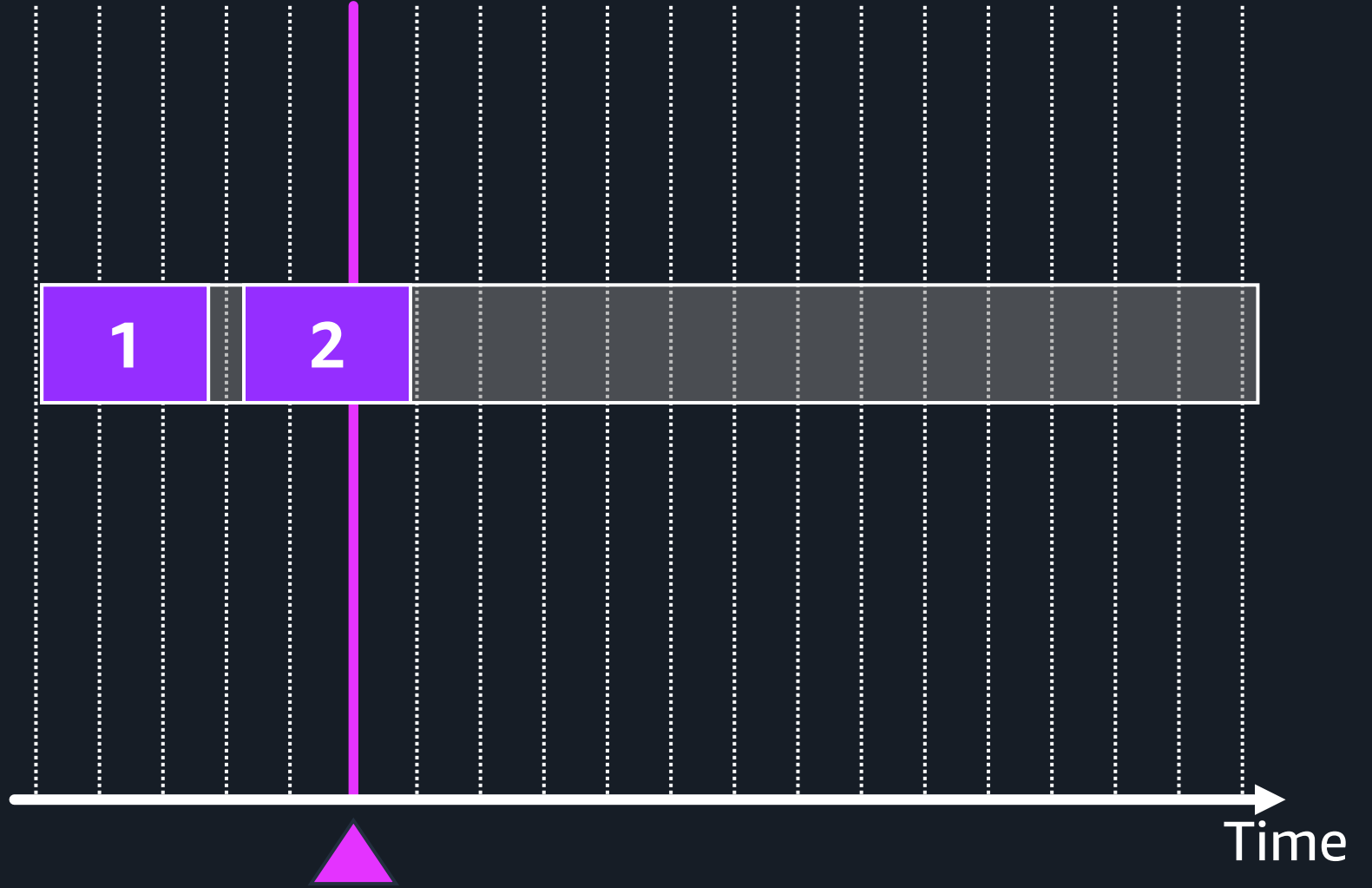
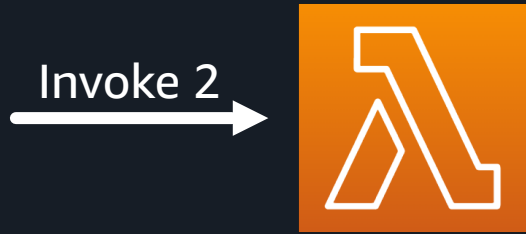
# Understanding Lambda concurrency



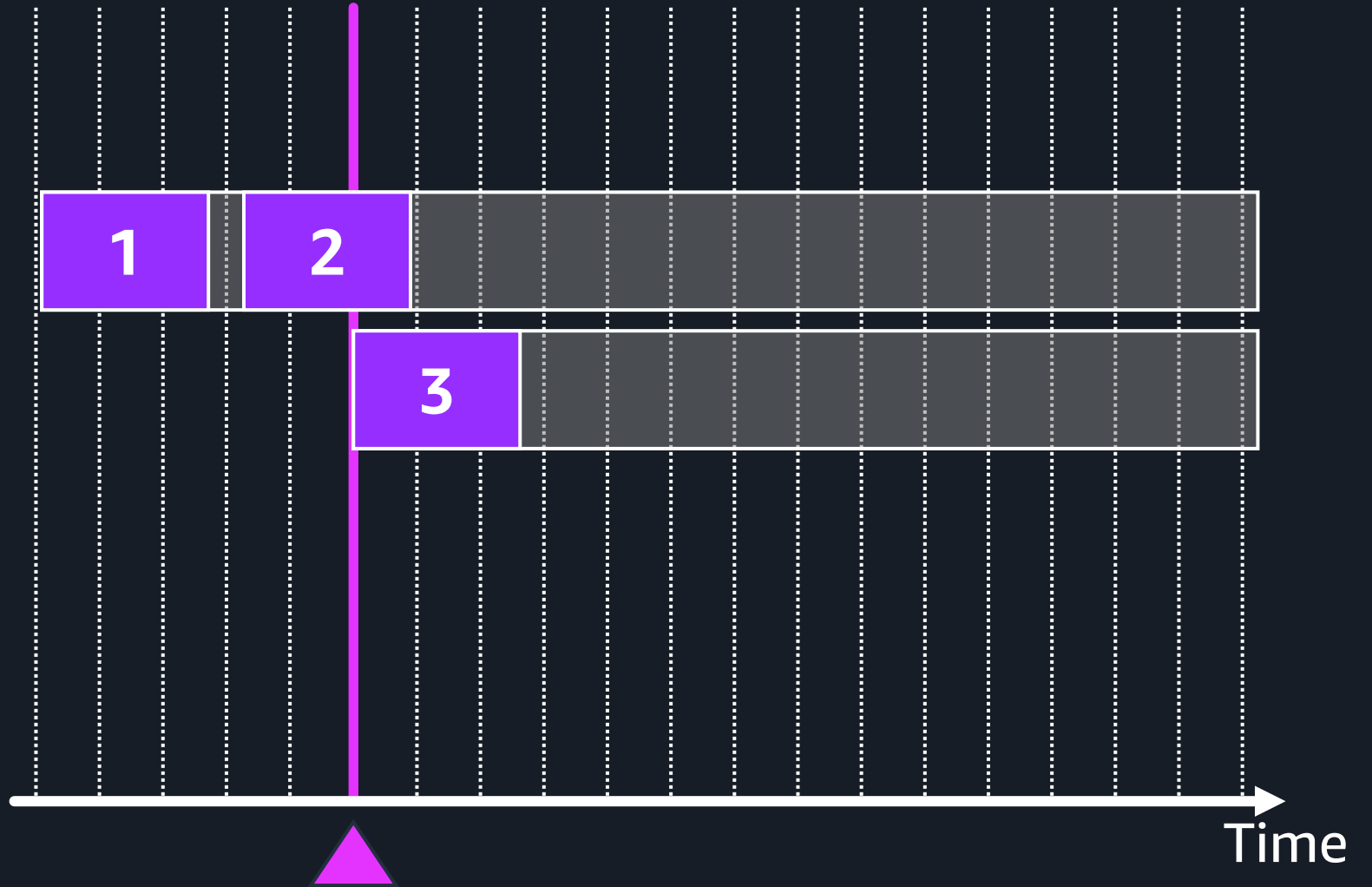
Active

Idle

# Understanding Lambda concurrency



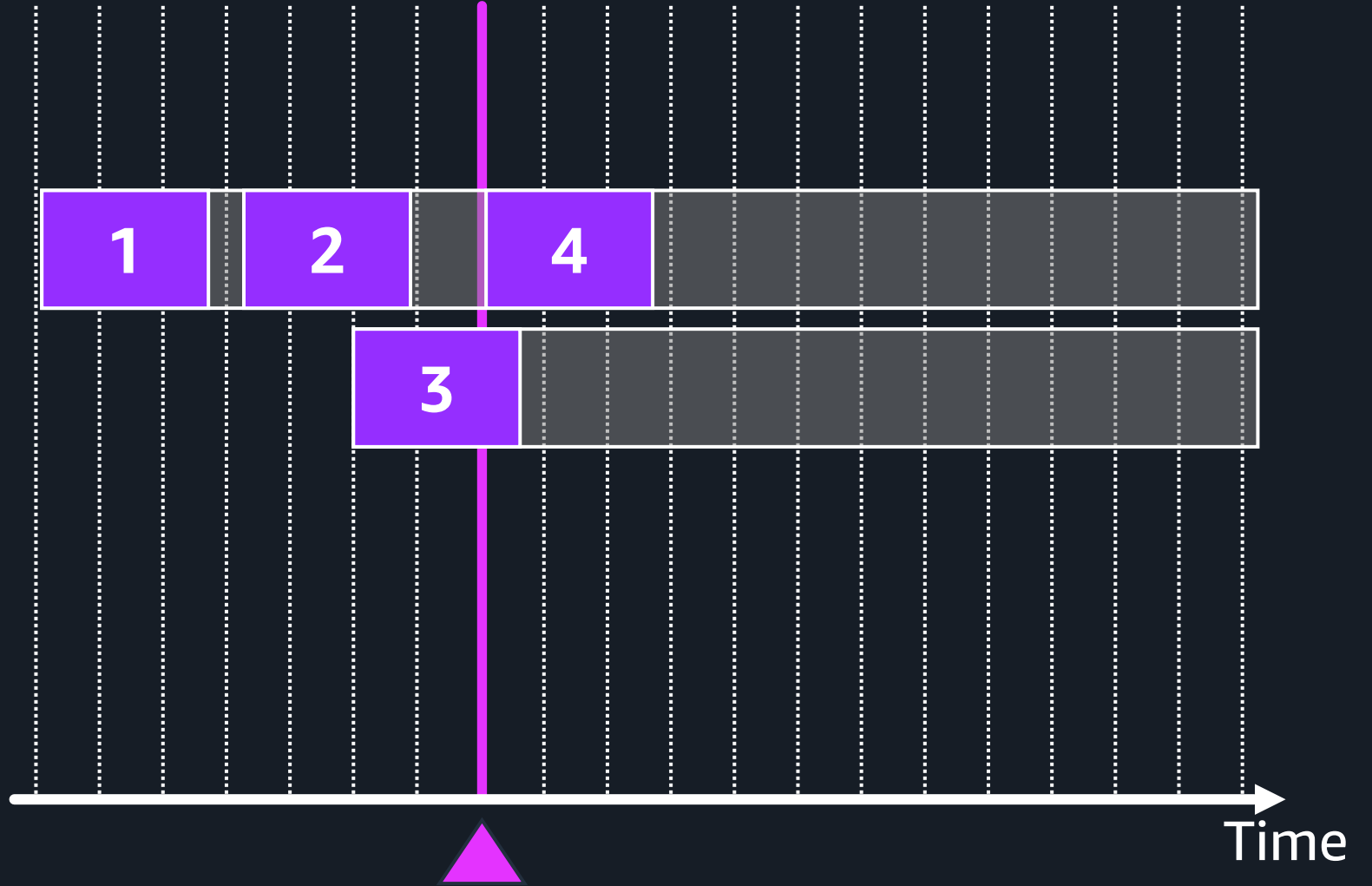
# Understanding Lambda concurrency



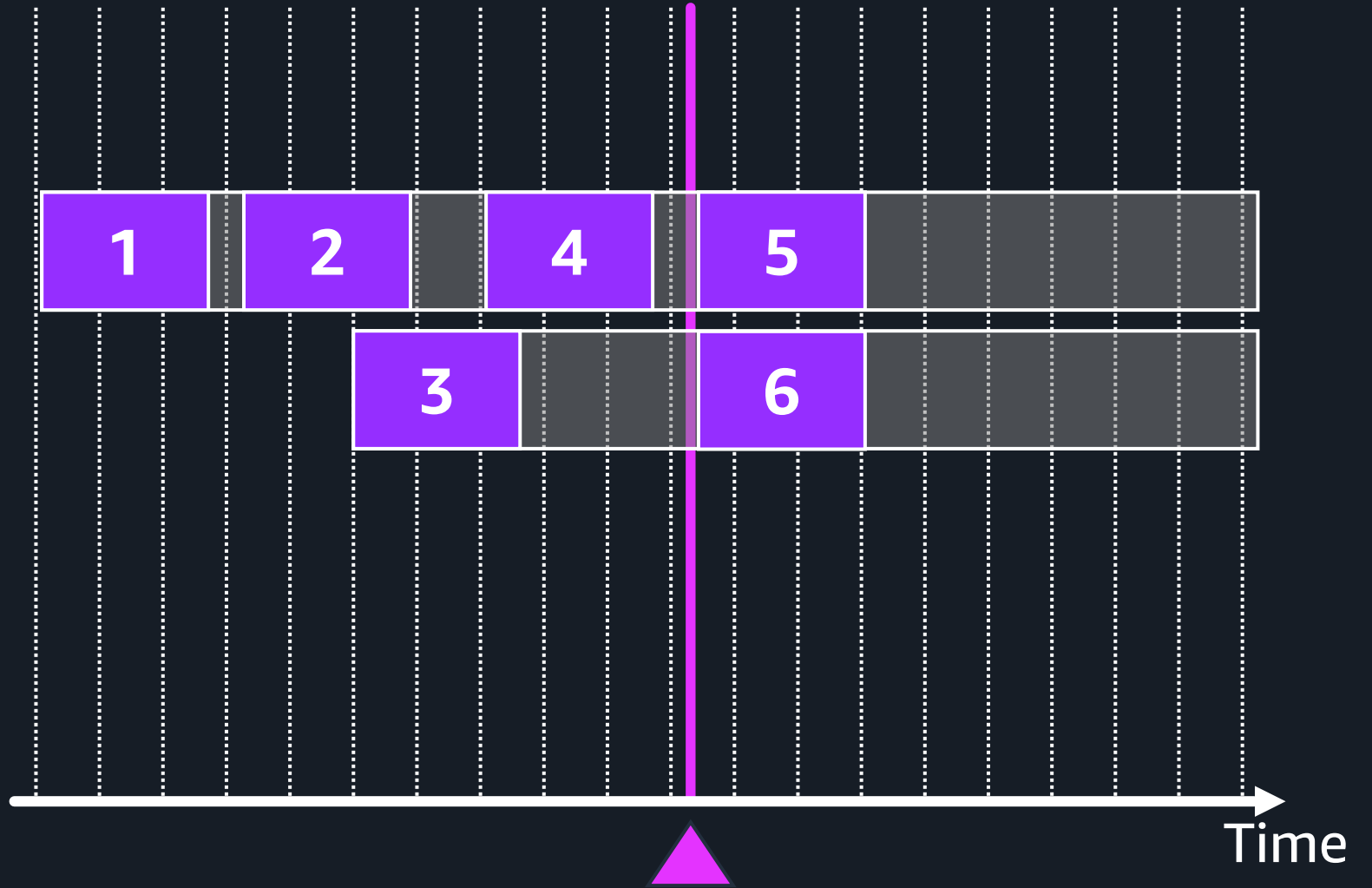
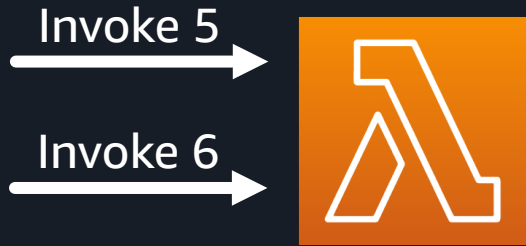
Active

Idle

# Understanding Lambda concurrency



# Understanding Lambda concurrency



# Understanding Lambda concurrency

Invoke 7

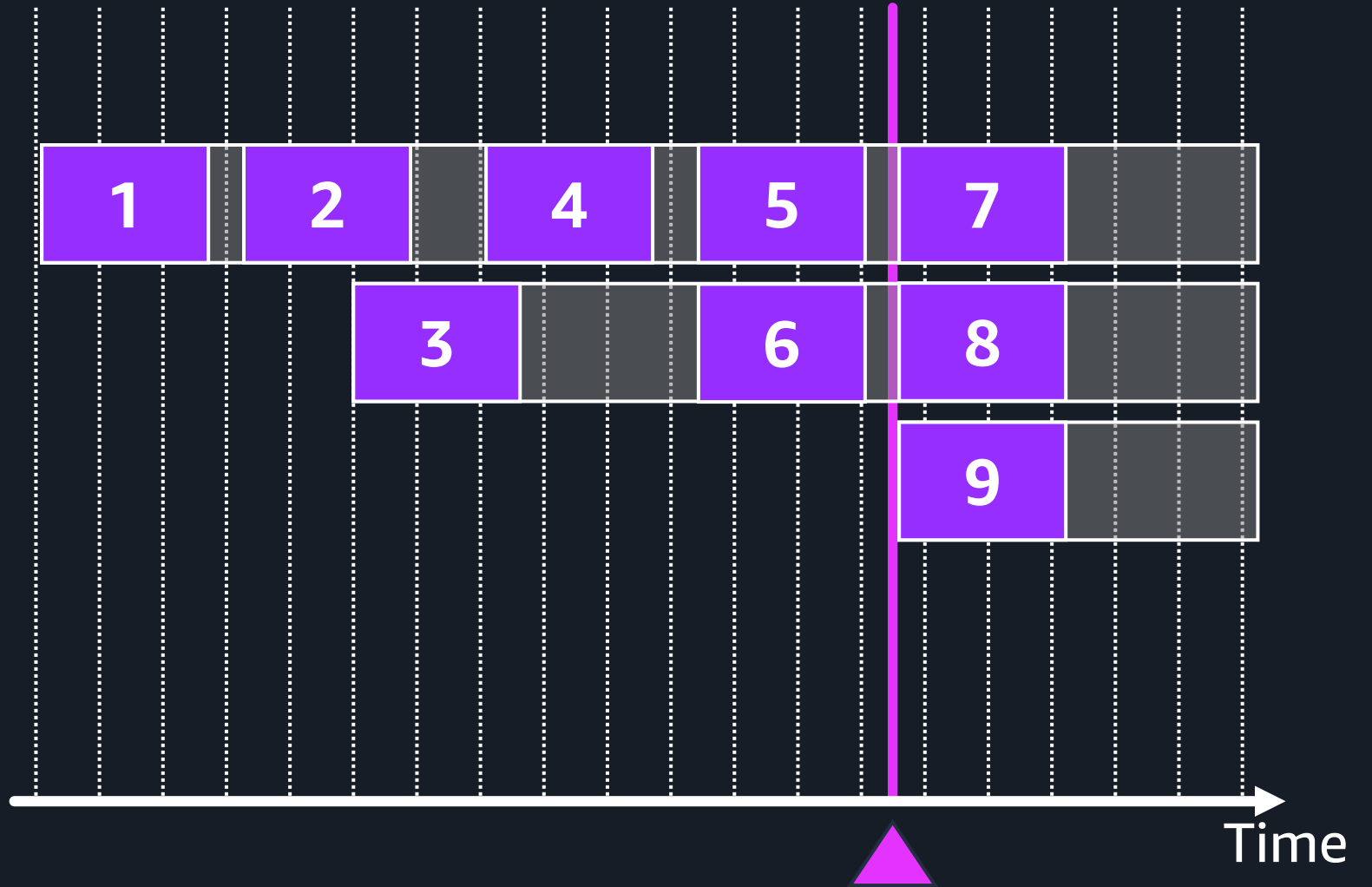
Invoke 8

Invoke 9



Active

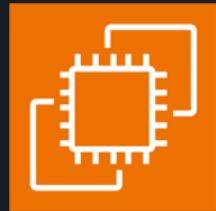
Idle



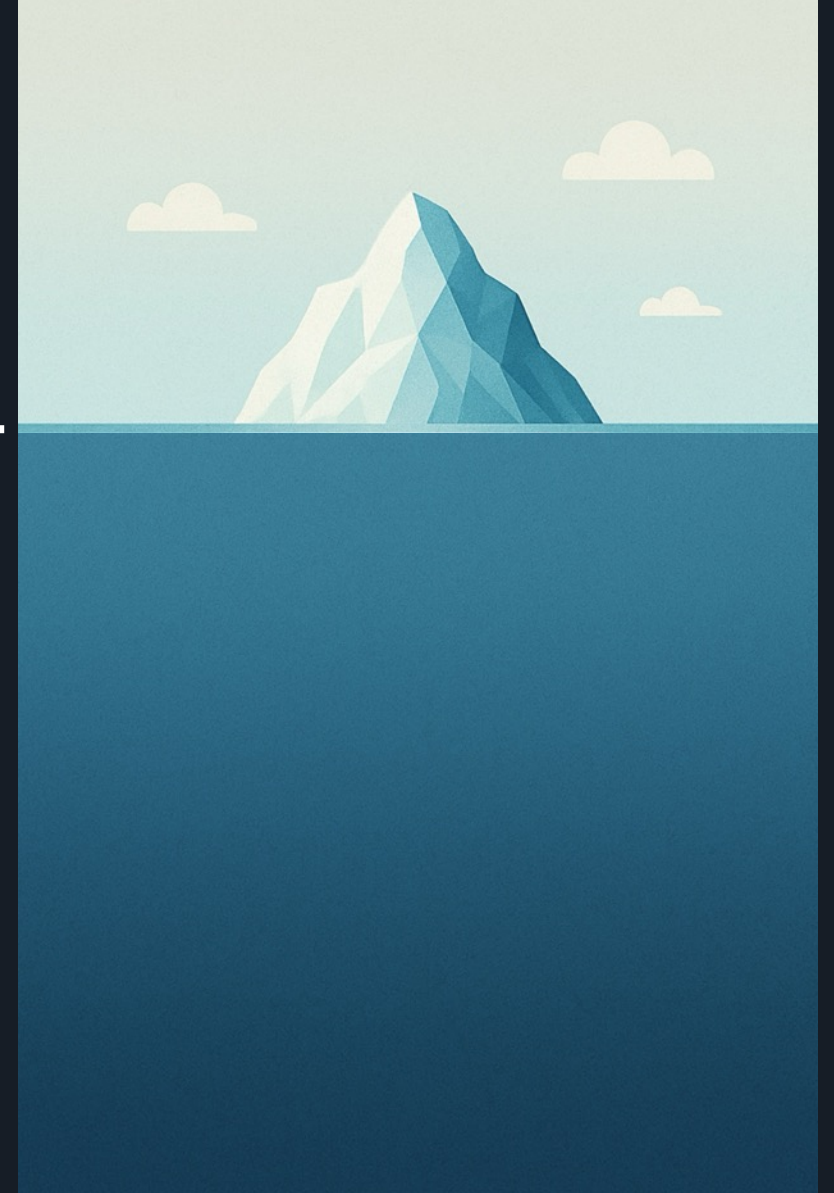
# The biggest secret of Serverless



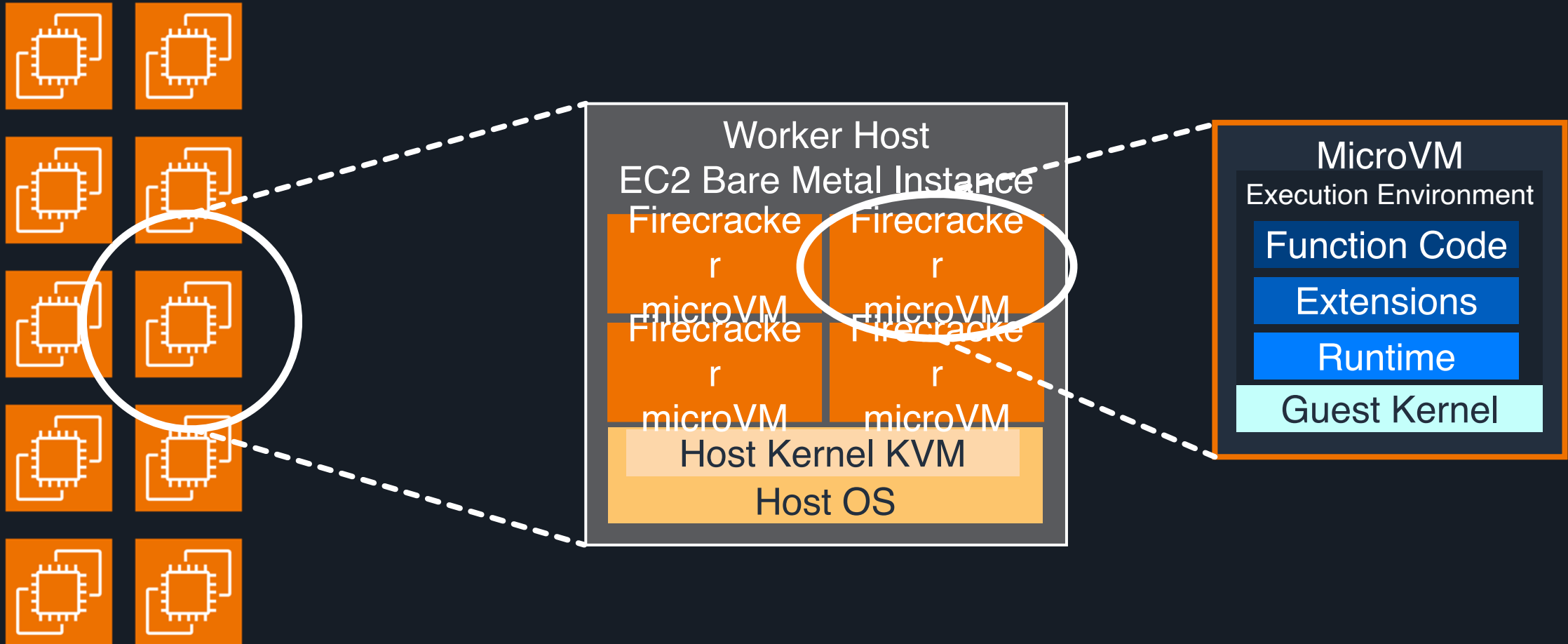
(and many other serverless services)



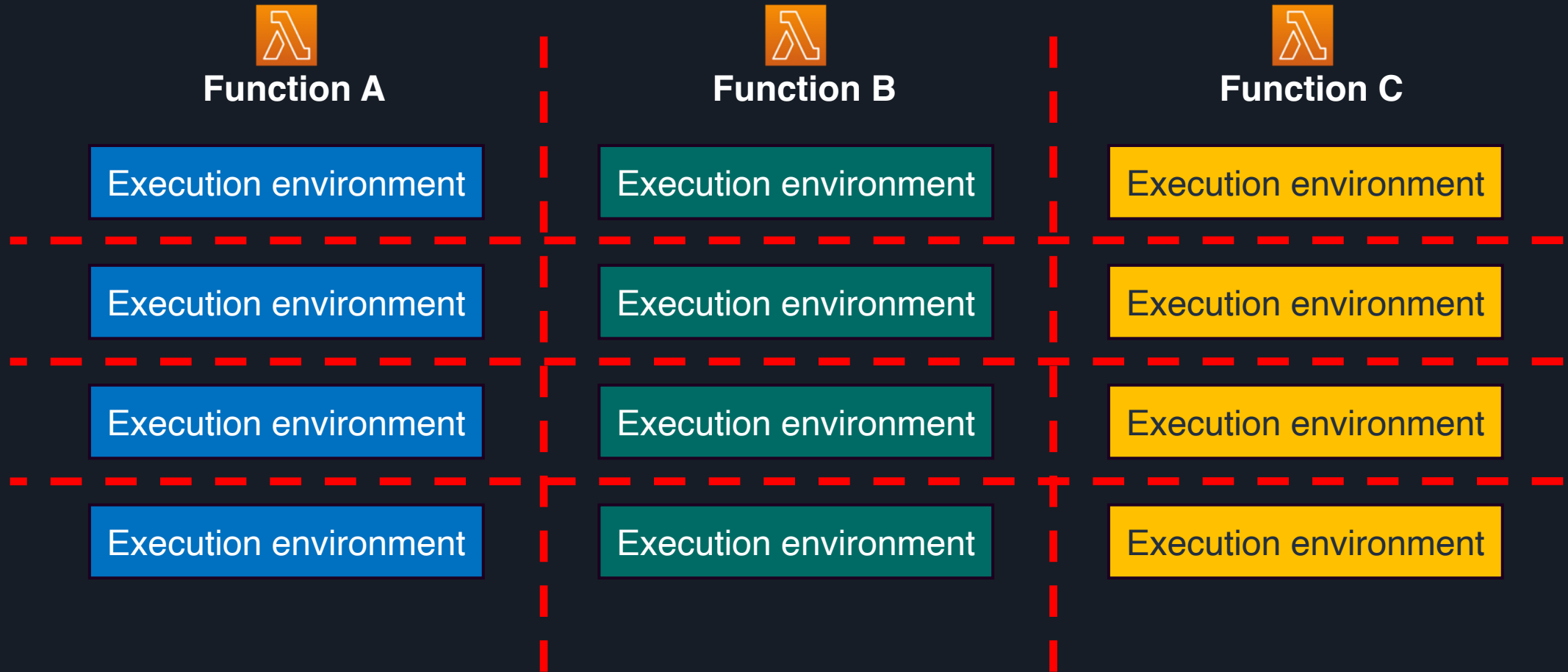
Hundreds of thousands of  
EC2 instances under-the-hood



# Lambda under-the-hood



# Lambda isolation boundaries

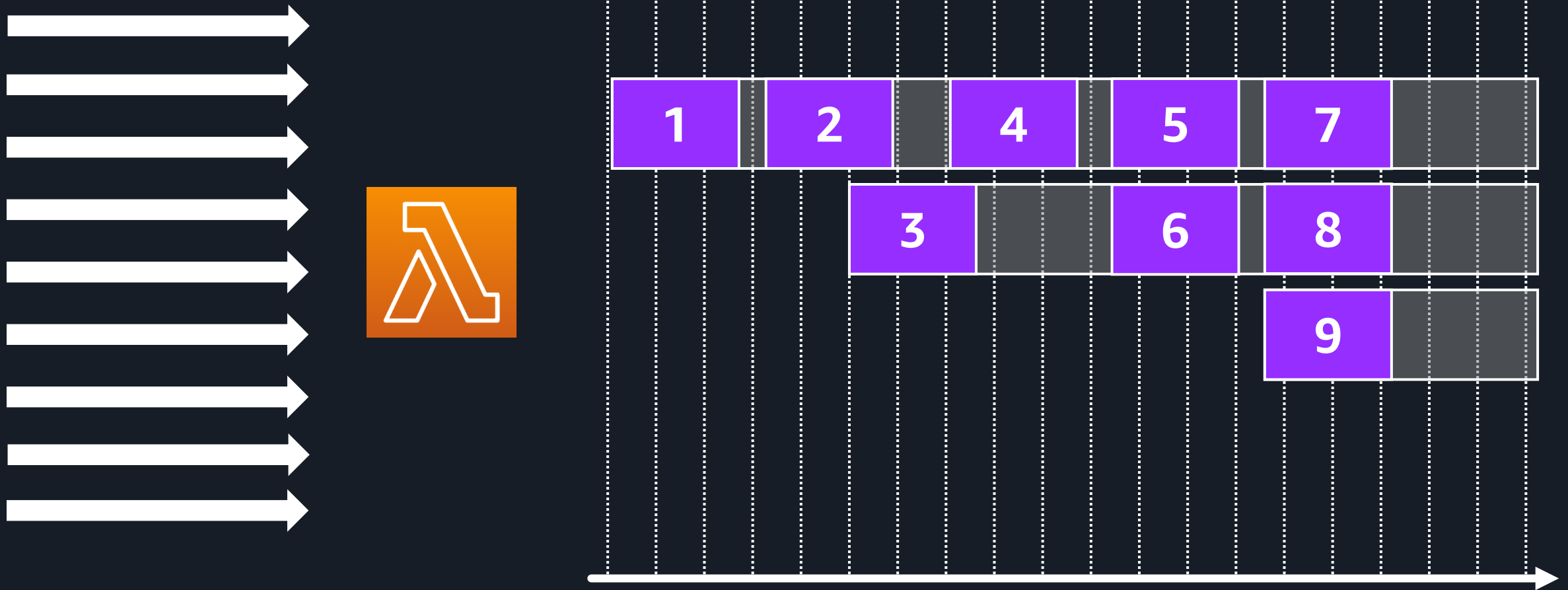


# What is shared across execution environments

	Execution environments across <b>different functions</b>	Execution environments that belong to the <b>same function</b>
vCPU	✗	✗
Memory	✗	✗
Network bandwidth	✗	✗
Virtual disk	✗	✗
Environment variables	✗	✓
IAM execution role / permissions	✗	✓
Code	✗	✓

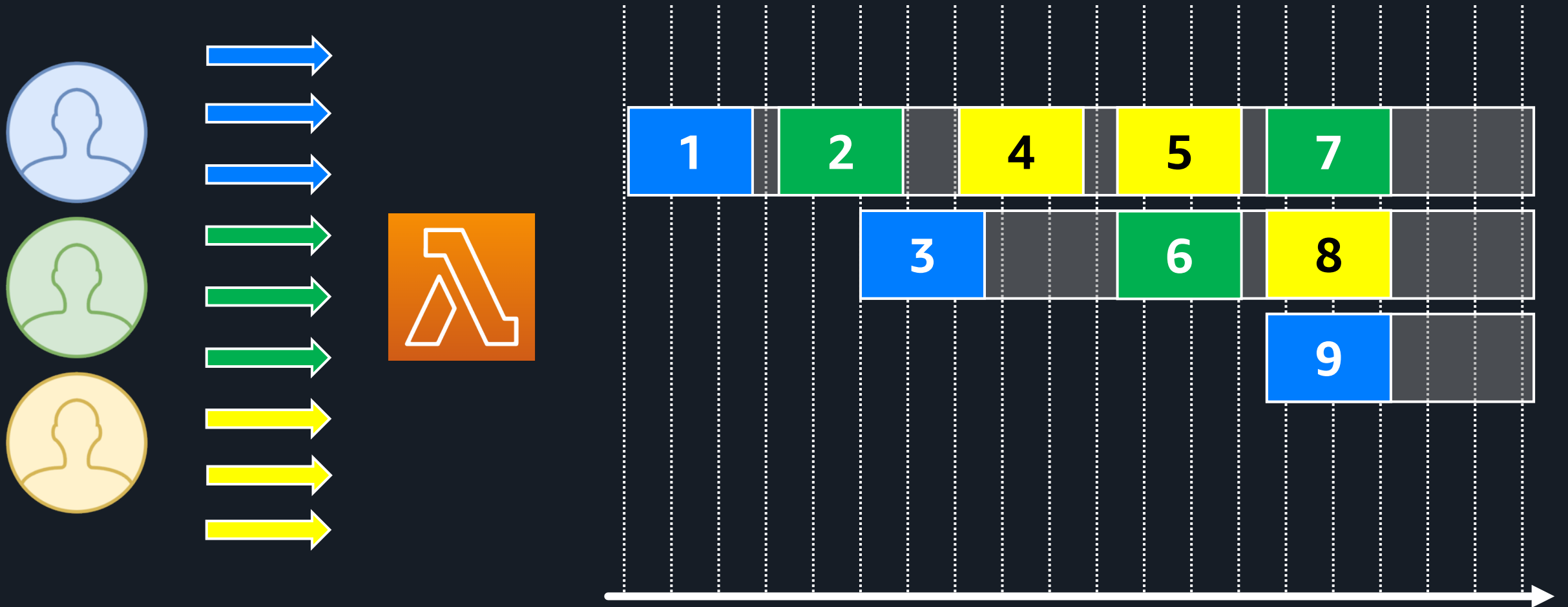
# Understanding Lambda concurrency

**Lots of invokes** are processed by **lots of execution environments**



# Understanding Lambda concurrency

**Lots of invokes** coming from **lots of tenants** are processed by **lots of execution environments**



# Demo time!

```
let memoryCounter = 0;

function getMemoryCounter(){
  return memoryCounter;
}

function setMemoryCounter(value){
  memoryCounter = value;
}
```

# Demo time!

```
function getDiskCounter(){
  checkFileExists();
  const diskCounterFileText = fs.readFileSync(
    DISK_COUNTER_FILE_NAME, 'utf8');
  let diskCounter = parseInt(diskCounterFileText);
  return diskCounter;
}

function setDiskCounter(value){
  fs.writeFileSync(DISK_COUNTER_FILE_NAME, value.toString());
}
```

# Demo time!

```
let memoryCounter = 0;

function getMemoryCounter(){
  return memoryCounter;
}

function setMemoryCounter(value){
  memoryCounter = value;
}
```

```
function getDiskCounter(){
  checkFileExists();
  const diskCounterFileText = fs.readFileSync(
    DISK_COUNTER_FILE_NAME, 'utf8');
  let diskCounter = parseInt(diskCounterFileText);
  return diskCounter;
}

function setDiskCounter(value){
  fs.writeFileSync(DISK_COUNTER_FILE_NAME, value.toString());
}
```



```
export const handler = async (event, ctx) => {
  const tenantId = ctx.tenantId;
  console.log(`tenantId=${tenantId}`);

  let memoryCounter = getMemoryCounter();
  console.log(`memoryCounter=${memoryCounter}`);
  memoryCounter++;
  setMemoryCounter(memoryCounter);

  let diskCounter = getDiskCounter();
  console.log(`diskCounter=${diskCounter}`);
  diskCounter++;
  setDiskCounter(diskCounter);

  return {
    statusCode: 200,
    body: JSON.stringify({
      memoryCounter, diskCounter, tenantId
    })
  };
};
```

# Demo time!

```
let memoryCounter = 0;

function getMemoryCounter(){
  return memoryCounter;
}

function setMemoryCounter(value){
  memoryCounter = value;
}
```

```
function getDiskCounter(){
  checkFileExists();
  const diskCounterFileText = fs.readFileSync(
    DISK_COUNTER_FILE_NAME, 'utf8');
  let diskCounter = parseInt(diskCounterFileText);
  return diskCounter;
}

function setDiskCounter(value){
  fs.writeFileSync(DISK_COUNTER_FILE_NAME, value.toString());
}
```

```
export const handler = async (event, ctx) => {
  const tenantId = ctx.tenantId;
  console.log(`tenantId=${tenantId}`);

  let memoryCounter = getMemoryCounter();
  console.log(`memoryCounter=${memoryCounter}`);
  memoryCounter++;
  setMemoryCounter(memoryCounter);

  let diskCounter = getDiskCounter();
  console.log(`diskCounter=${diskCounter}`);
  diskCounter++;
  setDiskCounter(diskCounter);

  return {
    statusCode: 200,
    body: JSON.stringify({
      memoryCounter, diskCounter, tenantId
    })
  };
};
```

# Demo time!

```
let memoryCounter = 0;

function getMemoryCounter(){
  return memoryCounter;
}

function setMemoryCounter(value){
  memoryCounter = value;
}
```

```
function getDiskCounter(){
  checkFileExists();
  const diskCounterFileText = fs.readFileSync(
    DISK_COUNTER_FILE_NAME, 'utf8');
  let diskCounter = parseInt(diskCounterFileText);
  return diskCounter;
}

function setDiskCounter(value){
  fs.writeFileSync(DISK_COUNTER_FILE_NAME, value.toString());
}
```

```
export const handler = async (event, ctx) => {
  const tenantId = ctx.tenantId;
  console.log(`tenantId=${tenantId}`);

  let memoryCounter = getMemoryCounter();
  console.log(`memoryCounter=${memoryCounter}`);
  memoryCounter++;
  setMemoryCounter(memoryCounter);

  let diskCounter = getDiskCounter();
  console.log(`diskCounter=${diskCounter}`);
  diskCounter++;
  setDiskCounter(diskCounter);

  return {
    statusCode: 200,
    body: JSON.stringify({
      memoryCounter, diskCounter, tenantId
    }),
  };
};
```



# Let's define the concern

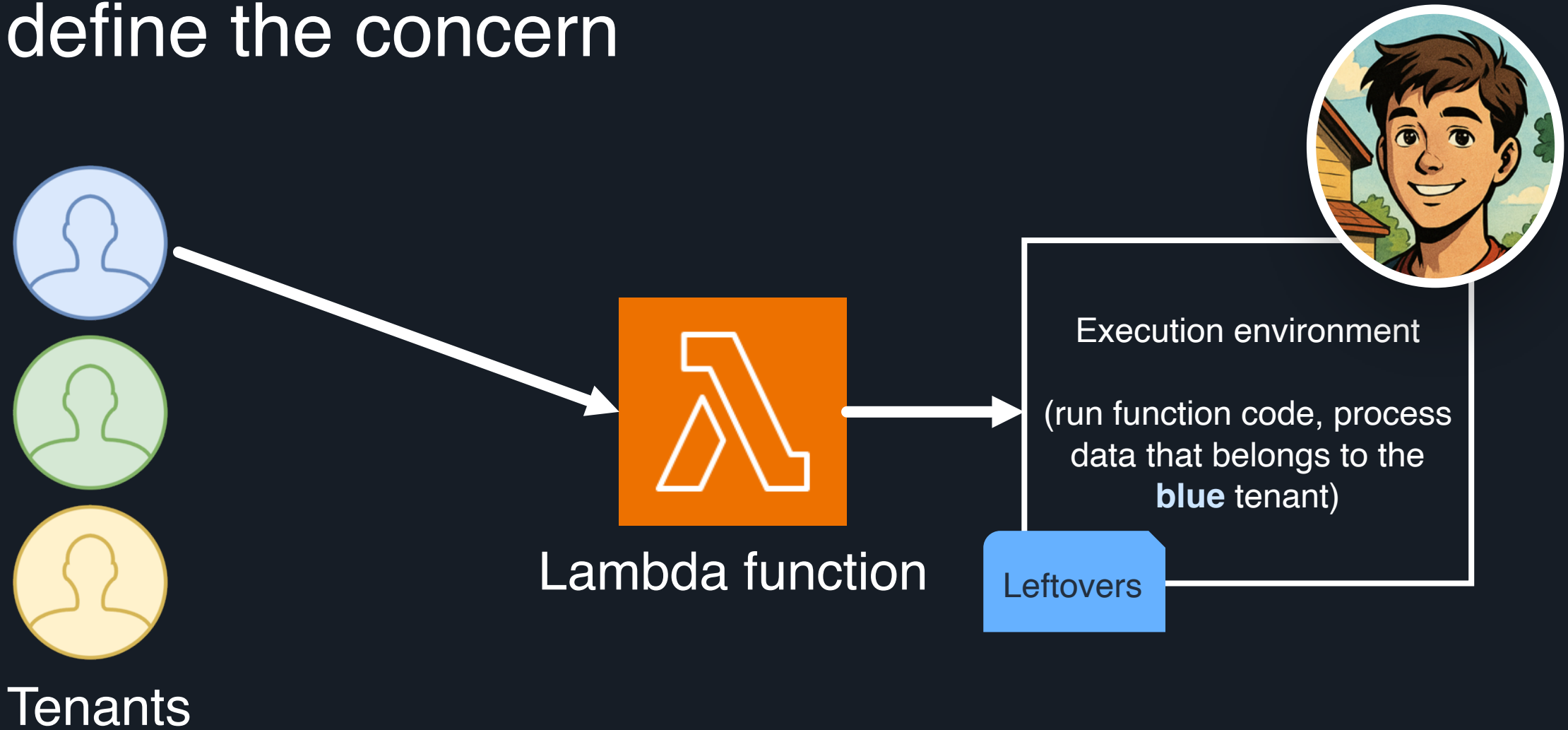


Tenants

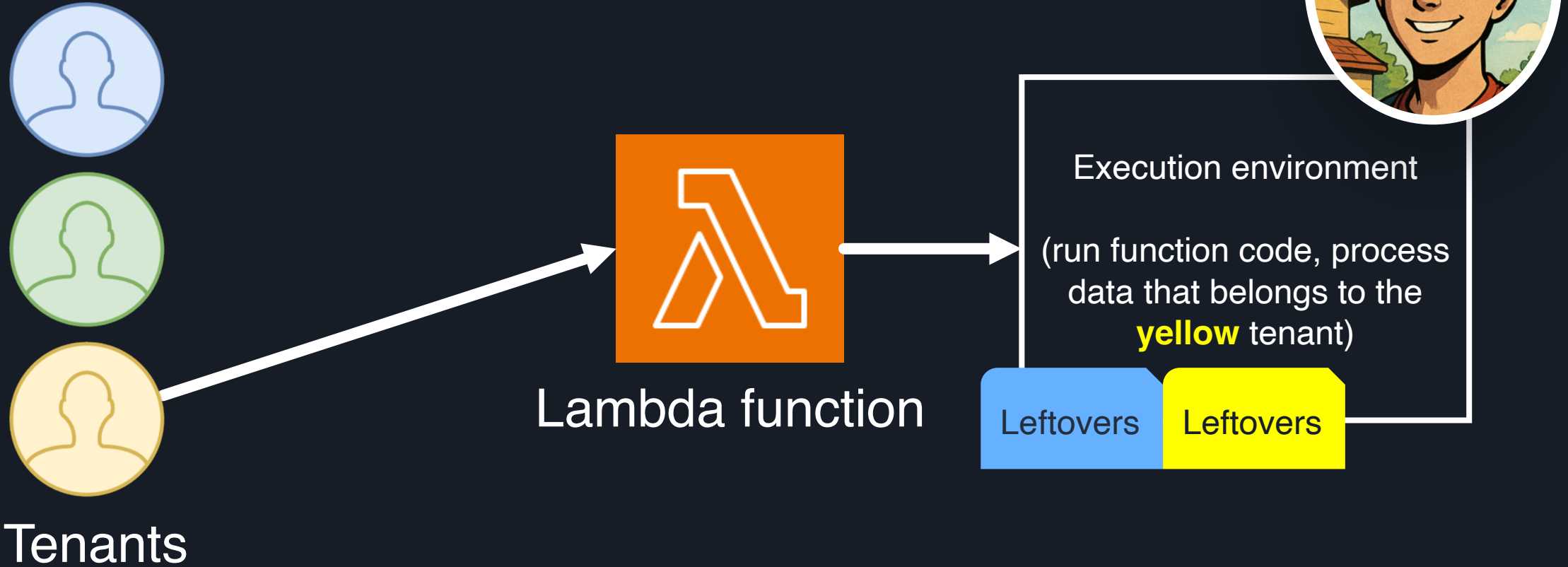


Lambda function

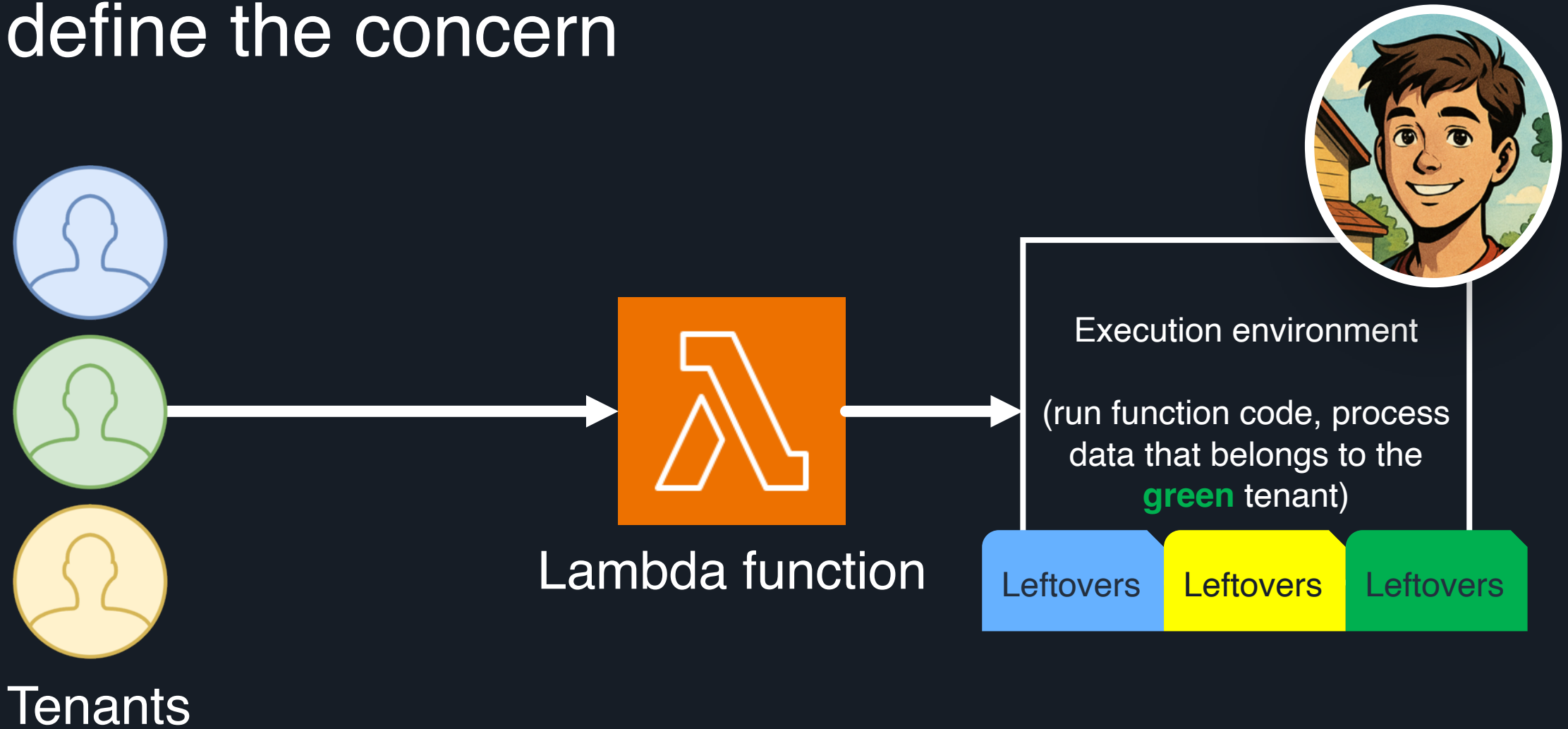
# Let's define the concern



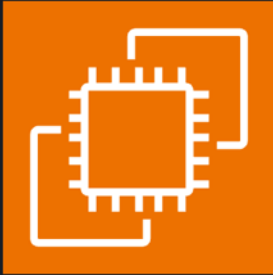
# Let's define the concern



# Let's define the concern



# A concern not unique to Lambda



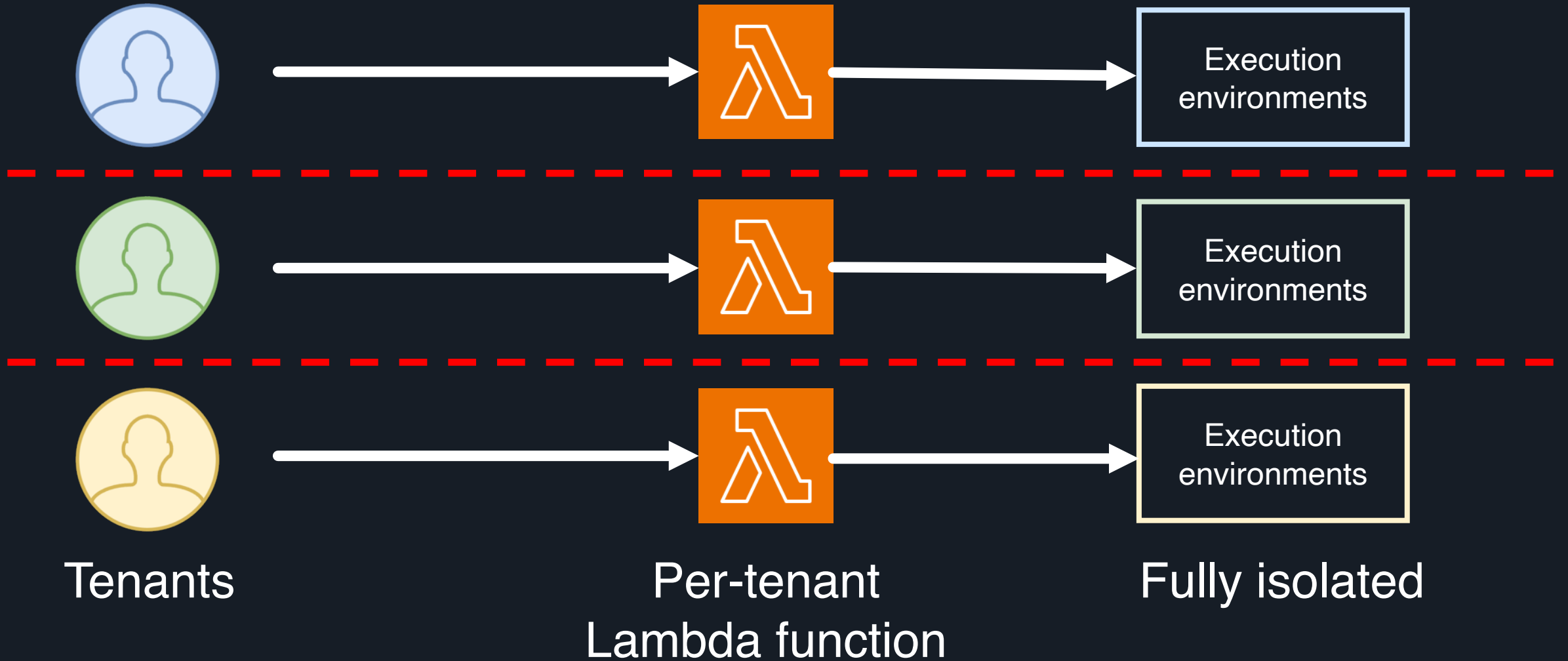
**(Or ANY other shared compute)**

# Existing solutions

(yes, it means there's a new one as well)



# Function-per-tenant (single tenant) model



# Function-per-tenant model

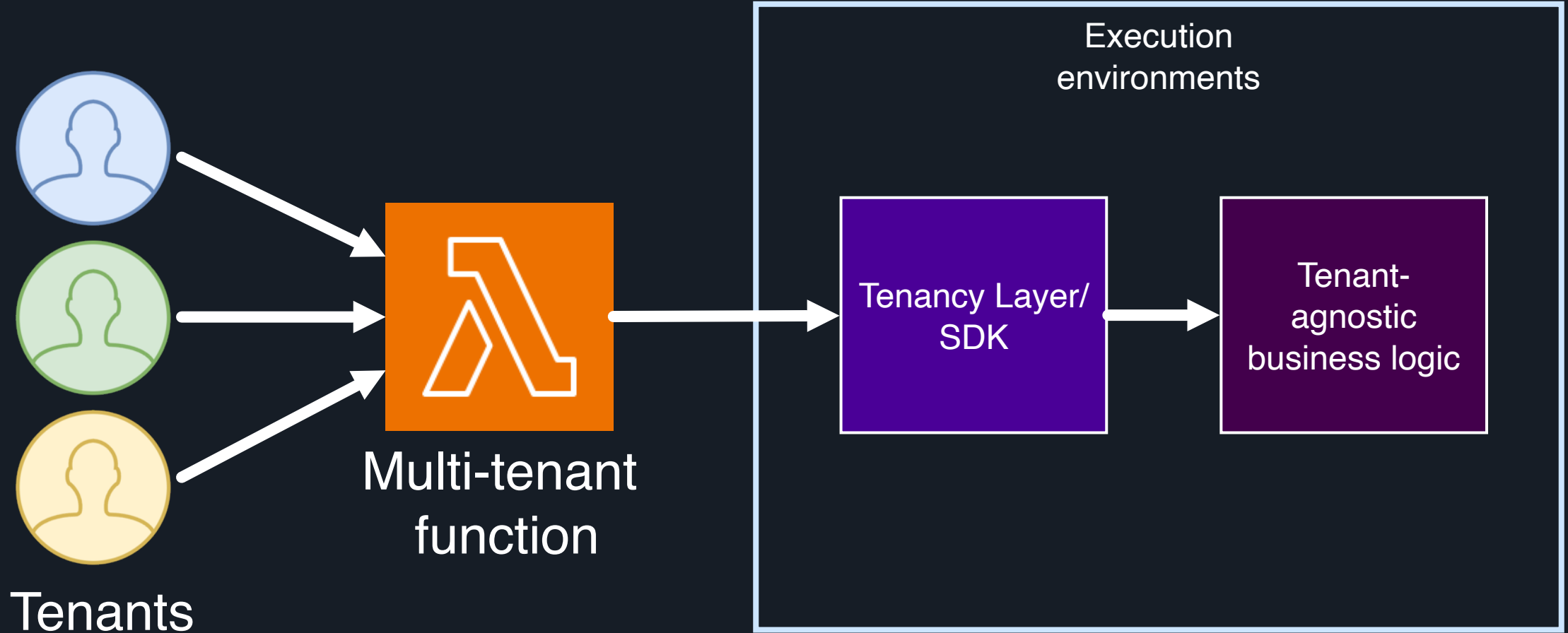
## Benefits

- Strong isolation
- Easier cost attribution
- Per-tenant resource configuration
- Per-tenant observability

## Considerations

- Operational sprawl
- Complex CI/CD
- Management API RPS limits
- Slower tenant onboarding
- Duplication and version drift
- Requires custom routing layer

# Multi-tenant function model





**CYBERARK**<sup>®</sup>  
THE IDENTITY SECURITY COMPANY<sup>®</sup>

Lambda powers our multi-tenant SaaS platform, **minimizing infrastructure management and enabling clean compute isolation** through our custom tenancy framework.



*- Ran Isenberg, AWS Serverless Hero,  
Principal Architect, CyberArk Platform Engineering  
Division*



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Multi-tenant functions model

## Benefits

- Higher resource reuse, potentially lower unit cost
- Faster tenant onboarding
- Simpler operations
- Rapid feature rollout
- Shared capacity
- Economies of scale

## Considerations

- Noisy neighbors potential
- Might require additional compute isolation
- Cleaning up the “leftovers”
- Per-tenant observability is harder
- Tenant-specific feature rollout is harder

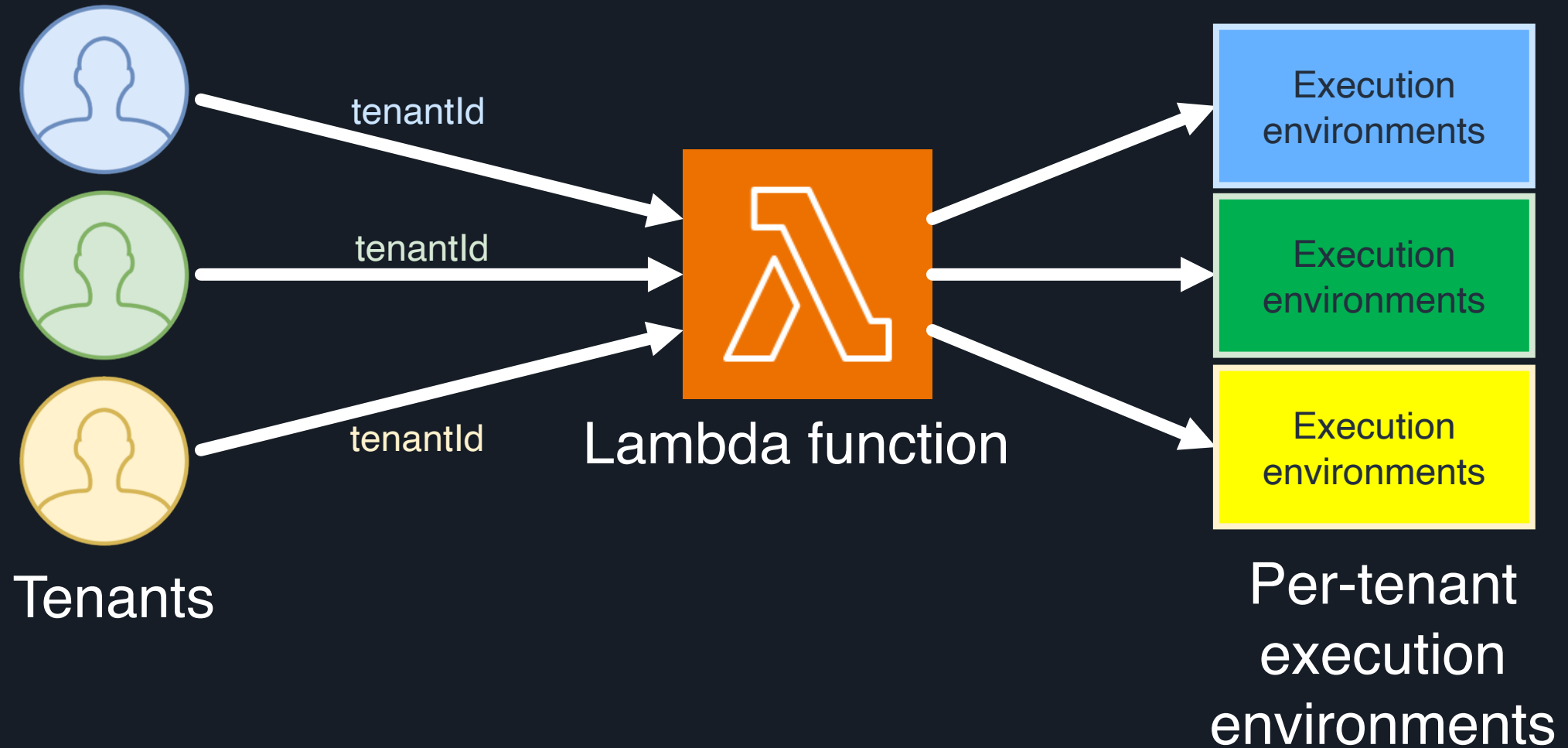


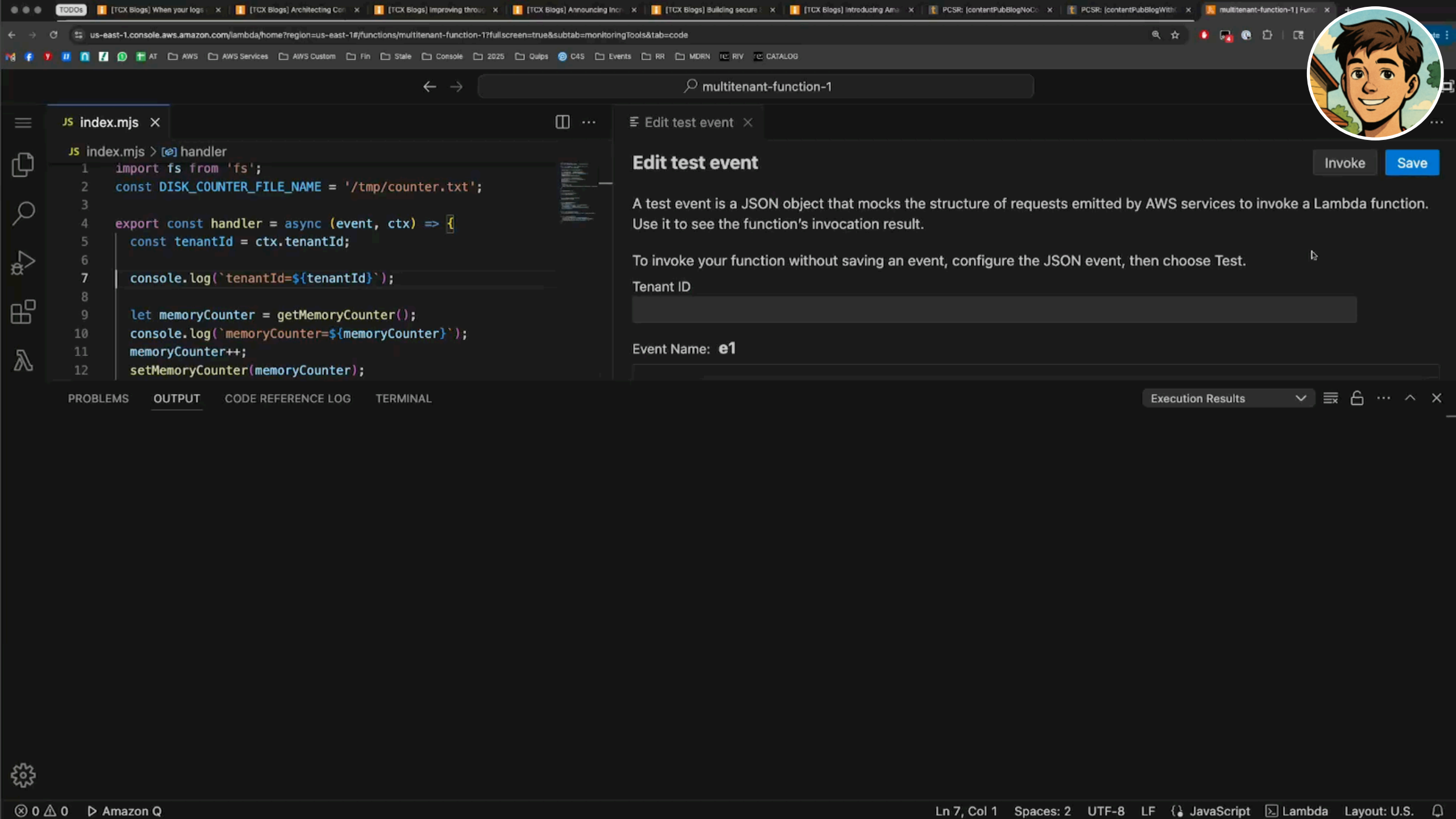
**IT WOULD BE GREAT  
TO HAVE A VENDOR-  
PROVIDED SOLUTION  
FOR TENANT COMPUTE  
ISOLATION...**

# Announcing the new Lambda Tenant Isolation Mode 🚀



# Announcing Lambda Tenant Isolation Mode





JS index.mjs

```

JS index.mjs > handler
1 import fs from 'fs';
2 const DISK_COUNTER_FILE_NAME = '/tmp/counter.txt';
3
4 export const handler = async (event, ctx) => {
5   const tenantId = ctx.tenantId;
6
7   console.log(`tenantId=${tenantId}`);
8
9   let memoryCounter = getMemoryCounter();
10  console.log(`memoryCounter=${memoryCounter}`);
11  memoryCounter++;
12  setMemoryCounter(memoryCounter);

```

Edit test event

### Edit test event

Invoke Save

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Tenant ID

Event Name: e1

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Execution Results



# Creating a function with tenant isolation

```
aws lambda create-function \  
  --function-name my-function1 \  
  --runtime nodejs22.x \  
  --zip-file fileb://my-function1.zip \  
  --handler index.handler \  
  --role arn:aws:iam:1234567890:role/my-function-role \  
  --tenancy-config '{"TenantIsolationMode": "PER_TENANT"}'
```

# Invoking a function with tenant isolation

```
aws lambda invoke \  
    --function-name my-function \  
    --tenant-id BlueTenant \  
    response.json
```

# Invoking a function with tenant isolation

```
export const handler = async (event, ctx) => {  
  const tenantId = ctx.tenantId;  
  console.log(`tenantId=${tenantId}`);  
}
```

ion: \$LATEST

07dd43 INFO

07dd43 INFO

tenantId=GreenTenant

memoryCounter=0

# How-to-use



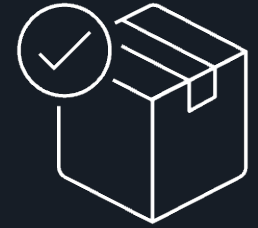
**No need to  
pre-register  
tenants**



**Unlimited  
number of  
tenants**



**Tenant ID is any  
arbitrary  
alphanumeric  
string (up to 128  
chars)**



**Supported for  
both ZIP and  
container  
functions**

# Considerations



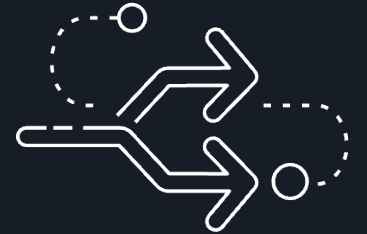
**Per-tenant cold starts**



**Concurrency quotas apply**

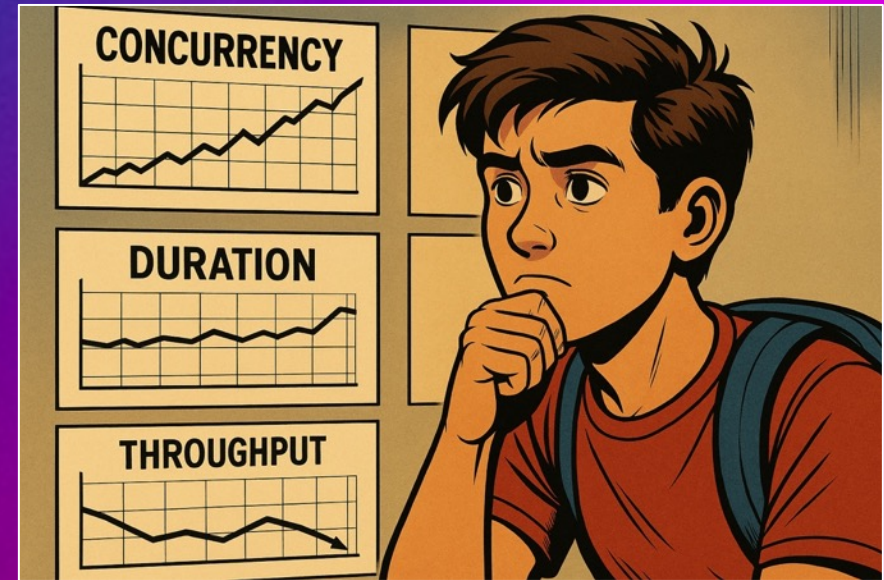


**Provision  
Concurrency not supported**



**Direct  
invocations or  
API Gateway  
only**

# Observability



# Observability – Use JSON logging

## Edit logging configuration

### Log content [Info](#)

#### Log format

For more control and better readability, we recommend choosing JSON for your log format. You can customize your function's logging level only for JSON.

Text

JSON

### Application log level [Info](#)

Application logs are the logs that your function code generates, also known as function logs. Choosing a more detailed logging level can incur additional costs from CloudWatch Logs.

TRACE (most detail)

DEBUG

INFO

WARN

ERROR

FATAL (least detail)

### System log level [Info](#)

System logs are the logs that Lambda generates, also known as system logs.

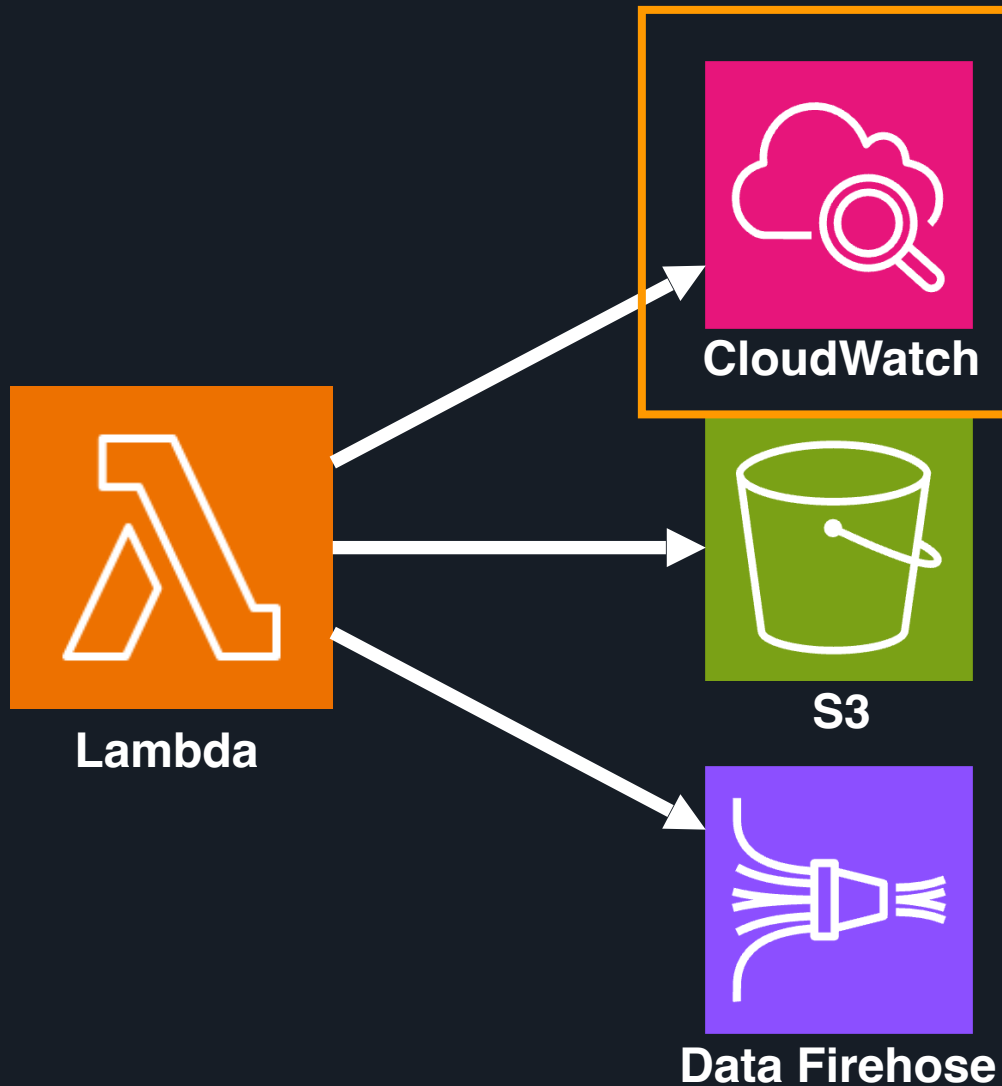
DEBUG (most detail)

INFO

WARN (least detail)

```
{
  "timestamp": "2025-10-14T00:51:13.651Z",
  "level": "INFO",
  "requestId": "ef733985-013a-4143-a53b-f1a12f82223d",
  "tenantId": BlueTenant,
  "message": "memoryCounter=0"
}
```

# Log taxonomy



Each function gets its own log group

## Log groups (7)

By default, we only load up to 10000 log groups.

🔍 Filter log groups or try pattern search

- Log group
- [/aws/lambda/f1](#)
- [/aws/lambda/lambda-movie-theater](#)
- [/aws/lambda/lambda-priming-sample](#)
- [/aws/lambda/lambda\\_deprecated\\_runtimes\\_api](#)
- [/aws/lambda/multi-tenant-function-1](#)
- [/aws/lambda/multitenant-function-1](#)
- [/lambdalogs/s3](#)

# Log taxonomy

Each function gets its own log group

Each execution environment gets its own log stream

**Log groups (7)**  
By default, we only load up to 10000 log groups.

Filter log groups or try pattern search

- Log group
- [/aws/lambda/f1](#)
- [/aws/lambda/lambda-movie-theater](#)
- [/aws/lambda/lambda-priming-sample](#)
- [/aws/lambda/lambda\\_deprecated\\_runtimes\\_api](#)
- [/aws/lambda/multi-tenant-function-1](#)
- [/aws/lambda/multitenant-function-1](#)
- [/lambda/logs/s3](#)

**Log streams (9)**

Filter log streams or try prefix search

- Log stream
- [2025/10/13/multitenant-function-1\[\\$LATEST\]6030176feb96486aac43471e2123abb9](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]2cd5c029b88d439bbee4ad12b3a2d80d](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]11eb05bd607b4657b1223f413d6c5fbc](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]4ff85b604b5845949756004c5b5a7ae6](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]7bedc17a4d3a4d84aced9d1e124b6287](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]59934ccaa9d64a5f82639449840f10a1](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]638ef94d36de4265839a2459b8412f86](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]10a159979ca240bdbd4ead54b3c0707d](#)
- [2025/10/13/multitenant-function-1\[\\$LATEST\]67afea53c76242cd9162a5577ebf4549](#)

# Log taxonomy



## Log streams (9)

🔍 Filter log streams or try prefix search

| Log stream

[2025/10/13/multitenant-function-1\[\\$LATEST\]6030176feb96486aac43471e2123abb9](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]2cd5c029b88d439bbee4ad12b3a2d80d](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]11eb05bd607b4657b1223f413d6c5fbc](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]4ff85b604b5845949756004c5b5a7ae6](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]7bedc17a4d3a4d84aced9d1e124b6287](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]59934ccaa9d64a5f82639449840f10a1](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]638ef94d36de4265839a2459b8412f86](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]10a159979ca240bdbd4ead54b3c0707d](#)

[2025/10/13/multitenant-function-1\[\\$LATEST\]67afea53c76242cd9162a5577ebf4549](#)



# Log taxonomy

2025/10/13/multitenant-function-1[\$LATEST]6030176feb96486aac43471e2123abb9

**Date**

**Function  
name**

**Version**

**Execution  
environment ID**



# Observability – filtering live logs

**Live Tail** [Info](#)

Highlight term

Highlight up to 5 terms (Not case sensitive)

**Filter** ×

Select log groups

Select up to 10 log groups from Standard log class. You can specify log streams if only one log group is selected.

Search and select log groups ▾

/aws/lambda/multitenant-function-1 ×

Clear all

Select log streams - optional

Select log streams by name ▾

Type in prefix

Add filter patterns (Case sensitive) - optional [Info](#)

BlueTenant

Apply filters

Select log group[s]

Add tenant Id as filter pattern

Select log stream[s]

# Observability – filtering live logs

## Live Tail [Info](#)

[Filter](#) [Actions](#) [Clear](#) [Cancel](#) [Start](#)

### Highlight term

Highlight up to 5 terms (Not case sensitive)

0 events/sec, 100% displayed

00:01:03

[View in columns](#)

### Filter

#### Select log groups

Select up to 10 log groups from Standard log class. You can specify log streams if only one log group is selected.

Search and select log groups

/aws/lambda/multitenant-function-1

Clear all

#### Select log streams - optional

Select log streams by name

Type in prefix

#### Add filter patterns (Case sensitive) - optional [Info](#)

BlueTenant

Apply filters

### Message

```
{ "time": "2025-10-13T23:37:53.095Z", "type": "platform.start", "record": { "requestId": "56960286-fbeb-4cb9-a164-c17e7d4f30b3", "functionArn": "arn:aws:lambda:us-east-1:28102... } }
{"timestamp": "2025-10-13T23:37:53.096Z", "level": "INFO", "requestId": "56960286-fbeb-4cb9-a164-c17e7d4f30b3", "tenantId": "BlueTenant", "message": "tenantId=BlueTenant"}
{"timestamp": "2025-10-13T23:37:53.096Z", "level": "INFO", "requestId": "56960286-fbeb-4cb9-a164-c17e7d4f30b3", "tenantId": "BlueTenant", "message": "memoryCounter=3"}
{"timestamp": "2025-10-13T23:37:53.096Z", "level": "INFO", "requestId": "56960286-fbeb-4cb9-a164-c17e7d4f30b3", "tenantId": "BlueTenant", "message": "diskCounter=3"}
{"time": "2025-10-13T23:37:53.115Z", "type": "platform.report", "record": { "requestId": "56960286-fbeb-4cb9-a164-c17e7d4f30b3", "metrics": { "durationMs": 1.934, "billedDuratio... } }
{"time": "2025-10-13T23:38:29.019Z", "type": "platform.start", "record": { "requestId": "40c6f1d8-603c-49d6-a1c5-10755e6c7cc0", "functionArn": "arn:aws:lambda:us-east-1:28102... } }
{"timestamp": "2025-10-13T23:38:29.034Z", "level": "INFO", "requestId": "40c6f1d8-603c-49d6-a1c5-10755e6c7cc0", "tenantId": "BlueTenant", "message": "tenantId=BlueTenant"}
{"timestamp": "2025-10-13T23:38:29.035Z", "level": "INFO", "requestId": "40c6f1d8-603c-49d6-a1c5-10755e6c7cc0", "tenantId": "BlueTenant", "message": "memoryCounter=4"}
{"timestamp": "2025-10-13T23:38:29.035Z", "level": "INFO", "requestId": "40c6f1d8-603c-49d6-a1c5-10755e6c7cc0", "tenantId": "BlueTenant", "message": "diskCounter=4"}
{"time": "2025-10-13T23:38:29.056Z", "type": "platform.report", "record": { "requestId": "40c6f1d8-603c-49d6-a1c5-10755e6c7cc0", "metrics": { "durationMs": 35.772, "billedDurati... } }
{"time": "2025-10-13T23:38:30.373Z", "type": "platform.start", "record": { "requestId": "ed981764-5209-4162-9e5d-55d870523101", "functionArn": "arn:aws:lambda:us-east-1:28102... } }
{"timestamp": "2025-10-13T23:38:30.374Z", "level": "INFO", "requestId": "ed981764-5209-4162-9e5d-55d870523101", "tenantId": "BlueTenant", "message": "tenantId=BlueTenant"}
{"timestamp": "2025-10-13T23:38:30.374Z", "level": "INFO", "requestId": "ed981764-5209-4162-9e5d-55d870523101", "tenantId": "BlueTenant", "message": "memoryCounter=5"}
{"timestamp": "2025-10-13T23:38:30.374Z", "level": "INFO", "requestId": "ed981764-5209-4162-9e5d-55d870523101", "tenantId": "BlueTenant", "message": "diskCounter=5"}
{"time": "2025-10-13T23:38:30.376Z", "type": "platform.report", "record": { "requestId": "ed981764-5209-4162-9e5d-55d870523101", "metrics": { "durationMs": 1.899, "billedDuratio... } } }
```

[Scroll to latest events](#)



# Observability – Logs Insights

**Select log groups by**


Log group name ▼

**Selection criteria**

Select up to 50 log groups

/aws/lambda/multitenant-function-1 ✕

```
1 fields @logStream, @message
2 | filter tenantId=='BlueTenant' or record.tenantId=='BlueTenant'
3 | stats count() as logCount by @logStream
4 | sort @timestamp desc
5
```

 **Query generator**

**Run query** **Cancel** **Save** **History**

# Observability – Logs Insights

Select log groups by


Log group name ▼

Selection criteria

Select up to 50 log groups

/aws/lambda/multitenant-function-1 X


#	@logStream	LogCount
▶ 1	<u>2025/10/13/multitenant-function-1[\$LATEST]7e79a4d1ffbf435099f0ed7f7605511c</u>	31
▶ 2	<u>2025/10/13/multitenant-function-1[\$LATEST]e8ce8a31f2e9440788ece2bf8ee7cc3f</u>	56
▶ 3	<u>2025/10/13/multitenant-function-1[\$LATEST]cf2dd11a22304118ba52be929ff1363c</u>	16

 Query generator

Run query Cancel Save History

# Observability – Logs Insights

```
1 fields @message
2 | filter tenantId=='BlueTenant' or record.tenantId=='BlueTenant'
3 | limit 1000
4
```

 **Query generator**

**Run query** **Cancel** **Save** **History**

# Observability – Logs Insights

```
# @message
▶ 1 {"time":"2025-10-13T23:38:30.376Z","type":"platform.report","record":{"requestId":"ed981764-5209-4162-9e5d-55d870523101","metrics":{"durationMs":1.899,"billedDuratio
▶ 2 {"timestamp":"2025-10-13T23:38:30.374Z","level":"INFO","requestId":"ed981764-5209-4162-9e5d-55d870523101","tenantId":"BlueTenant","message":"tenantId=BlueTenant"}
▶ 3 {"timestamp":"2025-10-13T23:38:30.374Z","level":"INFO","requestId":"ed981764-5209-4162-9e5d-55d870523101","tenantId":"BlueTenant","message":"memoryCounter=5"}
▶ 4 {"timestamp":"2025-10-13T23:38:30.374Z","level":"INFO","requestId":"ed981764-5209-4162-9e5d-55d870523101","tenantId":"BlueTenant","message":"diskCounter=5"}
▶ 5 {"time":"2025-10-13T23:38:30.373Z","type":"platform.start","record":{"requestId":"ed981764-5209-4162-9e5d-55d870523101","functionArn":"arn:aws:lambda:us-east-1:28102
▶ 6 {"time":"2025-10-13T23:38:29.056Z","type":"platform.report","record":{"requestId":"40c6f1d8-603c-49d6-a1c5-10755e6c7cc0","metrics":{"durationMs":35.772,"billedDuratio
▶ 7 {"timestamp":"2025-10-13T23:38:29.035Z","level":"INFO","requestId":"40c6f1d8-603c-49d6-a1c5-10755e6c7cc0","tenantId":"BlueTenant","message":"memoryCounter=4"}
▶ 8 {"timestamp":"2025-10-13T23:38:29.035Z","level":"INFO","requestId":"40c6f1d8-603c-49d6-a1c5-10755e6c7cc0","tenantId":"BlueTenant","message":"diskCounter=4"}
▶ 9 {"timestamp":"2025-10-13T23:38:29.034Z","level":"INFO","requestId":"40c6f1d8-603c-49d6-a1c5-10755e6c7cc0","tenantId":"BlueTenant","message":"tenantId=BlueTenant"}
▶ 10 {"time":"2025-10-13T23:38:29.019Z","type":"platform.start","record":{"requestId":"40c6f1d8-603c-49d6-a1c5-10755e6c7cc0","functionArn":"arn:aws:lambda:us-east-1:28102
▶ 11 {"time":"2025-10-13T23:37:53.115Z","type":"platform.report","record":{"requestId":"56960286-fbeb-4cb9-a164-c17e7d4f30b3","metrics":{"durationMs":1.934,"billedDuratio
▶ 12 {"timestamp":"2025-10-13T23:37:53.096Z","level":"INFO","requestId":"56960286-fbeb-4cb9-a164-c17e7d4f30b3","tenantId":"BlueTenant","message":"tenantId=BlueTenant"}
▶ 13 {"timestamp":"2025-10-13T23:37:53.096Z","level":"INFO","requestId":"56960286-fbeb-4cb9-a164-c17e7d4f30b3","tenantId":"BlueTenant","message":"memoryCounter=3"}
▶ 14 {"timestamp":"2025-10-13T23:37:53.096Z","level":"INFO","requestId":"56960286-fbeb-4cb9-a164-c17e7d4f30b3","tenantId":"BlueTenant","message":"diskCounter=3"}
▶ 15 {"time":"2025-10-13T23:37:53.095Z","type":"platform.start","record":{"requestId":"56960286-fbeb-4cb9-a164-c17e7d4f30b3","functionArn":"arn:aws:lambda:us-east-1:28102
▶ 16 {"time":"2025-10-13T23:37:03.674Z","type":"platform.report","record":{"requestId":"2080ab0b-83af-4017-9345-08509d3d0ae0","metrics":{"durationMs":2.216,"billedDuratio
▶ 17 {"timestamp":"2025-10-13T23:37:03.671Z","level":"INFO","requestId":"2080ab0b-83af-4017-9345-08509d3d0ae0","tenantId":"BlueTenant","message":"diskCounter=2"}
```

# Observability – Custom Metrics

```
import { Metrics, MetricUnit } from '@aws-lambda-powertools/metrics';

const metrics = new Metrics({
  namespace: 'BookingApp',
  serviceName: process.env.SERVICE_NAME,
  region: process.env.AWS_REGION
});

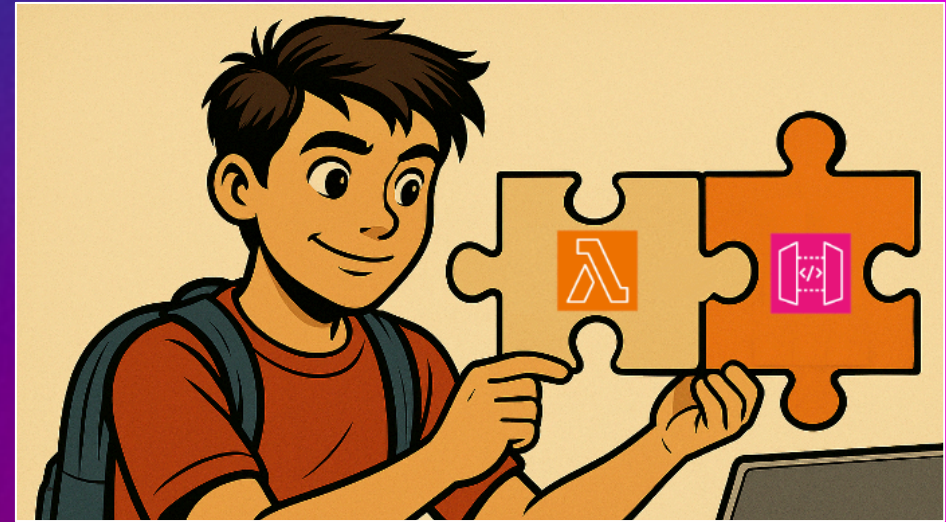
export const handler = async (event, context) => {

  metrics.addDimension('tenant', context.tenantId);

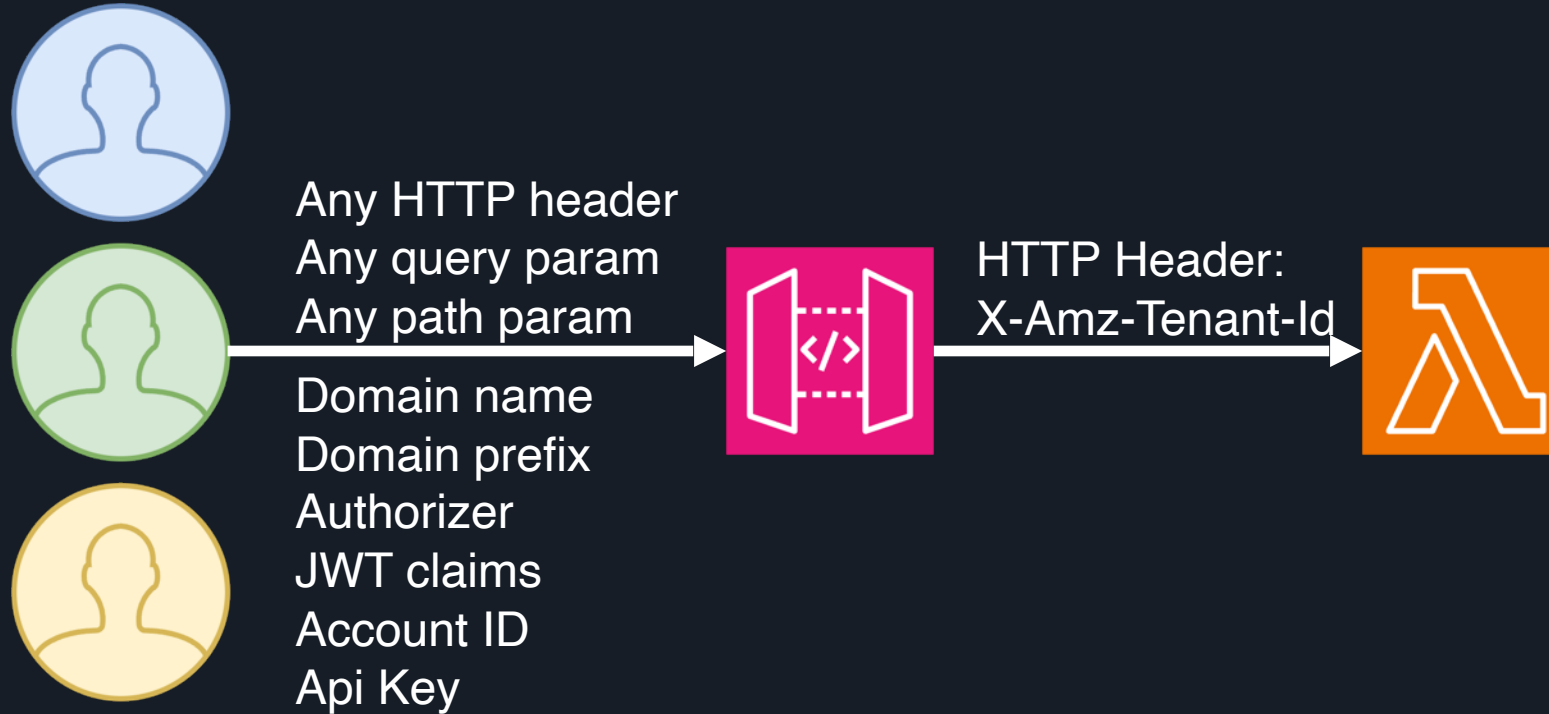
  // Do business logic

  metrics.addMetric('successfulBooking', MetricUnit.Count, 1);
  metrics.publishStoredMetrics();
};
```

# Integrating with API Gateway



# Integrating with API Gateway



# Integrating with API Gateway



# Integrating with API Gateway



```
const lambdaIntegration = new ApiGw.LambdaIntegration(fn, {  
  requestParameters: {  
    'integration.request.header.X-Amz-Tenant-Id': 'method.request.header.x-tenant-id'  
  }  
});
```

```
resource.addMethod('GET', lambdaIntegration, {  
  requestParameters: {  
    'method.request.header.x-tenant-id': true  
  }  
});
```

# Integrating with API Gateway



```
const lambdaIntegration = new ApiGw.LambdaIntegration(fn, {  
  requestParameters: {  
    'integration.request.header.X-Amz-Tenant-Id': 'method.request.header.x-tenant-id'  
  }  
});  
  
resource.addMethod('GET', lambdaIntegration, {  
  requestParameters: {  
    'method.request.header.x-tenant-id': true  
  }  
});
```

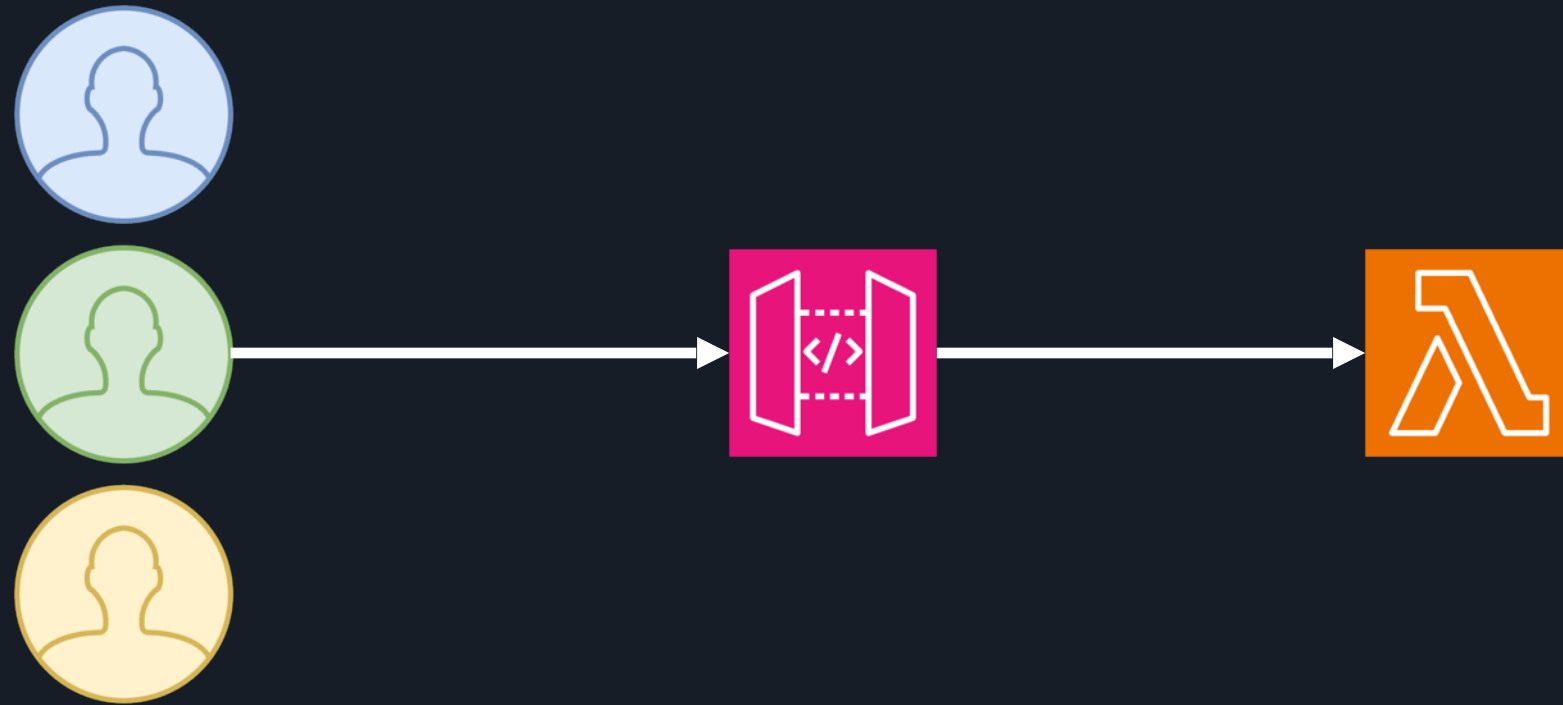
# Integrating with API Gateway



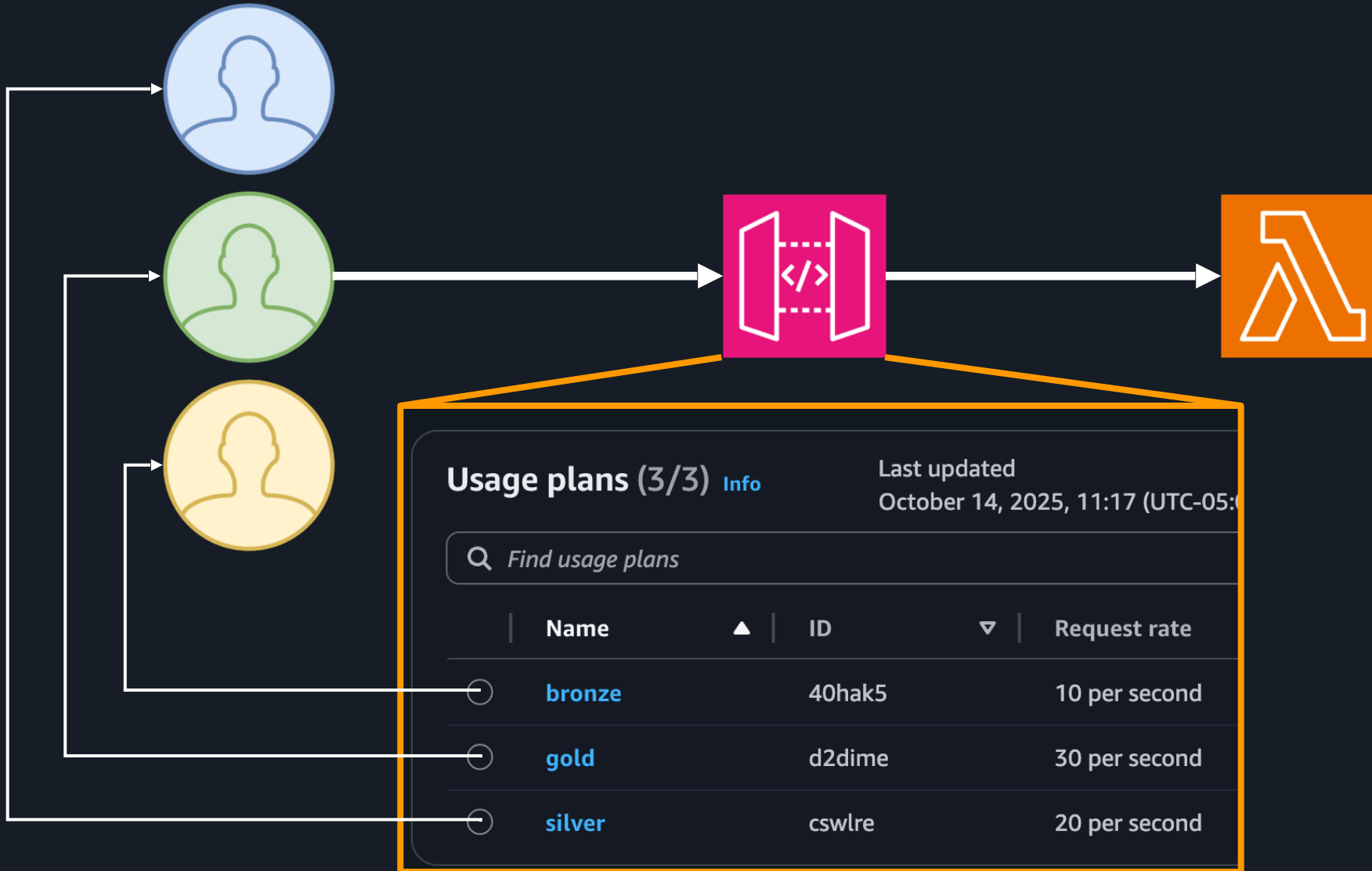
```
const lambdaIntegration = new ApiGw.LambdaIntegration(fn, {  
  requestParameters: {  
    'integration.request.header.X-Amz-Tenant-Id': value  
  }  
});
```

- method.**request.header**.{header-name}
- method.**request.querystring**.{parameter-name}
- method.**request.path**.{parameter-name}
- method.**request.body**.{json\_path\_expression}
- context.**authorizer.principalId**
- context.**authorizer**.{custom-property-name}
- context.**domainPrefix**
- And many many more...

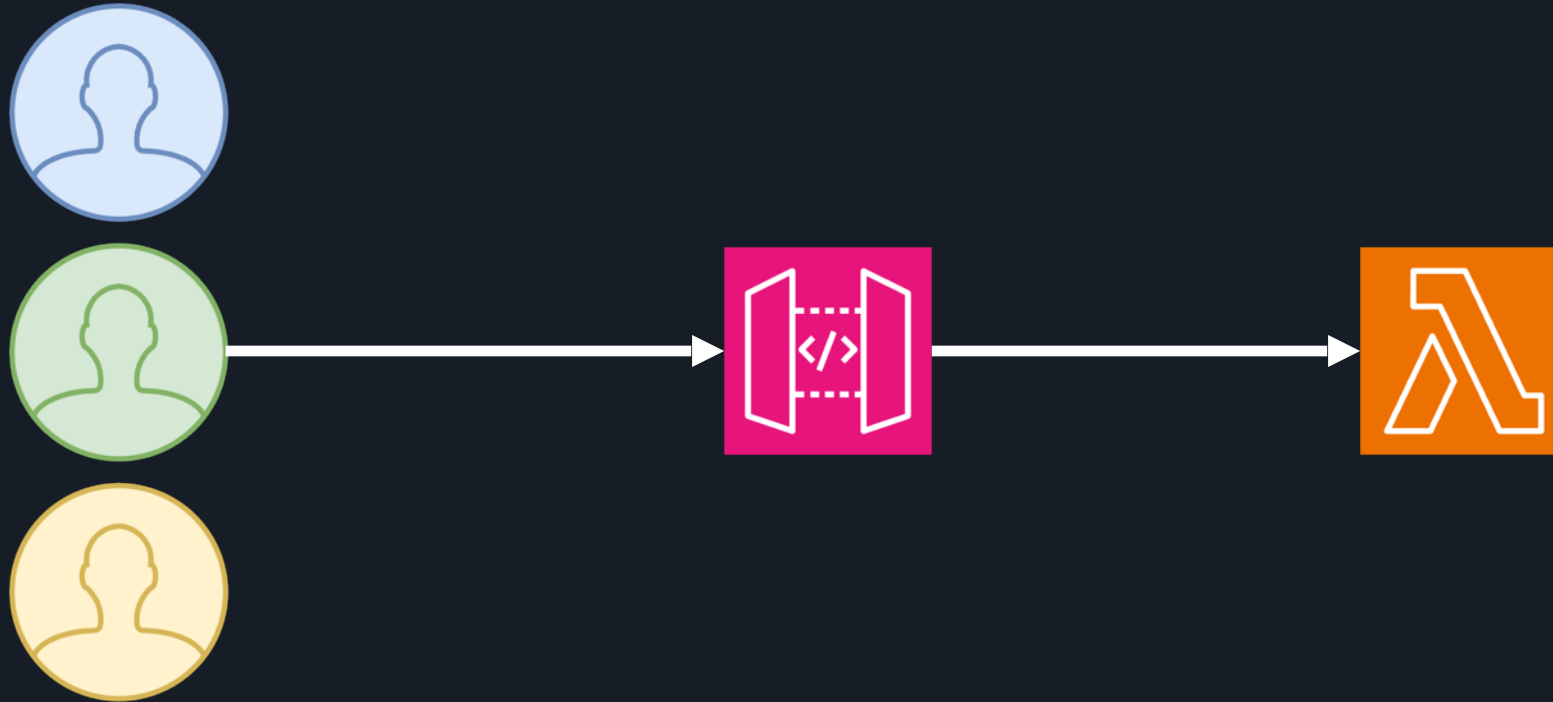
# Addressing the noisy neighbours problem



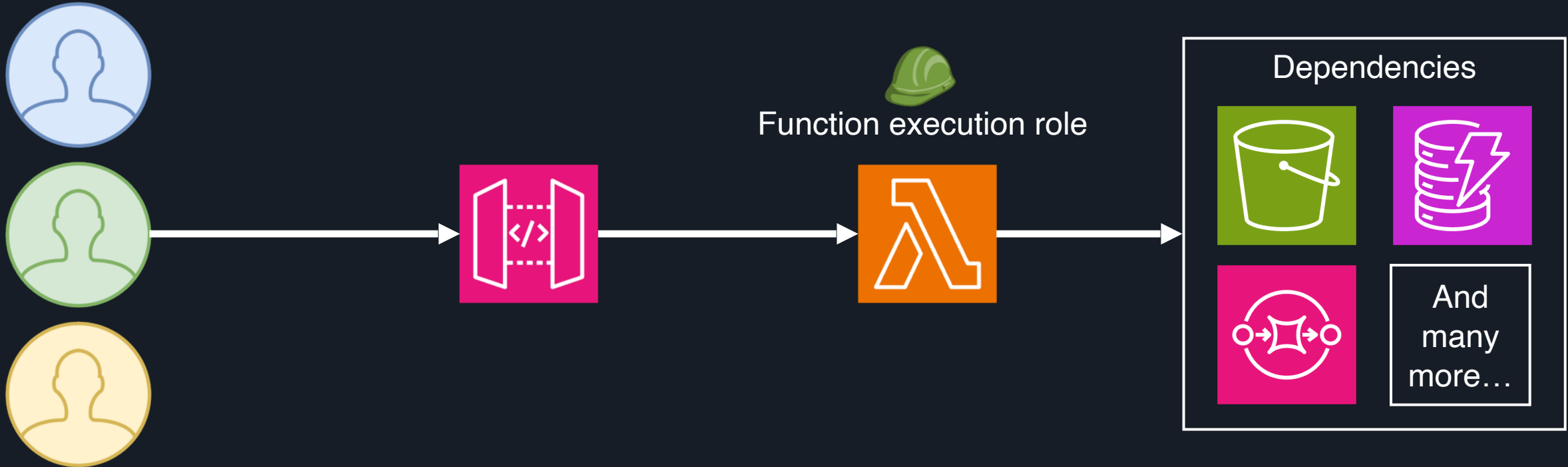
# Addressing the noisy neighbours problem



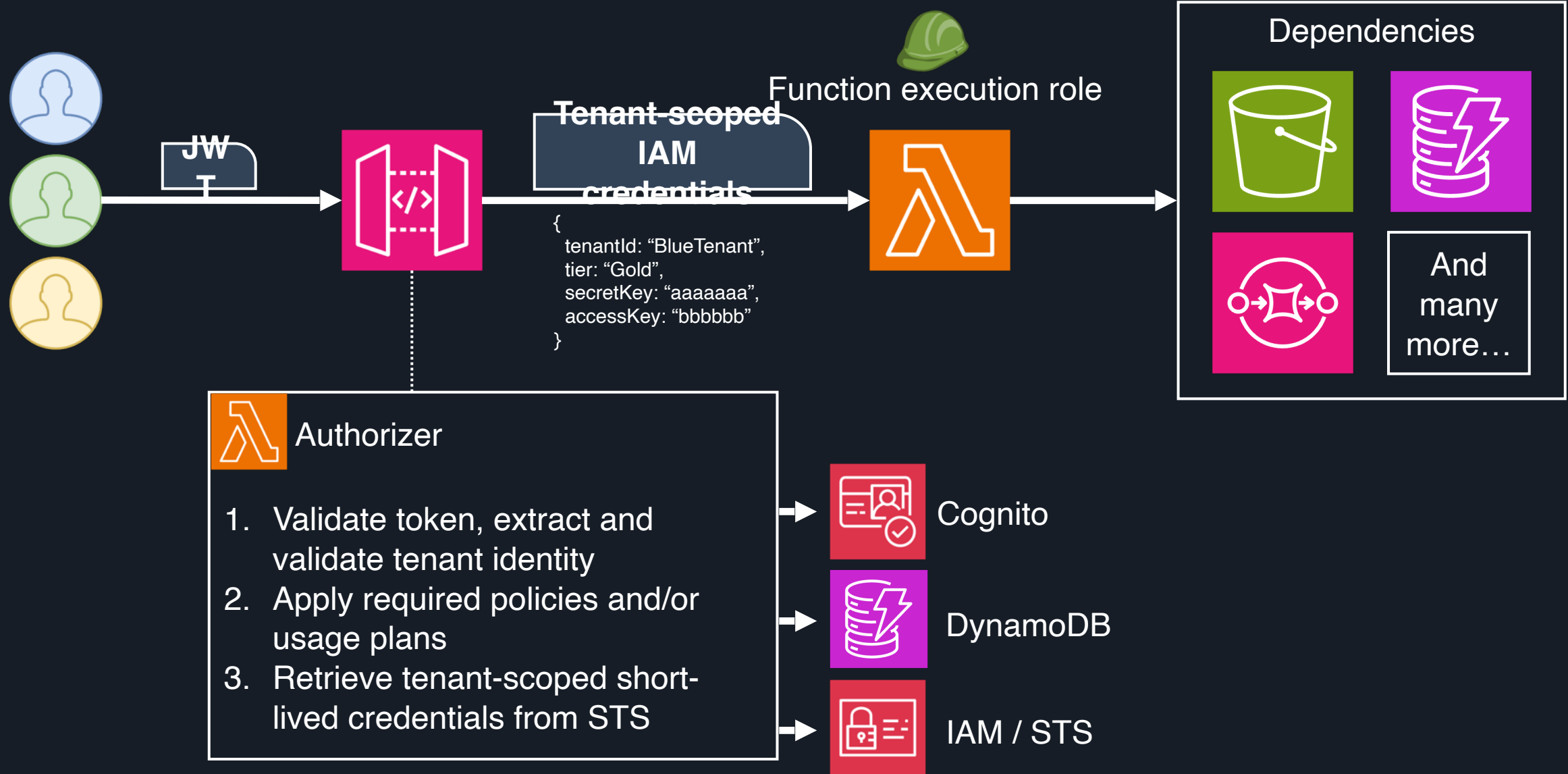
# Integrating with API Gateway



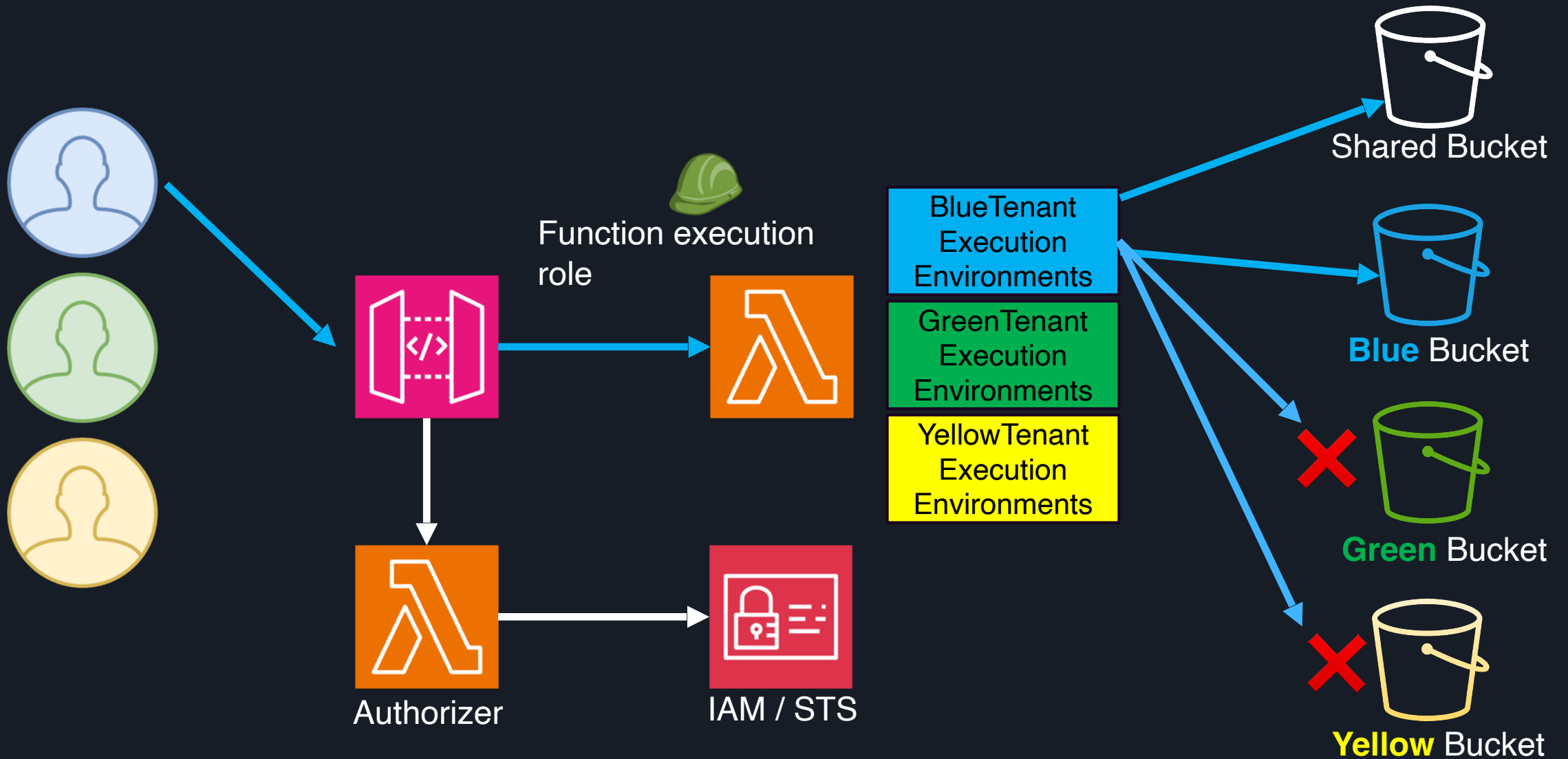
# Integrating with API Gateway



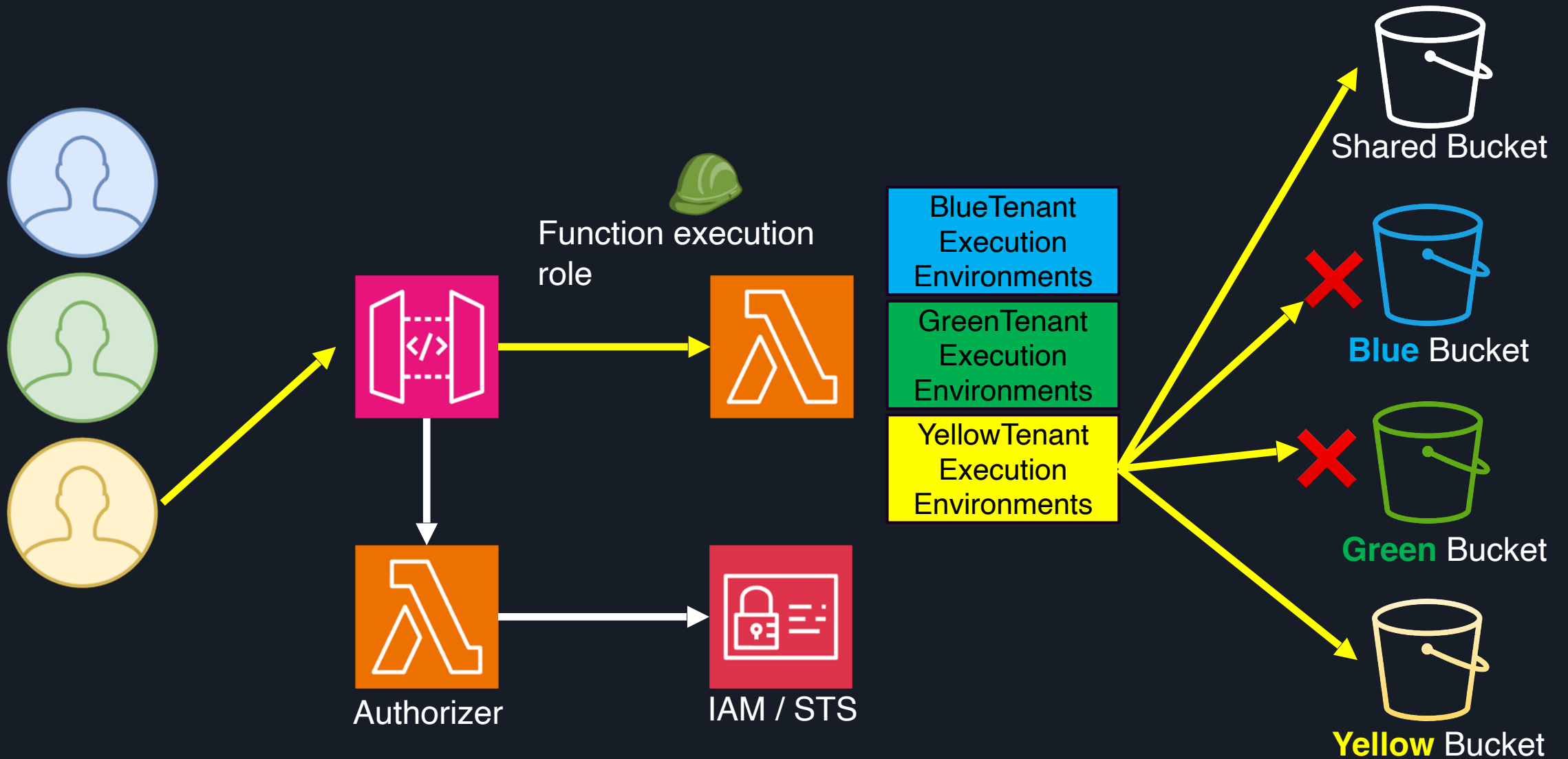
# Integrating with API Gateway



# Integrating with API Gateway

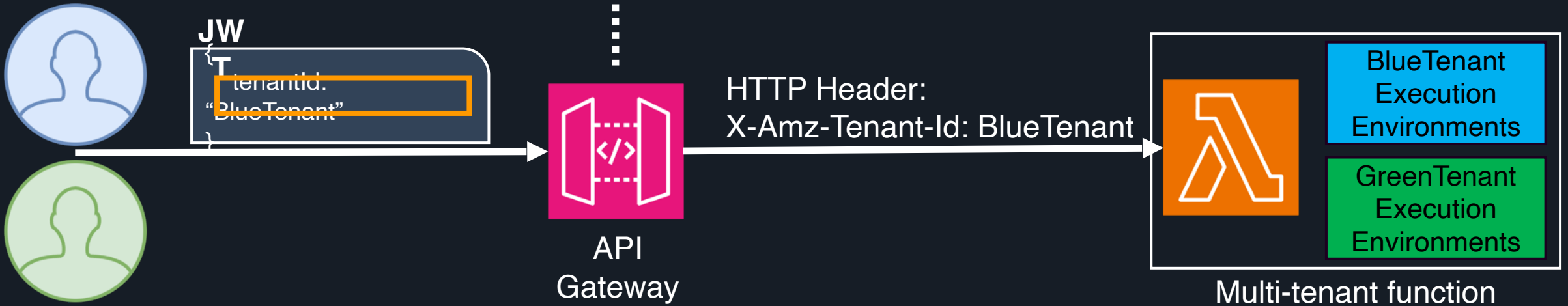


# Integrating with API Gateway



# Demo time!

```
const lambdaIntegration = new ApiGw.LambdaIntegration(fn, {  
  requestParameters: {  
    'integration.request.header.X-Amz-Tenant-Id': 'context.authorizer.tenantId'  
  }  
});
```



```
return {  
  principalId,  
  policyDocument: { ...  
},  
  context: {  
    tenantId  
  },  
  usageIdentifierKey: usageKey  
}
```



https://5n0una6qhb.execute-api.us-east-1.amazonaws.com/prod/



Save



Share



GET



https://5n0una6qhb.execute-api.us-east-1.amazonaws.com/prod/

Send



Params

Auth

Headers (7)

Body

Scripts

Settings

Cookies

Headers

7 hidden

	Key	Value	De...	...	Bulk Edit	Presets	...
	Key	Value	Description				

Body



401 Unauthorized

61 ms

497 B



{ } JSON



Preview



Debug with AI



```
1 {
2   "message": "Unauthorized"
3 }
```

# Multi-tenant functions model

## Benefits

- Higher resource reuse, potentially lower unit cost
- Faster tenant onboarding
- Simpler operations
- Rapid feature rollout
- Shared capacity
- Economies of scale

## Considerations

- Noisy neighbors potential
- Might require additional compute isolation
- Cleaning up the “leftovers”
- Per-tenant observability is harder
- Tenant-specific feature rollout is harder

# Multi-tenant functions **with tenant isolation mode**

## Benefits

- Higher resource reuse, potentially lower unit cost
- Faster tenant onboarding
- Simpler operations
- Rapid feature rollout
- Shared capacity
- Economies of scale

## Considerations

- ✓ Noisy neighbors potential  
(addressed with API Gateway usage plans)
- ✓ Might require additional compute isolation
- ✓ Cleaning up the “leftovers”
- ✓ Per-tenant observability is harder
- ✓ Tenant-specific feature rollout is harder



# Conclusion

# 1

Understand tenant isolation requirements of your workload and implementation specifics within your architecture.

# 2

Leverage Lambda's tenant isolation mode for multi-tenant SaaS applications that require high degree of vendor-provided compute isolation.

# 3

Use built in observability features for per-tenant monitoring and API Gateway integrations for stronger security in your SaaS apps.

# Check out these other sessions

## **CNS370: Secure user code execution with AWS Lambda for SaaS platforms**

Tuesday (Dec 2) @ 5:30pm – MGM 306 – Chalk Talk

## **CNS403: Best practices for serverless developers**

Tuesday (Dec 2) @ 12:00p – Mandalay Bay Islander F – Breakout Session

## **API309: Serverlesspresso: Build an event-driven application**

Thursday (Dec 4) @ 12:00p – MGM Grand 120 - Workshop



# Serverless & AppInt Resources

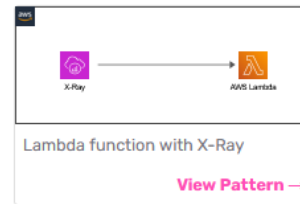
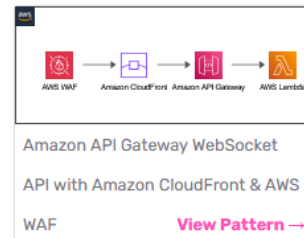
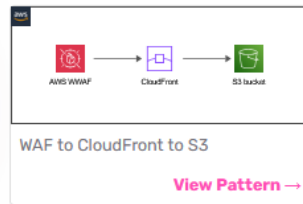


<https://s12d.com/RIV2025>

Continue your AWS serverless learning  
Powertools for AWS Lambda  
Serverless Land patterns

## Serverless **Patterns** Collection

Build integrations using infrastructure as code with **serverless patterns**





# Thank you!

**Anton Aleksandrov**

 antonal80

 [aal80.github.io/whoami](https://github.com/aal80/whoami)

**Bill Tarr**

 billtarr



This session's content  
and much more

Please complete the session  
survey in the mobile app