BUILDING A HIGH-PERFORMANCE ON-OFF LOAD TEST ENVIRONMENT ON AWS

Pascal Euhus

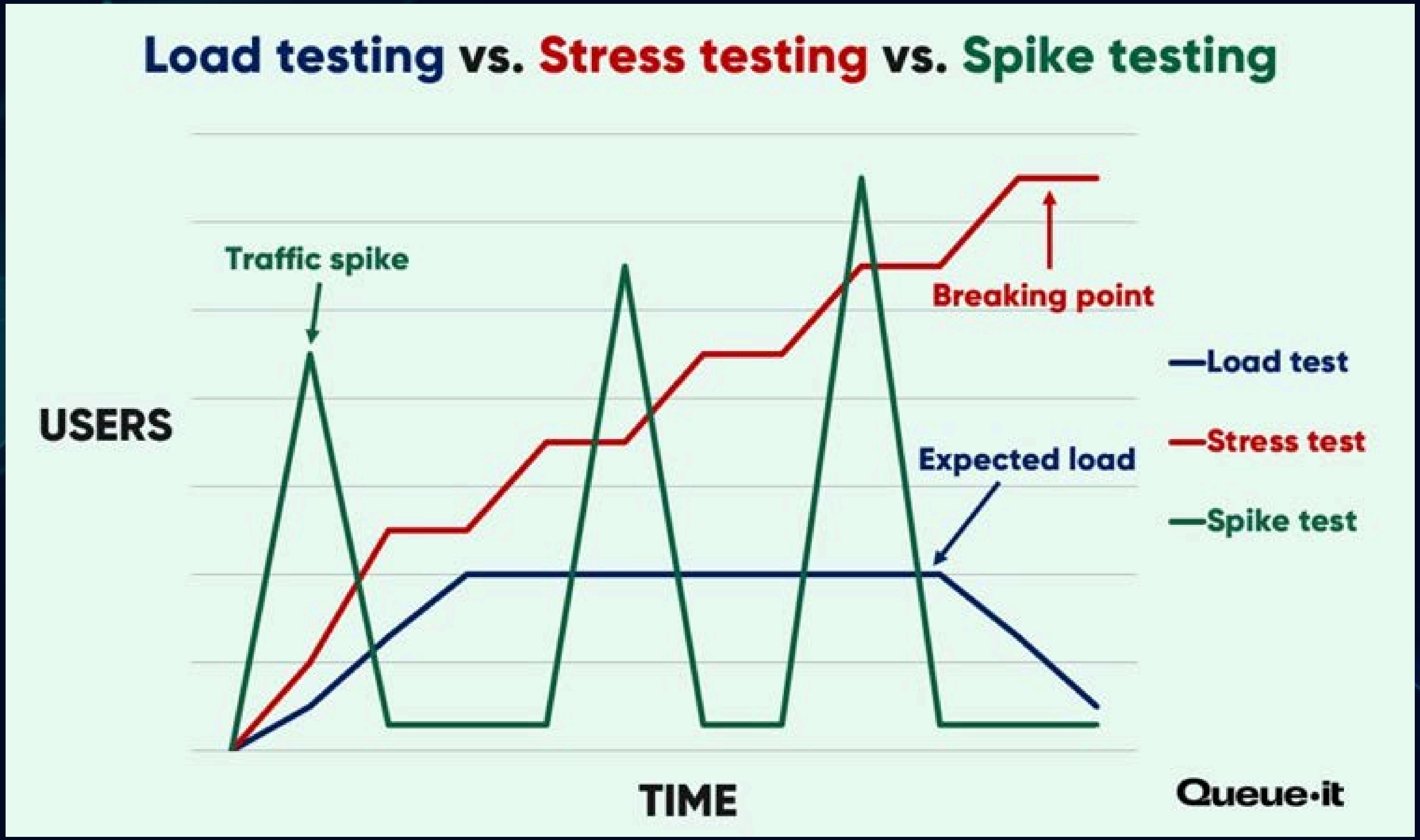
pascal.euhus.dev

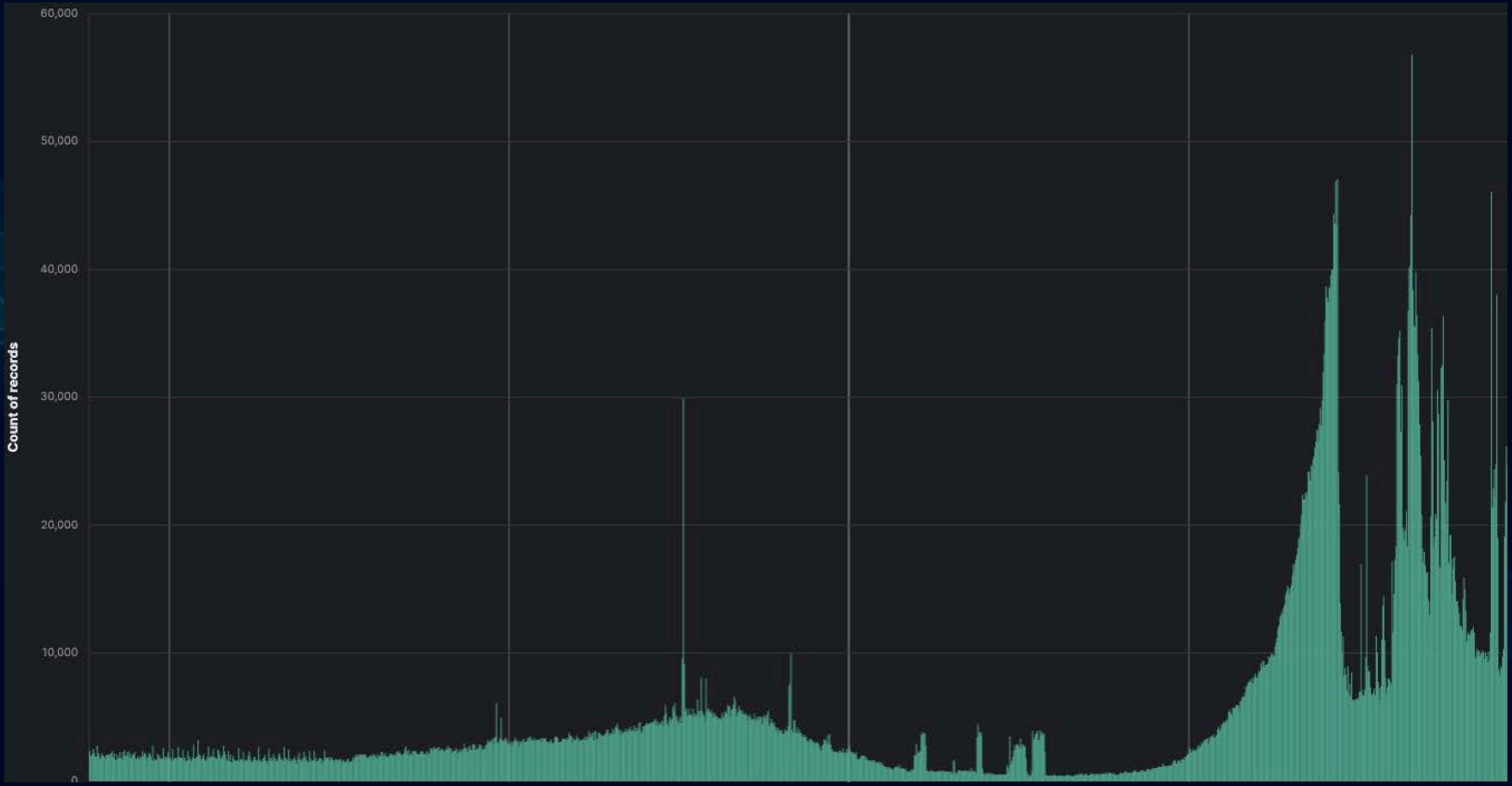
reservix
dein ticketportal

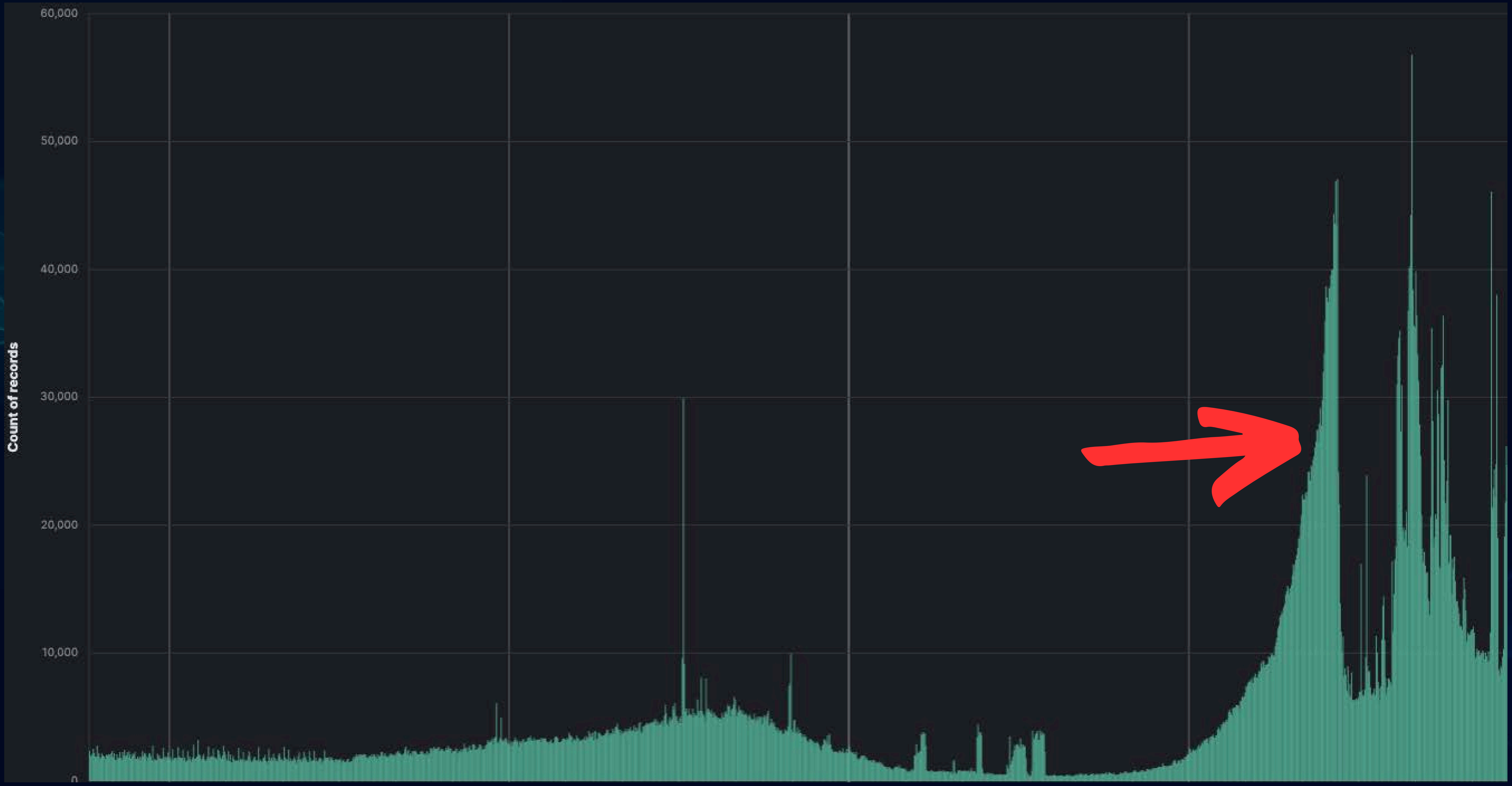
Pascal Euhus

- Lead Architect @ Reservix
- Freiburg
- AWS Community Builder - Serverless
- 7+ years building with AWS

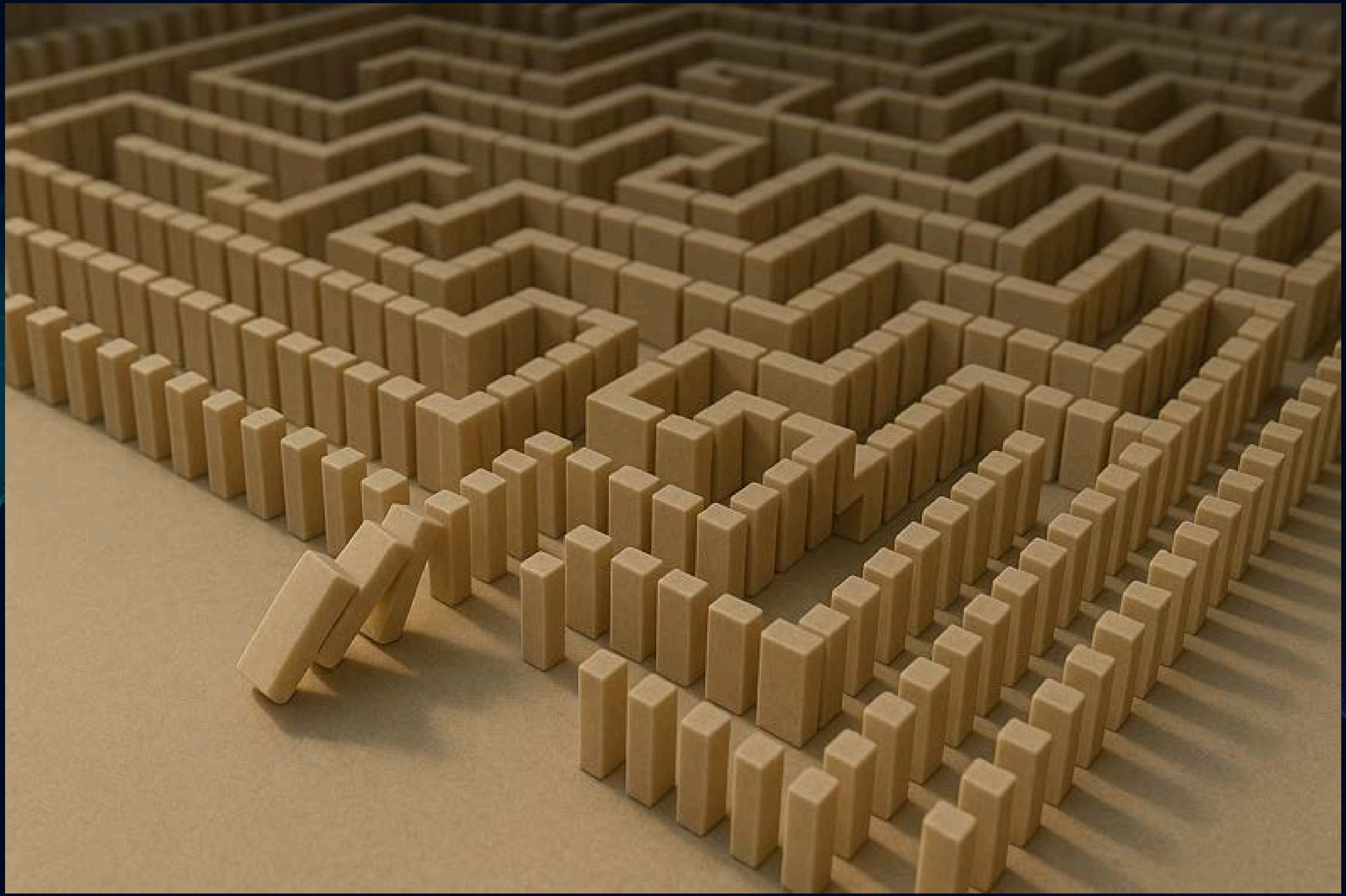
FC ST. PAULI
1910

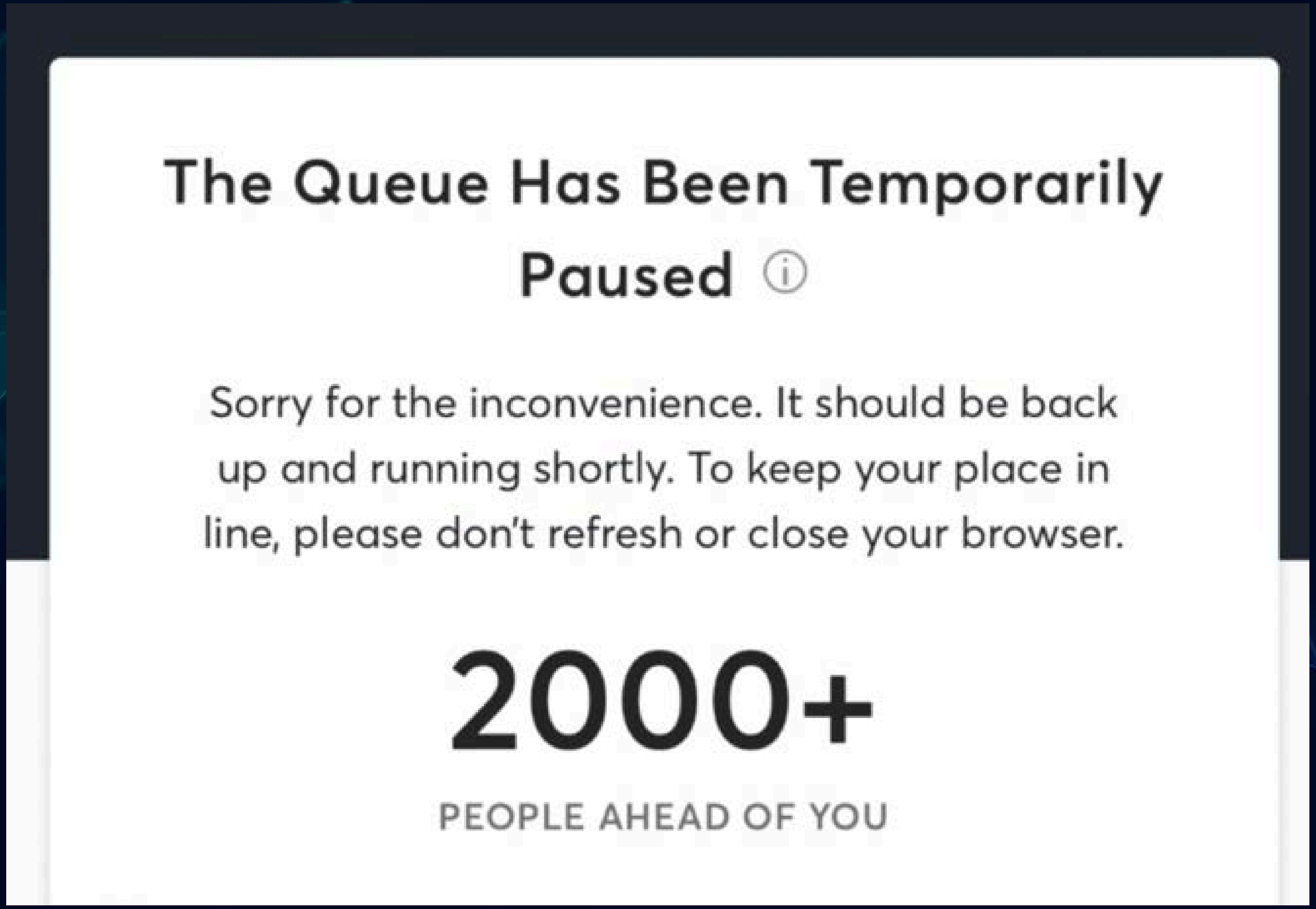

Load testing vs. Stress testing vs. Spike testing
Traffic spike
Breaking point
USERS
Expected load
TIME
—Load test
—Stress test
—Spike test
Queue·it


Count of records
60,000
50,000
40,000
30,000
20,000
10,000
0



reservix
dein ticketportal





The Queue Has Been Temporarily Paused ⓘ

Sorry for the inconvenience. It should be back up and running shortly. To keep your place in line, please don't refresh or close your browser.

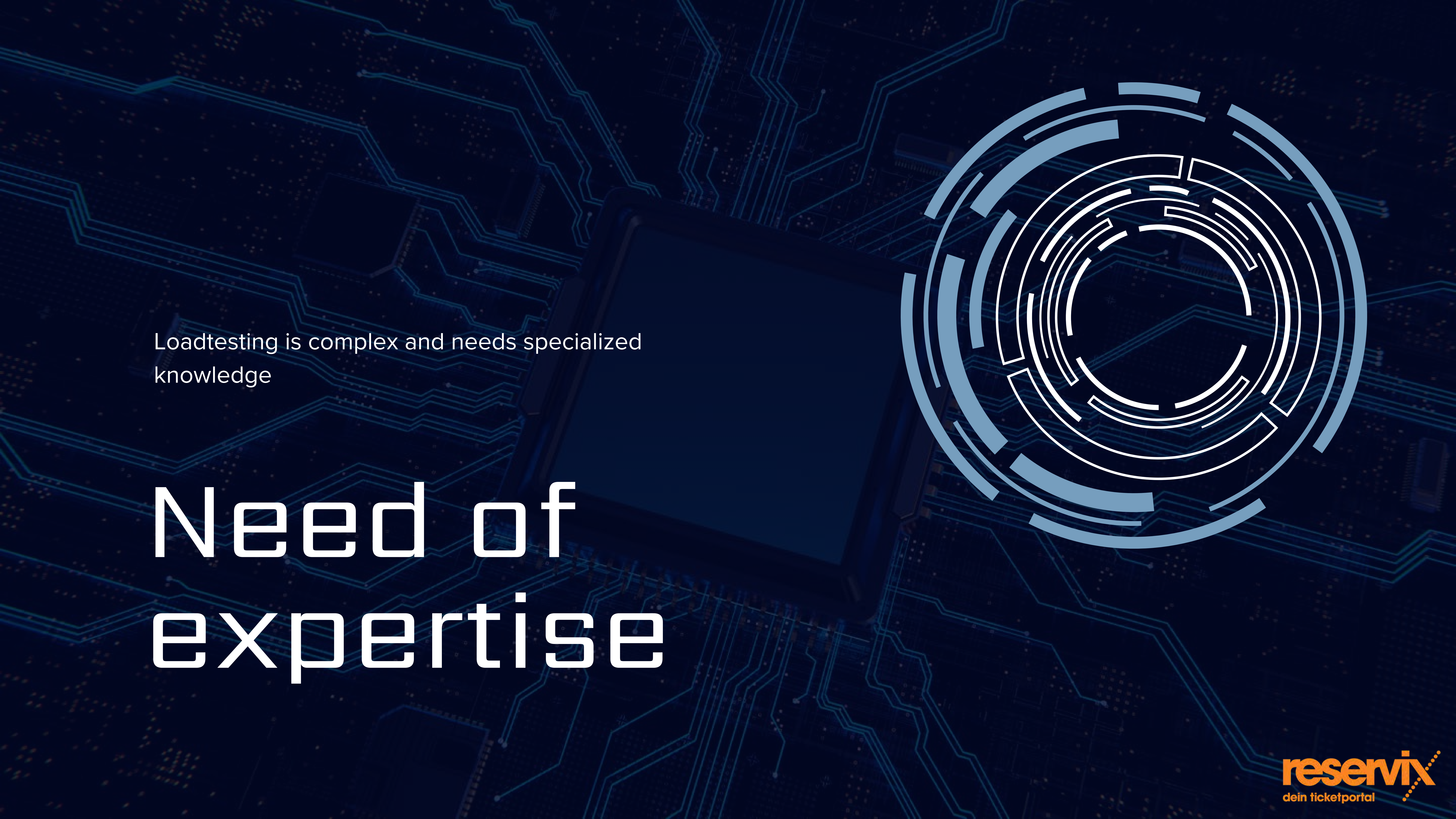
2000+

PEOPLE AHEAD OF YOU

How can I verify my webapp's reliability under heavy user load?

LOADTESTS

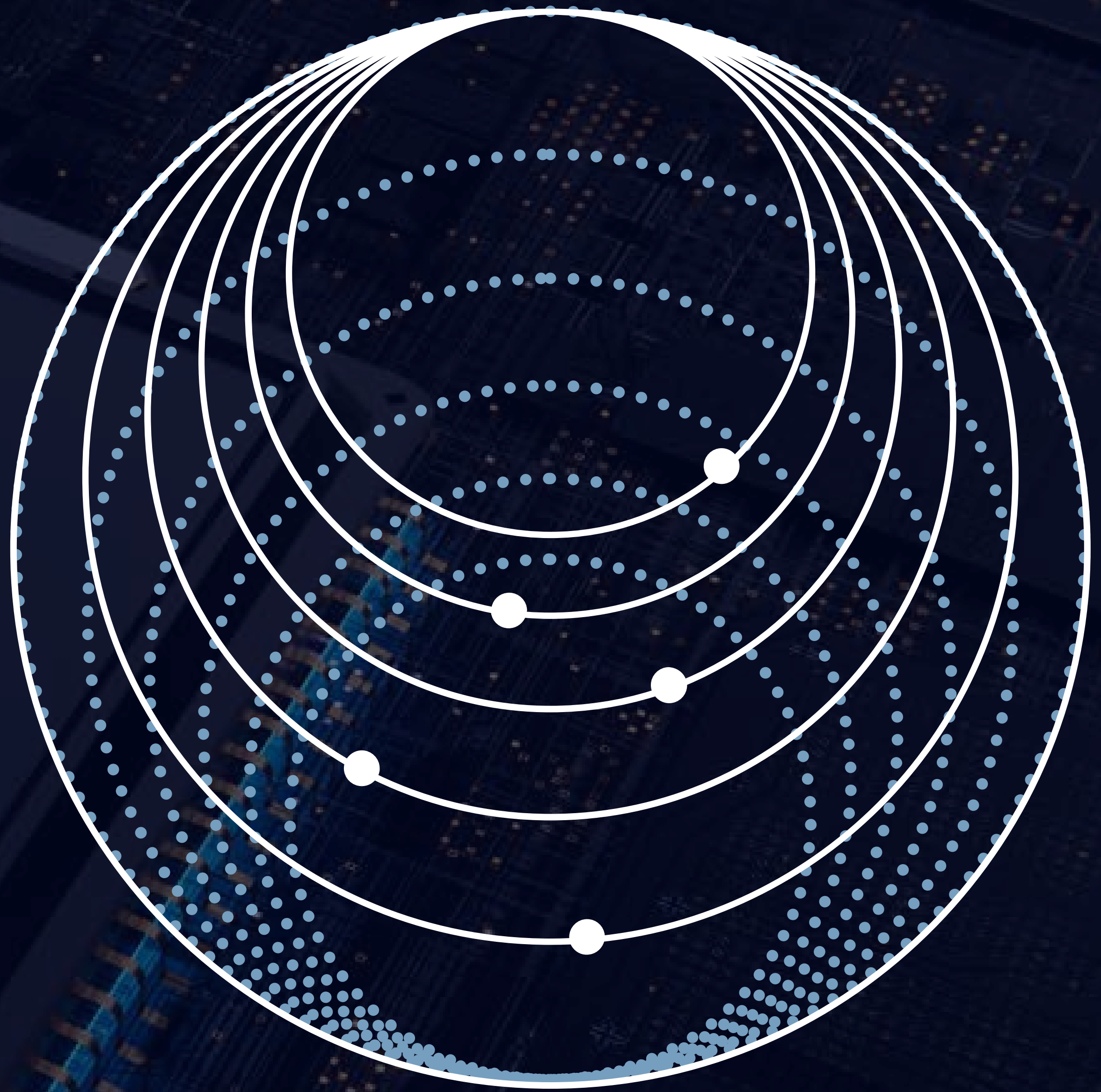
- simple answer -



Loadtesting is complex and needs specialized knowledge

Need of expertise

Motivation



Loadtesting for “everyone”

Requirements

Usability

Easy to use

Adaptibility

Integrating with existing and new testsuites

Feasibility

Hide complexity of loadtest environments

Requirements

Affordability

Ensure cost efficiency

Accessibility

Persistent and easy access to test results

Extensibility

Easy to customize

SaaS

AWS serverless components

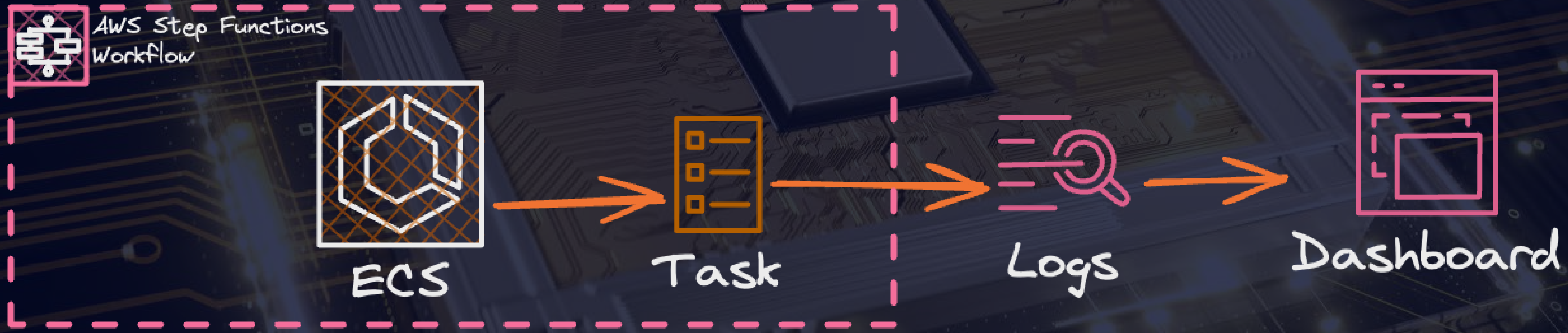
- AWS CDK
- K6
- Open Telemetry (ADOT)

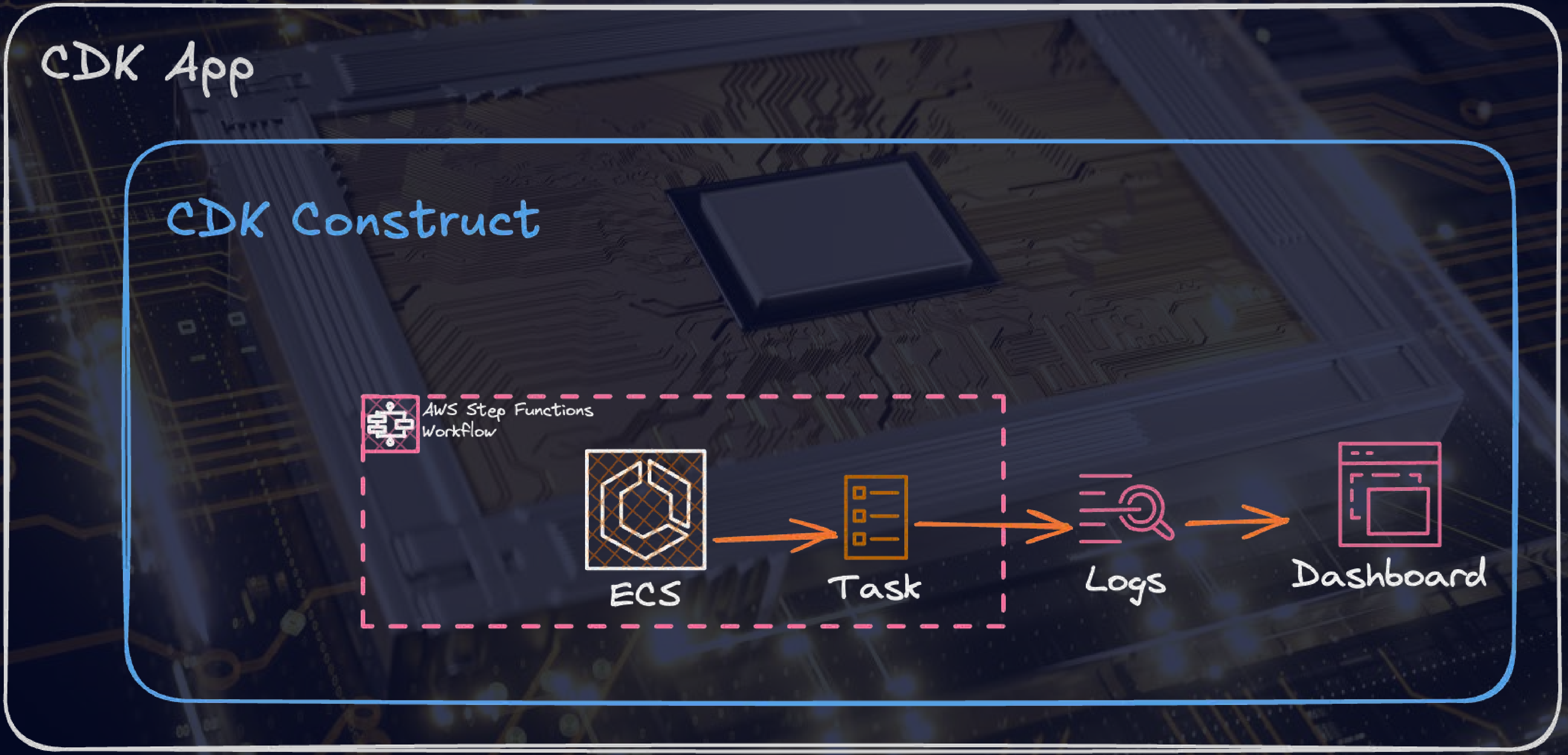
Toolchain

reservix
dein ticketportal

How-To Loadtest

- Launch the Loadtest runner
- Choose the correct settings
- Upload test scripts
- Run test suite
- Review result


AWS Step Functions Workflow
ECS
Task
Logs
Dashboard



CDK App

CDK Construct

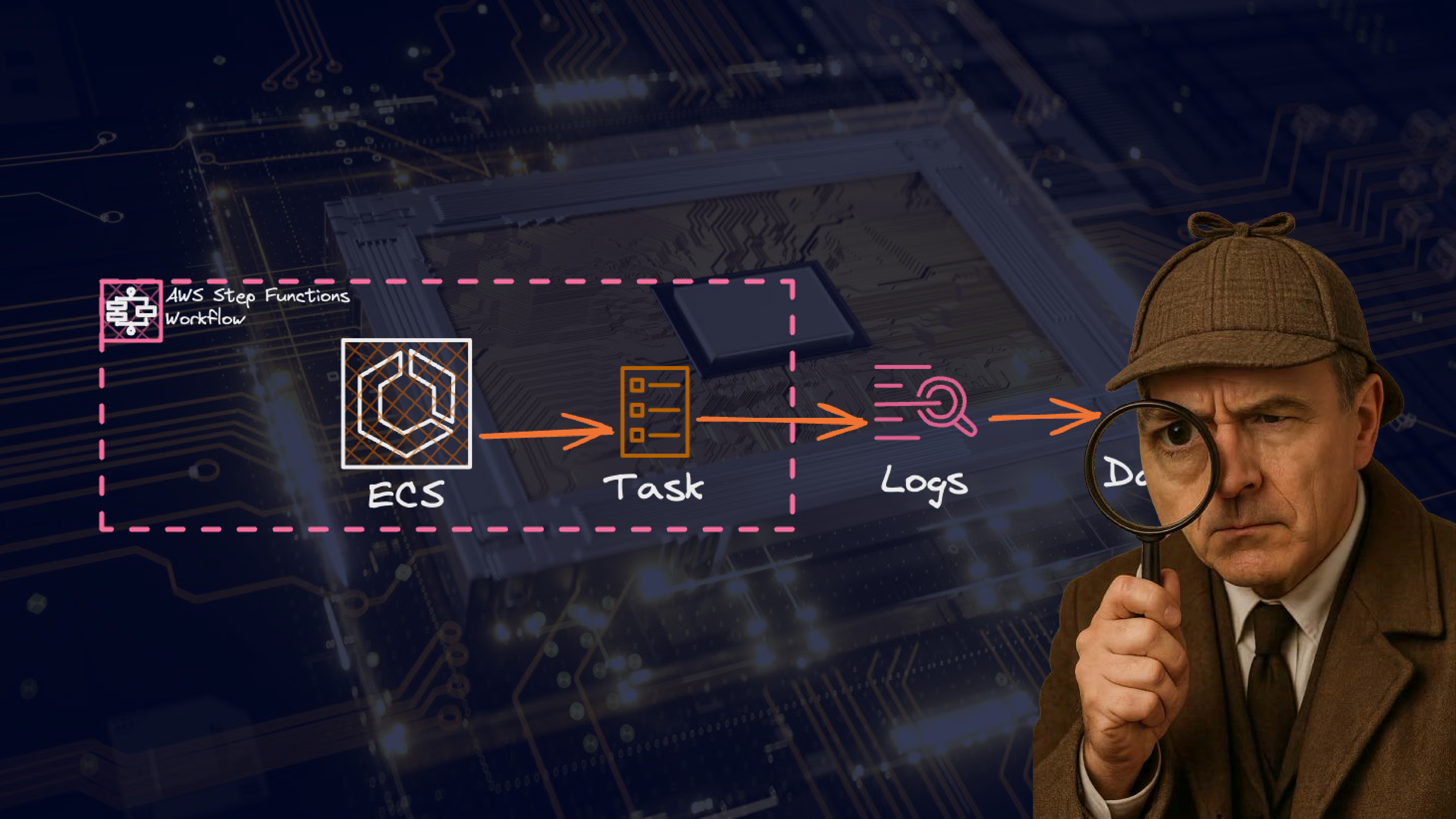
AWS Step Functions Workflow

ECS

Task

Logs

Dashboard


AWS Step Functions Workflow
ECS
Task
Logs
Do

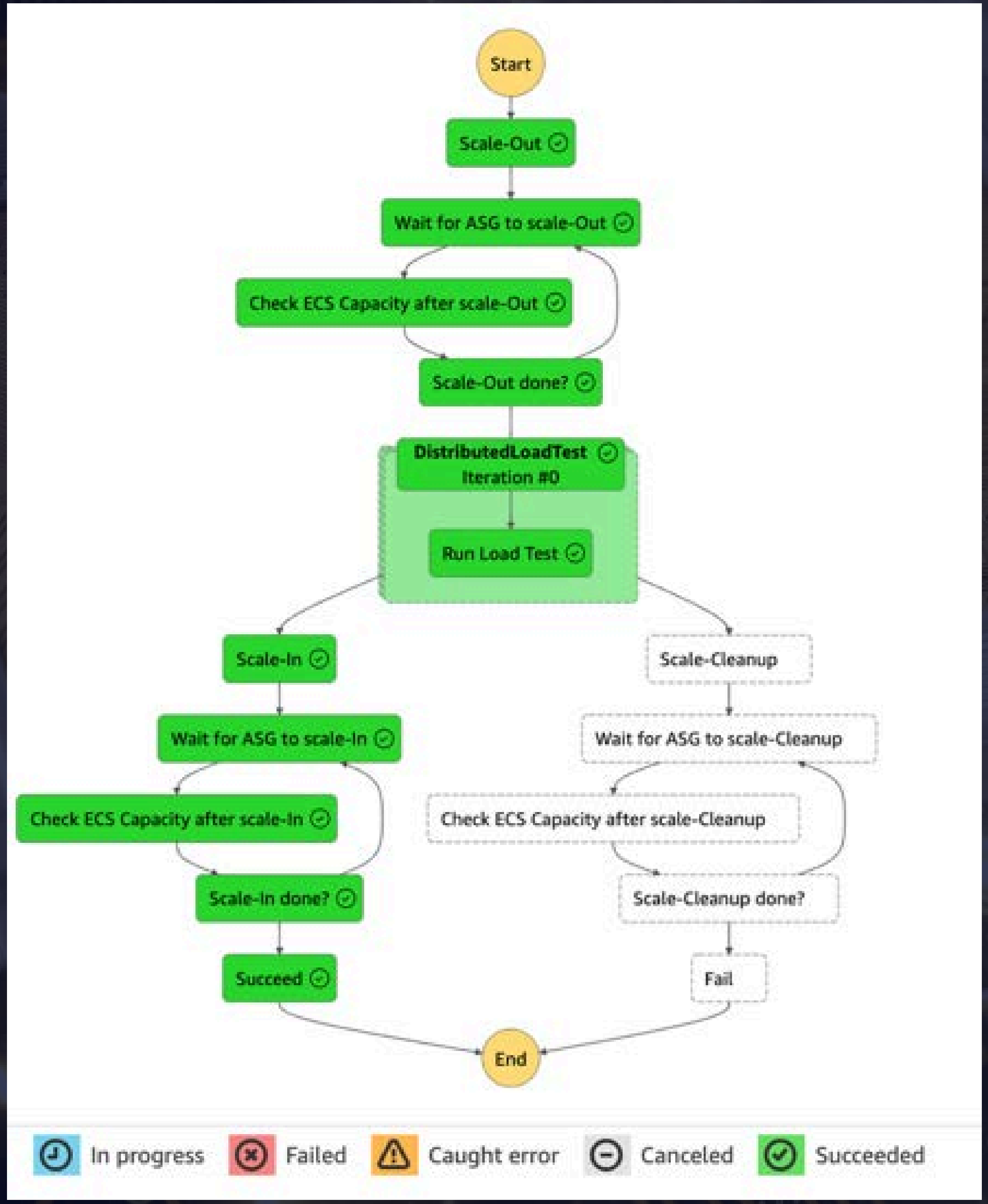
EC2 vs. Fargate

5.60$/ month	6.02$/ month
CPU burst	exact allocation
Resources limited by instance type	Fargate quotas
EC2 management	No operational overhead

EC2 vs. Fargate

5.60$/ month	6.02$/ month
CPU burst	exact allocation
Resources limited by instance type	Fargate quotas
EC2 management	No operational overhead

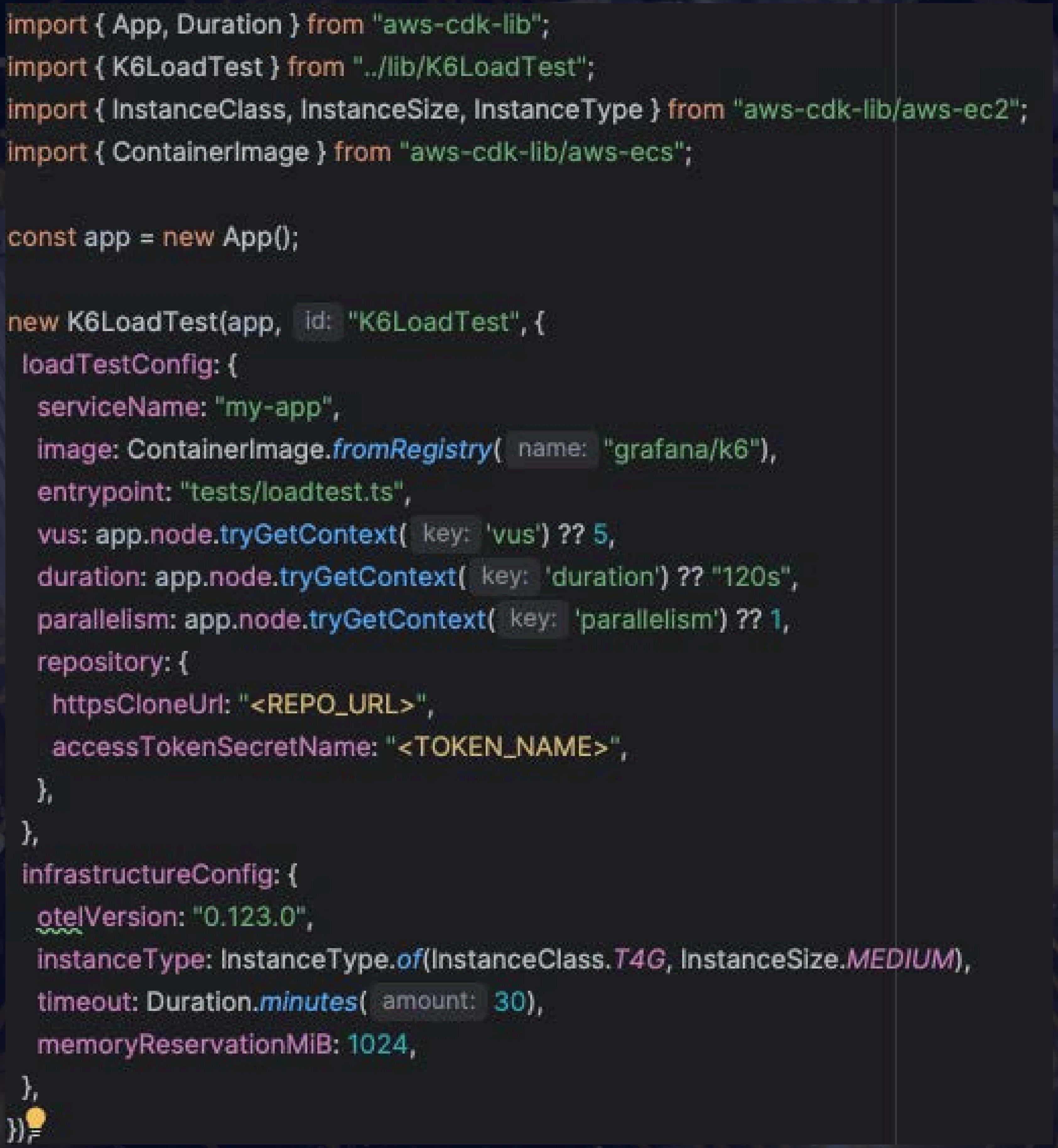
+


Start
Scale-Out
Wait for ASG to scale-Out
Check ECS Capacity after scale-Out
Scale-Out done?
DistributedLoadTest
Iteration #0
Run Load Test
Scale-In
Wait for ASG to scale-In
Check ECS Capacity after scale-In
Scale-In done?
Succeed
Scale-Cleanup
Wait for ASG to scale-Cleanup
Check ECS Capacity after scale-Cleanup
Scale-Cleanup done?
Fail
End
In progress
Failed
Caught error
Canceled
Succeeded

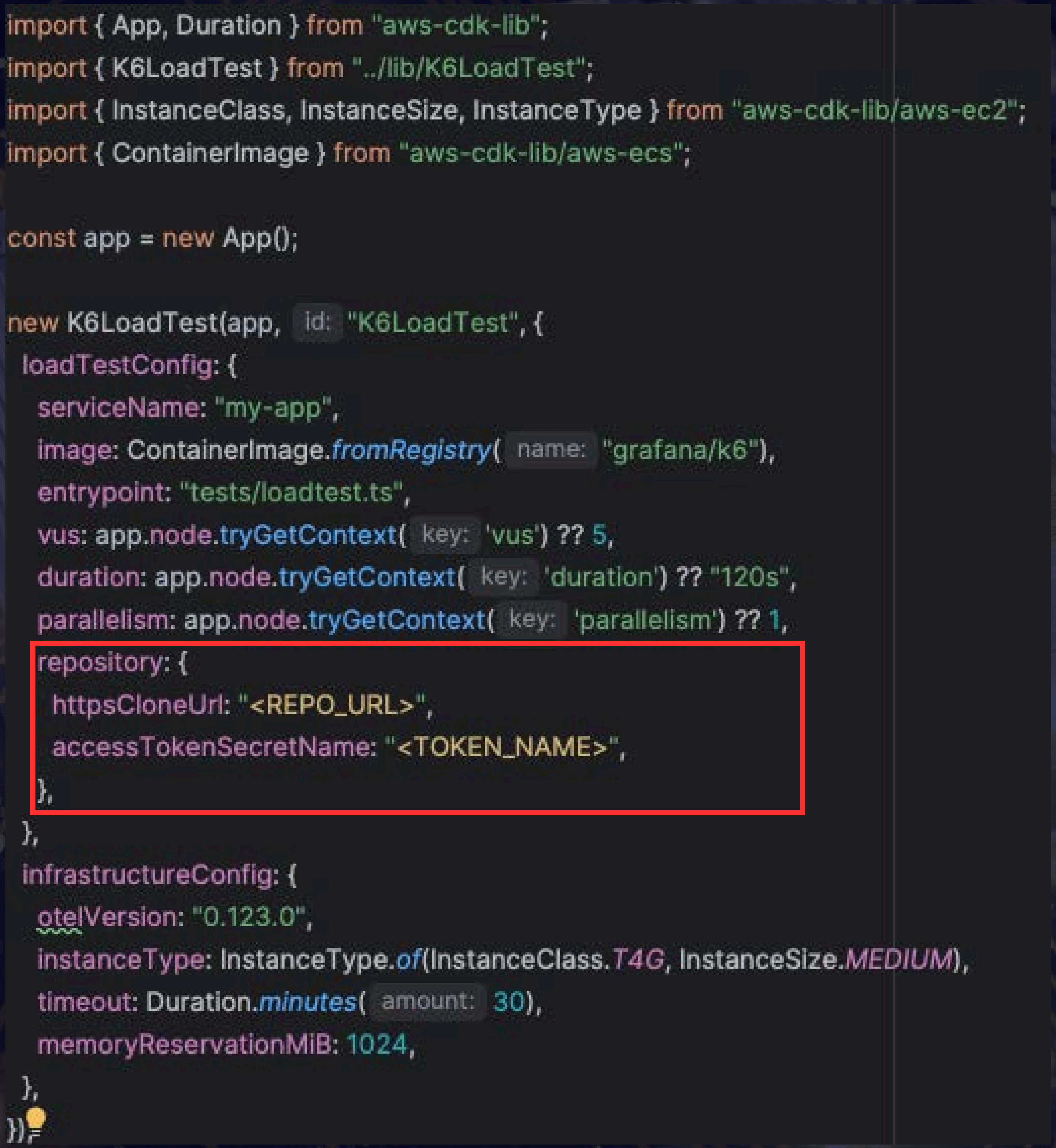
USABILITY

:

Upload Test scripts

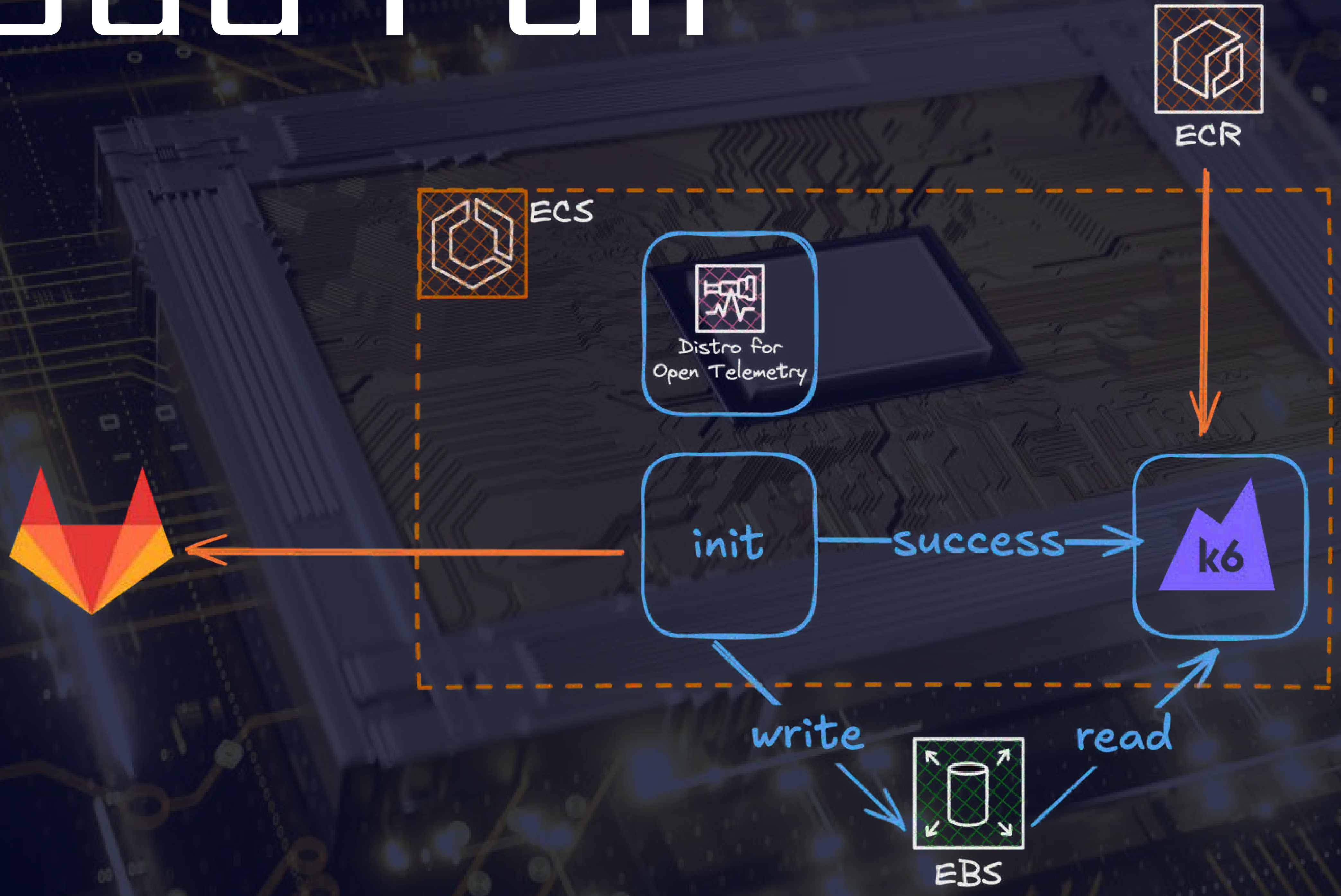


```
import { App, Duration } from "aws-cdk-lib";
import { K6LoadTest } from "../lib/K6LoadTest";
import { InstanceClass, InstanceSize, InstanceType } from "aws-cdk-lib/aws-ec2";
import { ContainerImage } from "aws-cdk-lib/aws-ecs";

const app = new App();

new K6LoadTest(app, id: "K6LoadTest", {
  loadTestConfig: {
    serviceName: "my-app",
    image: ContainerImage.fromRegistry( name: "grafana/k6"),
    entrypoint: "tests/loadtest.ts",
    vus: app.node.tryGetContext( key: 'vus') ?? 5,
    duration: app.node.tryGetContext( key: 'duration') ?? "120s",
    parallelism: app.node.tryGetContext( key: 'parallelism') ?? 1,
    repository: {
      httpsCloneUrl: "<REPO_URL>",
      accessTokenSecretName: "<TOKEN_NAME>",
    },
  },
  infrastructureConfig: {
    otelVersion: "0.123.0",
    instanceType: InstanceType.of(InstanceClass.T4G, InstanceSize.MEDIUM),
    timeout: Duration.minutes( amount: 30),
    memoryReservationMiB: 1024,
  },
});
```



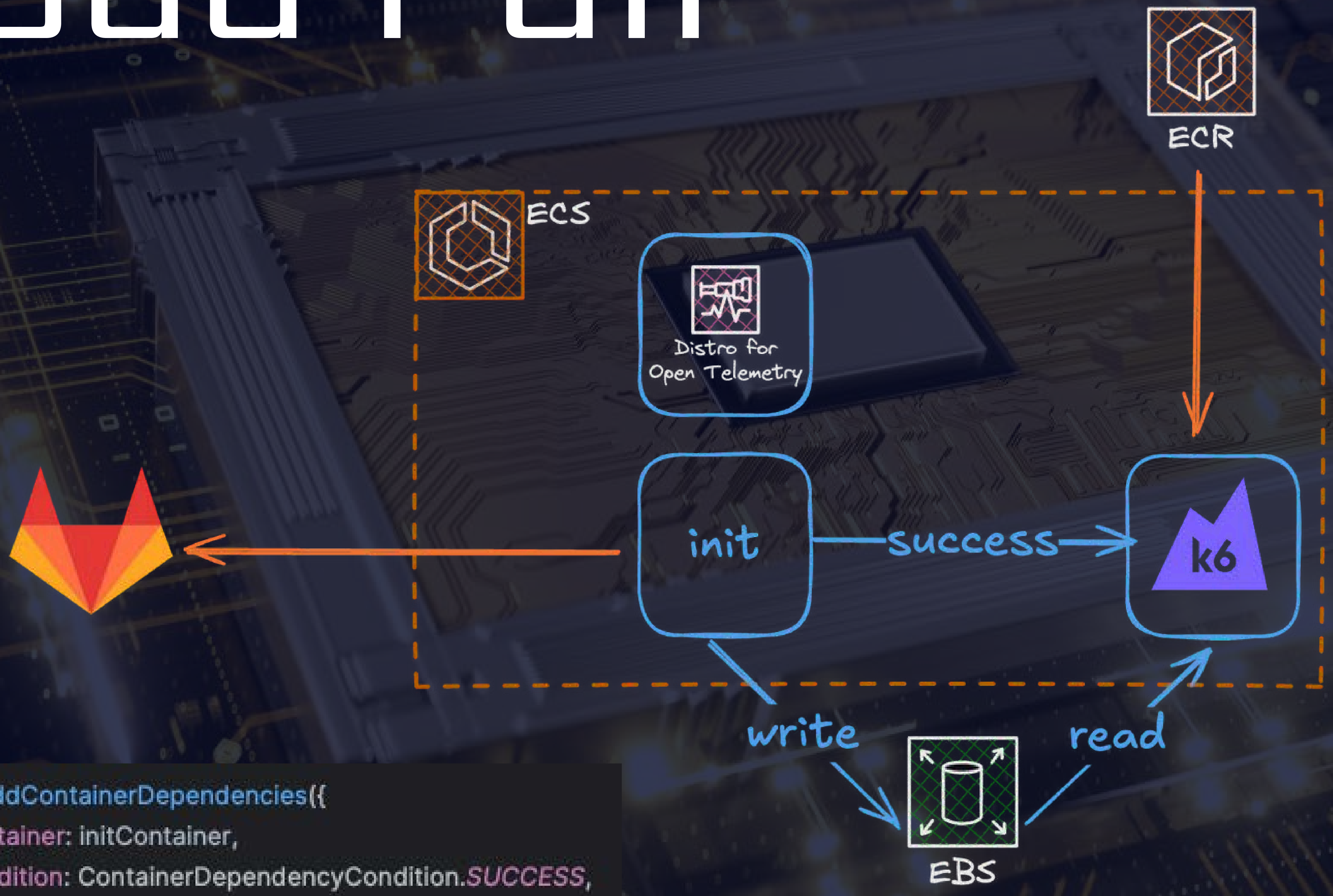
```
import { App, Duration } from "aws-cdk-lib";
import { K6LoadTest } from "../lib/K6LoadTest";
import { InstanceClass, InstanceSize, InstanceType } from "aws-cdk-lib/aws-ec2";
import { ContainerImage } from "aws-cdk-lib/aws-ecs";

const app = new App();

new K6LoadTest(app, id: "K6LoadTest", {
  loadTestConfig: {
    serviceName: "my-app",
    image: ContainerImage.fromRegistry( name: "grafana/k6"),
    entrypoint: "tests/loadtest.ts",
    vus: app.node.tryGetContext( key: 'vus') ?? 5,
    duration: app.node.tryGetContext( key: 'duration') ?? "120s",
    parallelism: app.node.tryGetContext( key: 'parallelism') ?? 1,
    repository: {
      httpsCloneUrl: "<REPO_URL>",
      accessTokenSecretName: "<TOKEN_NAME>",
    },
  },
  infrastructureConfig: {
    otelVersion: "0.123.0",
    instanceType: InstanceType.of(InstanceClass.T4G, InstanceSize.MEDIUM),
    timeout: Duration.minutes( amount: 30),
    memoryReservationMiB: 1024,
  },
});
```

~~Upload~~ Pull



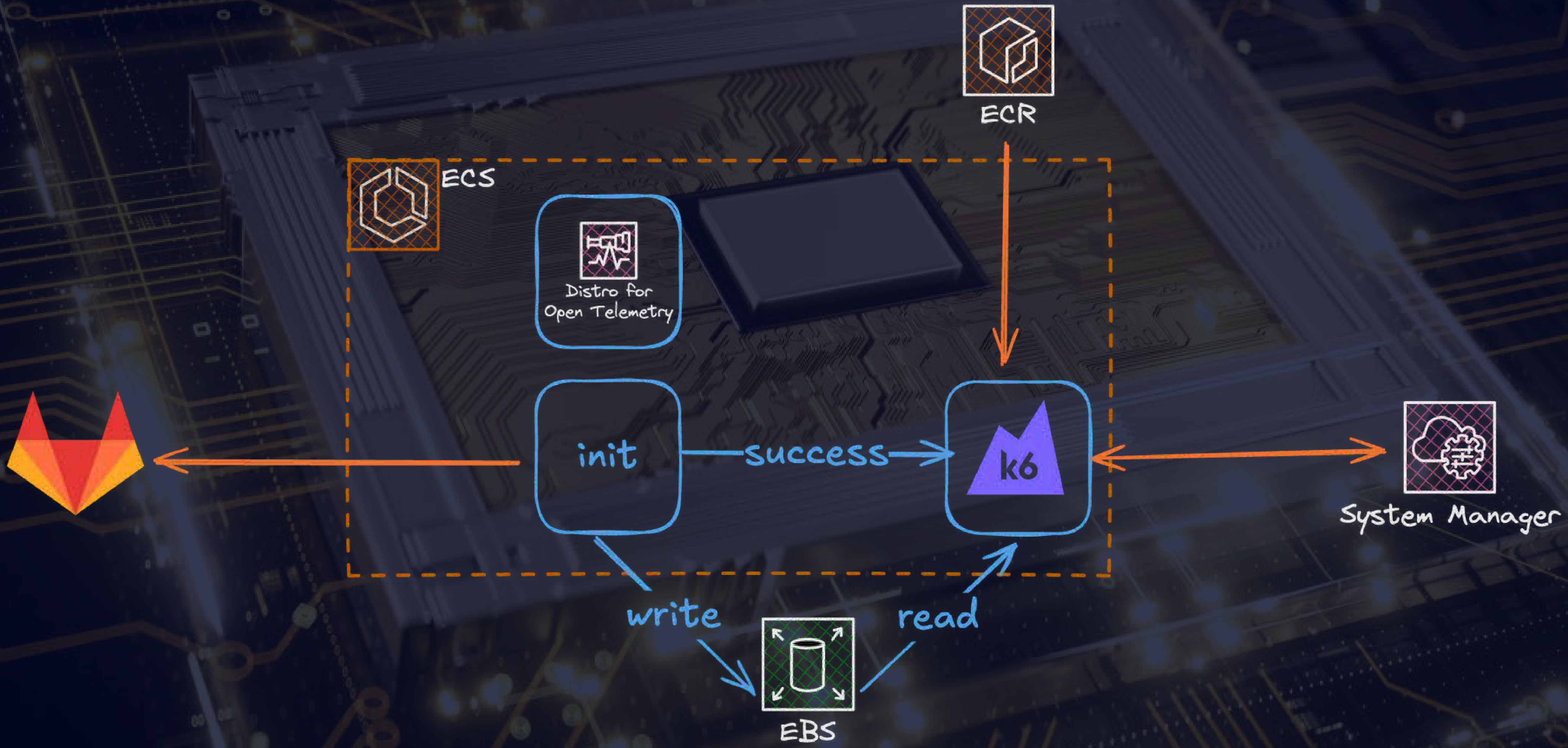
reservix
dein ticketportal

~~Upload~~ Pull



```
k6.addContainerDependencies({
  container: initContainer,
  condition: ContainerDependencyCondition.SUCCESS,
});
```

~~Upload~~ Pull

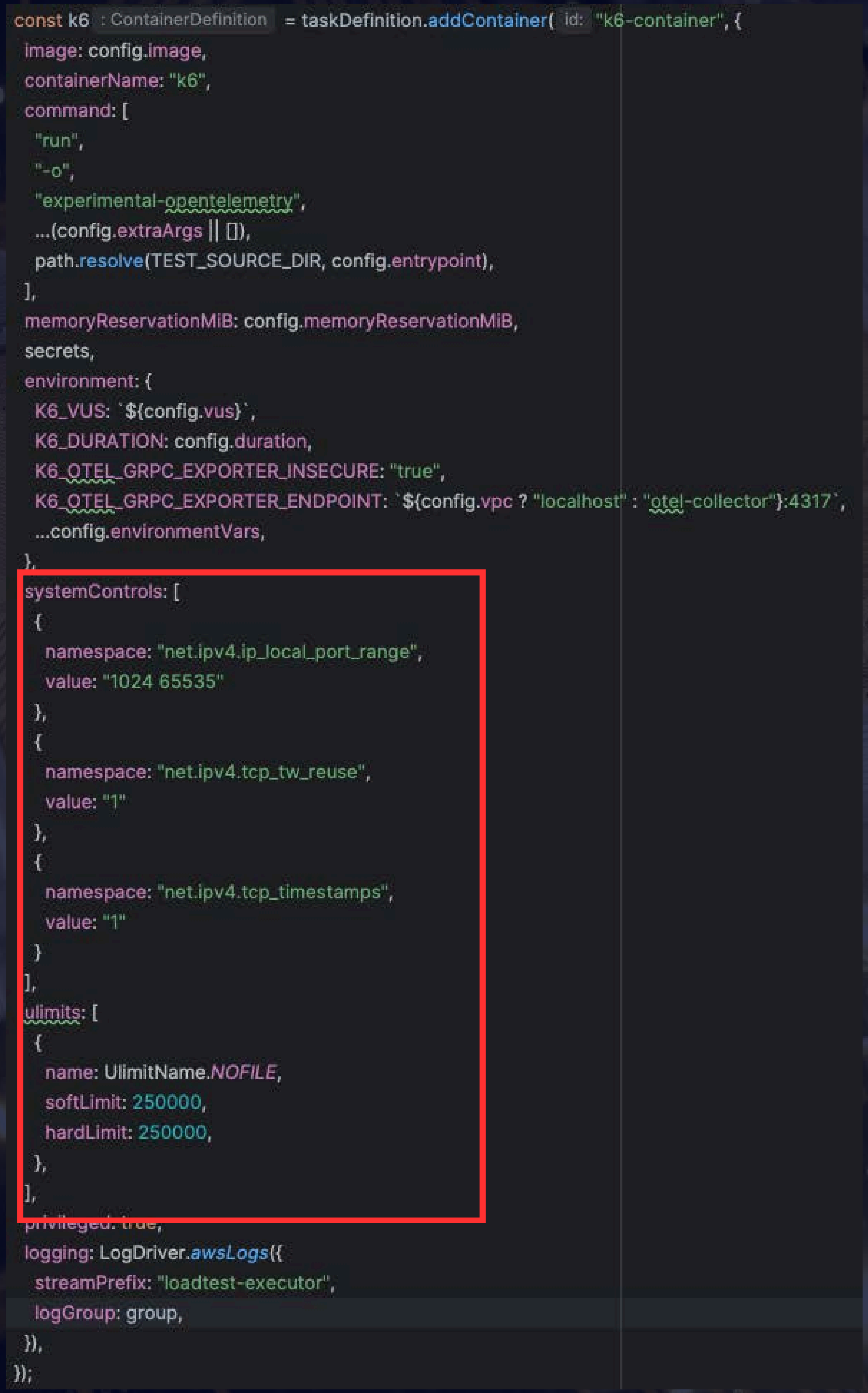


reservix
dein ticketportal

Feasibility

:

Run test suite

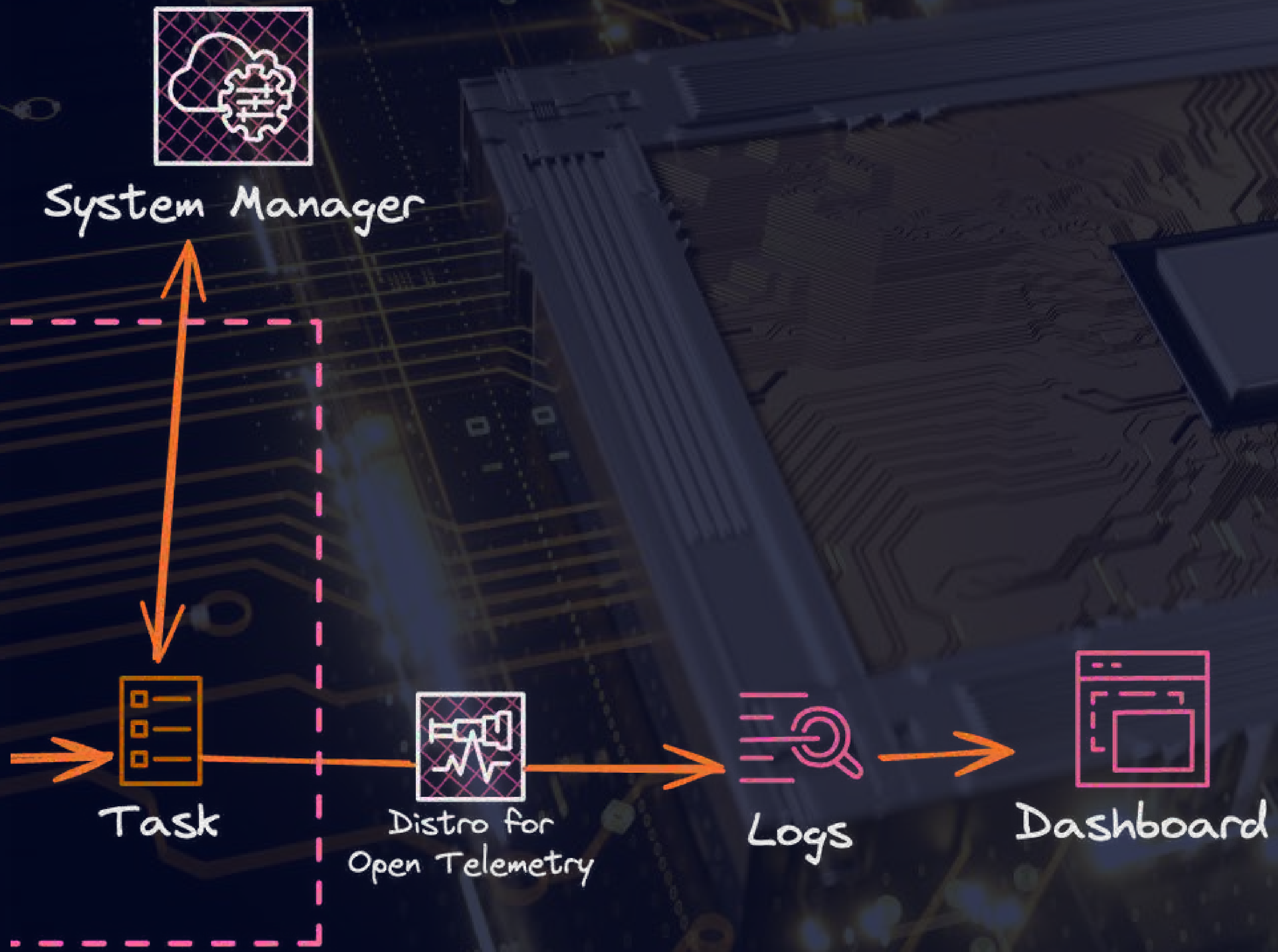


```
const k6 : ContainerDefinition = taskDefinition.addContainer( id: "k6-container", {
  image: config.image,
  containerName: "k6",
  command: [
    "run",
    "-o",
    "experimental-opentelemetry",
    ...(config.extraArgs || []),
    path.resolve(TEST_SOURCE_DIR, config.entrypoint),
  ],
  memoryReservationMiB: config.memoryReservationMiB,
  secrets,
  environment: {
    K6_VUS: `${config.vus}`,
    K6_DURATION: config.duration,
    K6_OTEL_GRPC_EXPORTER_INSECURE: "true",
    K6_OTEL_GRPC_EXPORTER_ENDPOINT: `${config.vpc ? "localhost" : "otel-collector"}:4317`,
    ...config.environmentVars,
  },
  systemControls: [
    {
      namespace: "net.ipv4.ip_local_port_range",
      value: "1024 65535"
    },
    {
      namespace: "net.ipv4.tcp_tw_reuse",
      value: "1"
    },
    {
      namespace: "net.ipv4.tcp_timestamps",
      value: "1"
    }
  ],
  ulimits: [
    {
      name: UlimitName.NOFILE,
      softLimit: 250000,
      hardLimit: 250000,
    },
  ],
  privileged: true,
  logging: LogDriver.awsLogs({
    streamPrefix: "loadtest-executor",
    logGroup: group,
  }),
});
```

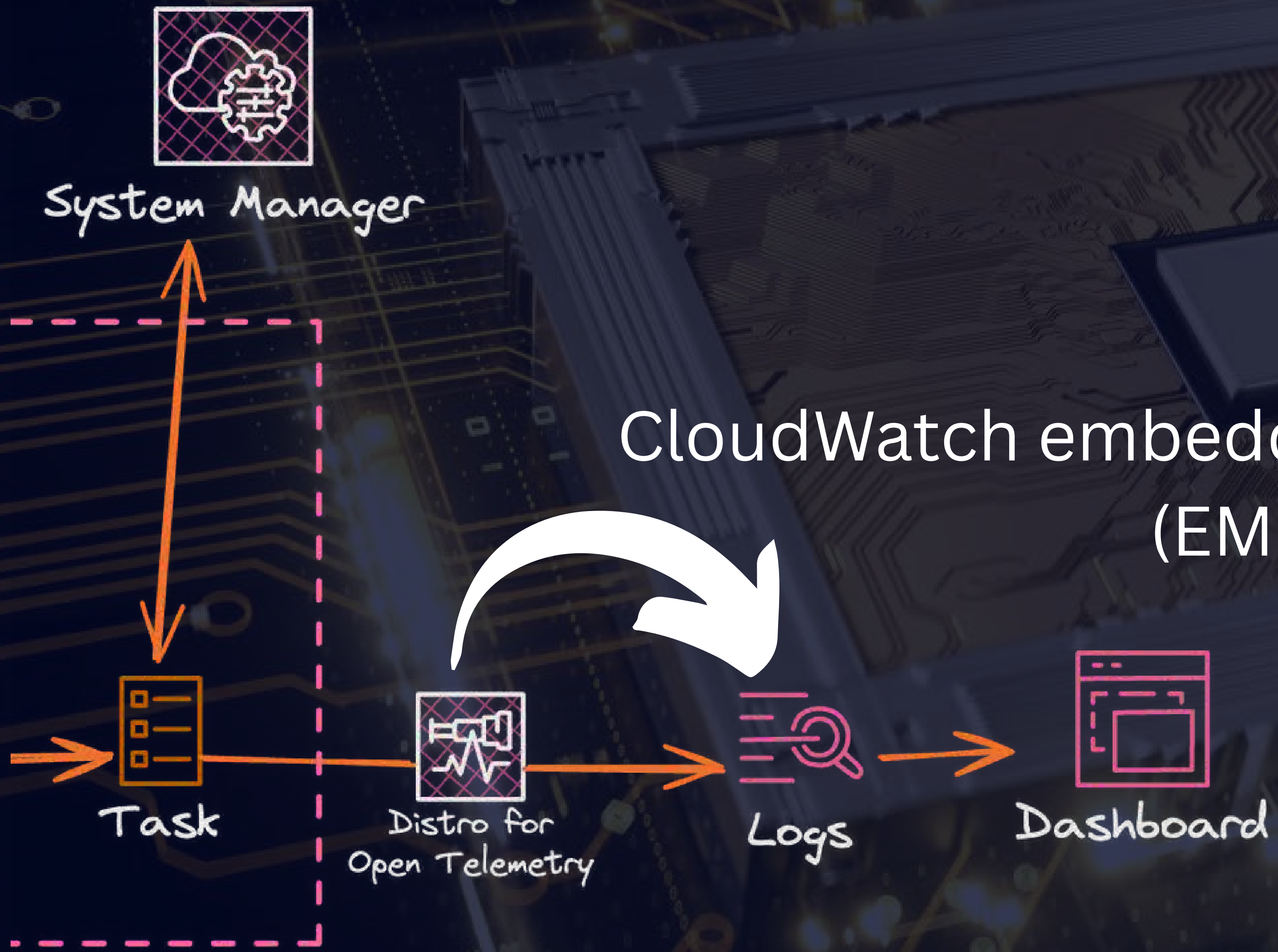
Accessibility

:

Review result

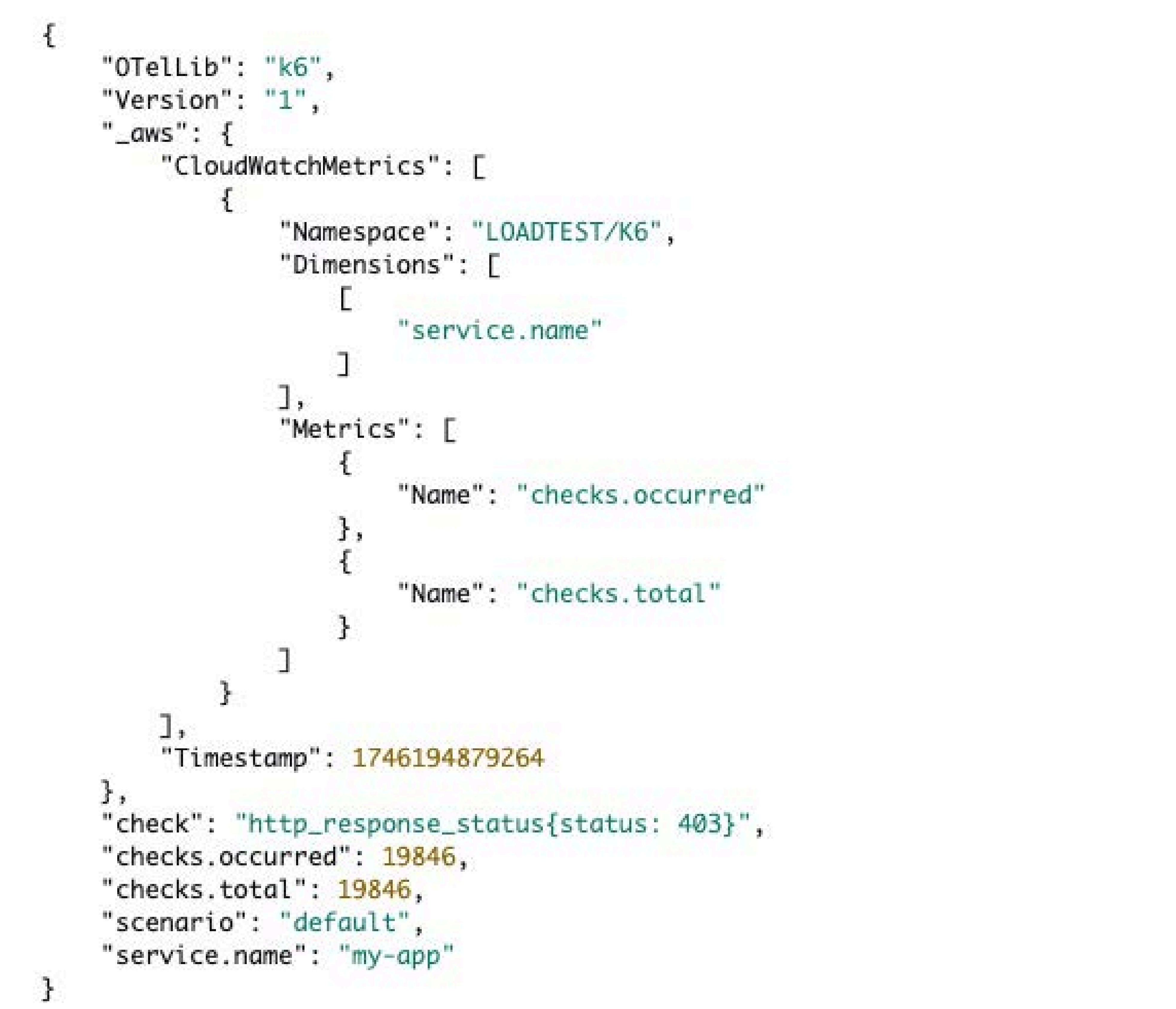


reservix
dein ticketportal



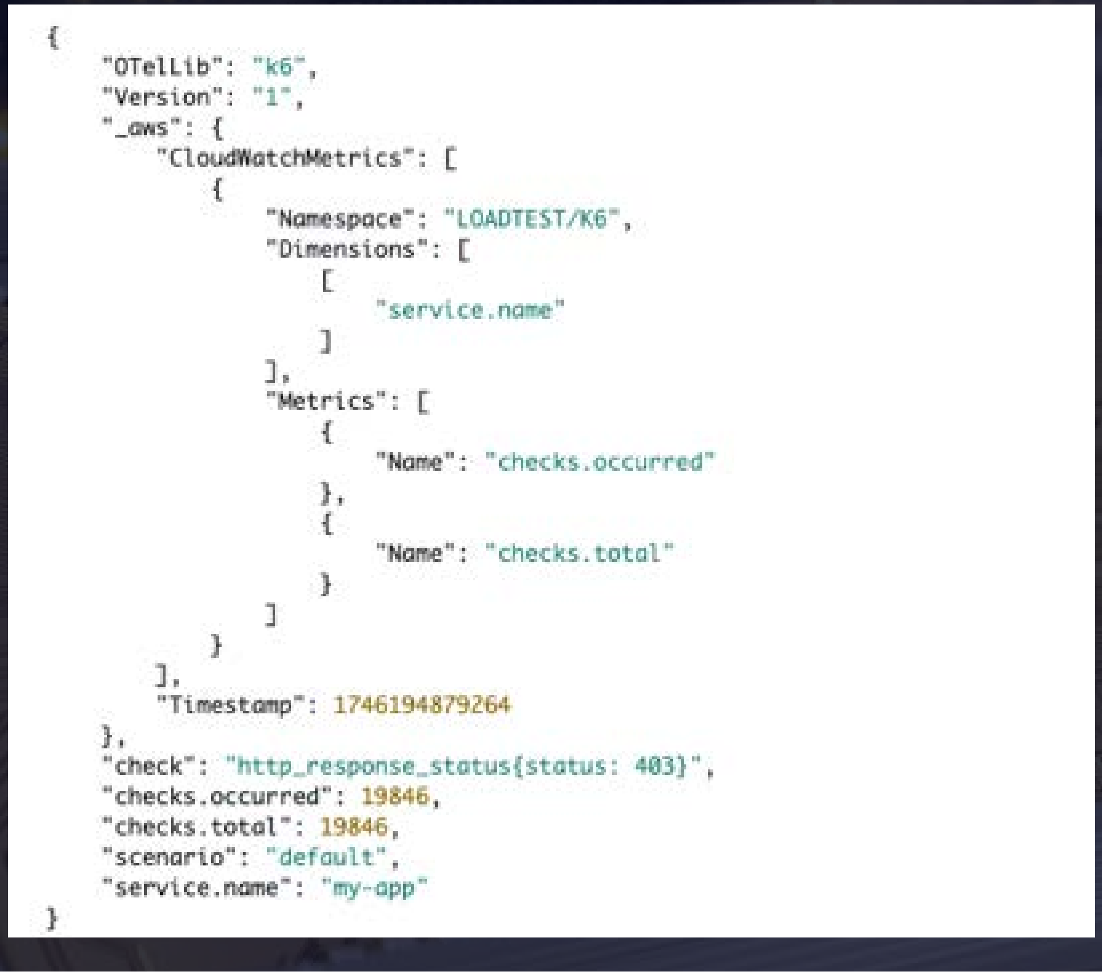
CloudWatch embedded metric format (EMF)

CloudWatch embedded metric format

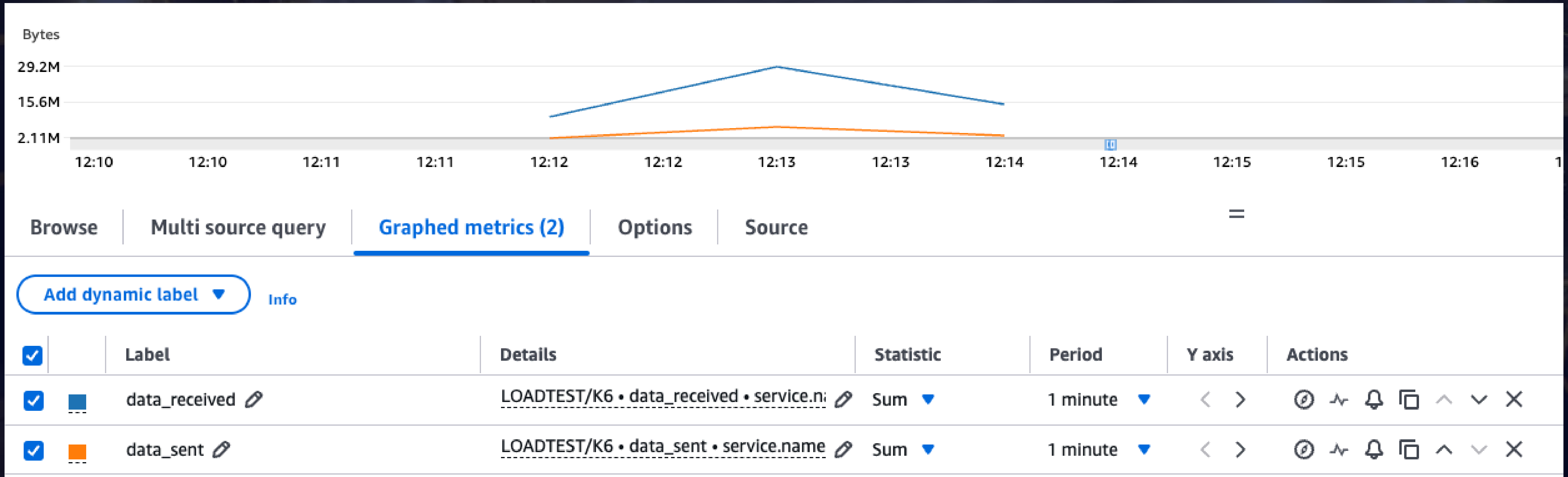


```
{
    "OTelLib": "k6",
    "Version": "1",
    "_aws": {
        "CloudWatchMetrics": [
            {
                "Namespace": "LOADTEST/K6",
                "Dimensions": [
                    [
                        "service.name"
                    ]
                ],
                "Metrics": [
                    {
                        "Name": "checks.occurred"
                    },
                    {
                        "Name": "checks.total"
                    }
                ]
            }
        ],
        "Timestamp": 1746194879264
    },
    "check": "http_response_status{status: 403}",
    "checks.occurred": 19846,
    "checks.total": 19846,
    "scenario": "default",
    "service.name": "my-app"
}
```

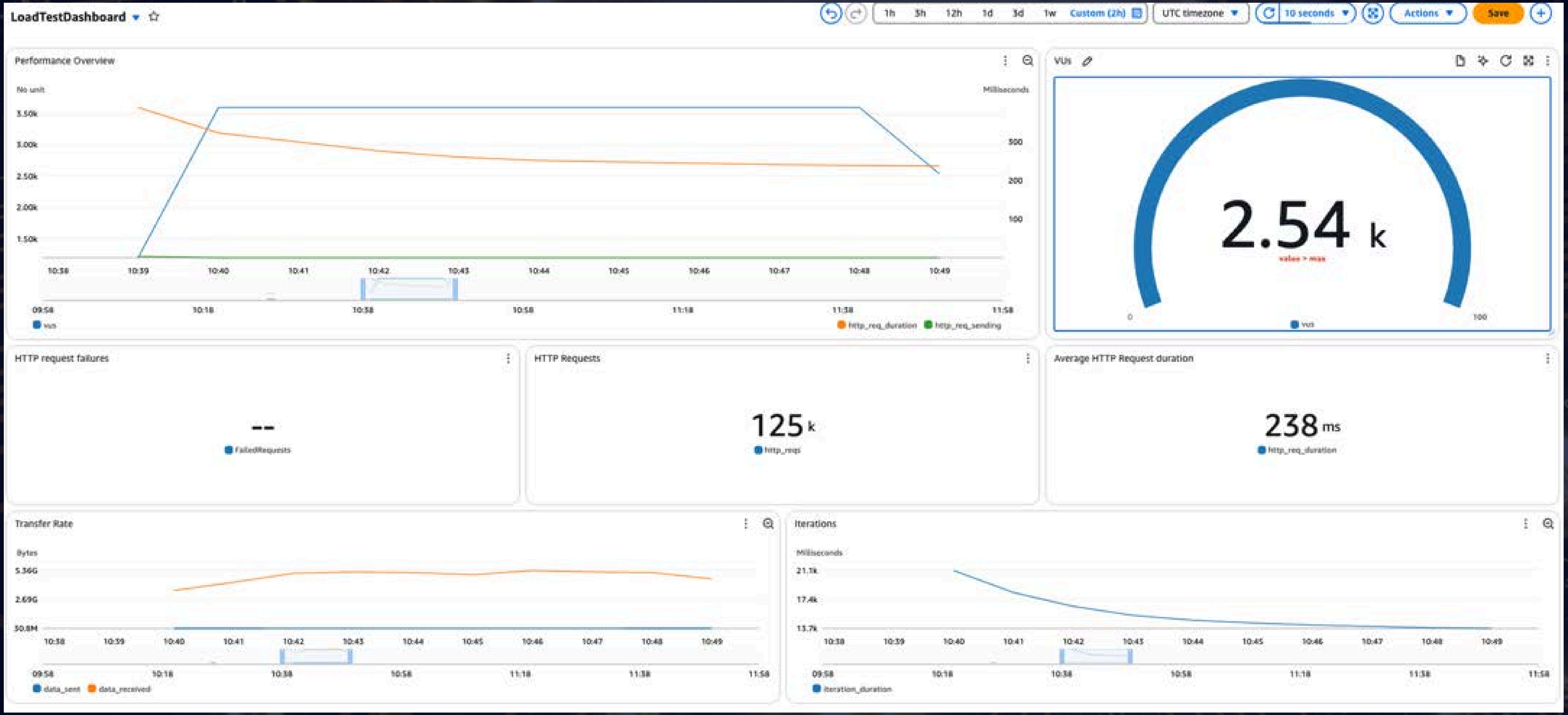
CloudWatch embedded metric format

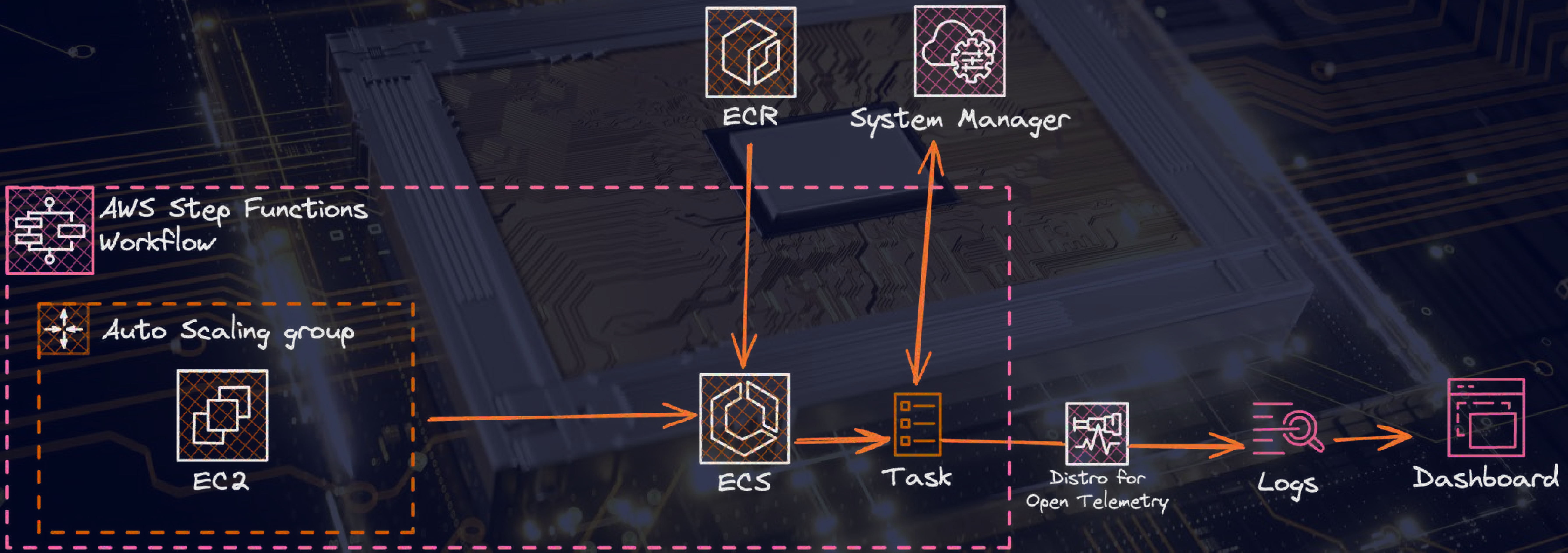


```
{
    "OTelLib": "k6",
    "Version": "1",
    "_aws": {
        "CloudWatchMetrics": [
            {
                "Namespace": "LOADTEST/K6",
                "Dimensions": [
                    [
                        "service.name"
                    ]
                ],
                "Metrics": [
                    {
                        "Name": "checks.occurred"
                    },
                    {
                        "Name": "checks.total"
                    }
                ]
            }
        ],
        "Timestamp": 1746194879264
    },
    "check": "http_response_status{status: 403}",
    "checks.occurred": 19846,
    "checks.total": 19846,
    "scenario": "default",
    "service.name": "my-app"
}
```



☑		Label	Details	Statistic	Period	Y axis	Actions
☑	🟦	data_received	LOADTEST/K6 • data_received • service.n	Sum ▼	1 minute ▼	< >	
☑	🟧	data_sent	LOADTEST/K6 • data_sent • service.name	Sum ▼	1 minute ▼	< >	


LoadTestDashboard
1h
3h
12h
1d
3d
1w
Custom (2h)
UTC timezone
10 seconds
Actions
Save
Performance Overview
No unit
3.50k
3.00k
2.50k
2.00k
1.50k
Milliseconds
300
200
100
10:38
10:39
10:40
10:41
10:42
10:43
10:44
10:45
10:46
10:47
10:48
10:49
09:58
10:18
10:38
10:58
11:18
11:38
11:58
vus
http_req_duration
http_req_sending
VUs
2.54 k
value > max
0
100
vus
HTTP request failures
--
FailedRequests
HTTP Requests
125 k
http_reqs
Average HTTP Request duration
238 ms
http_req_duration
Transfer Rate
Bytes
5.36G
2.69G
30.8M
10:38
10:39
10:40
10:41
10:42
10:43
10:44
10:45
10:46
10:47
10:48
10:49
09:58
10:18
10:38
10:58
11:18
11:38
11:58
data_sent
data_received
Iterations
Milliseconds
21.1k
17.4k
13.7k
10:38
10:39
10:40
10:41
10:42
10:43
10:44
10:45
10:46
10:47
10:48
10:49
09:58
10:18
10:38
10:58
11:18
11:38
11:58
iteration_duration


ECR
System Manager
AWS Step Functions
Workflow
Auto Scaling group
EC2
ECS
Task
Distro for
Open Telemetry
Logs
Dashboard

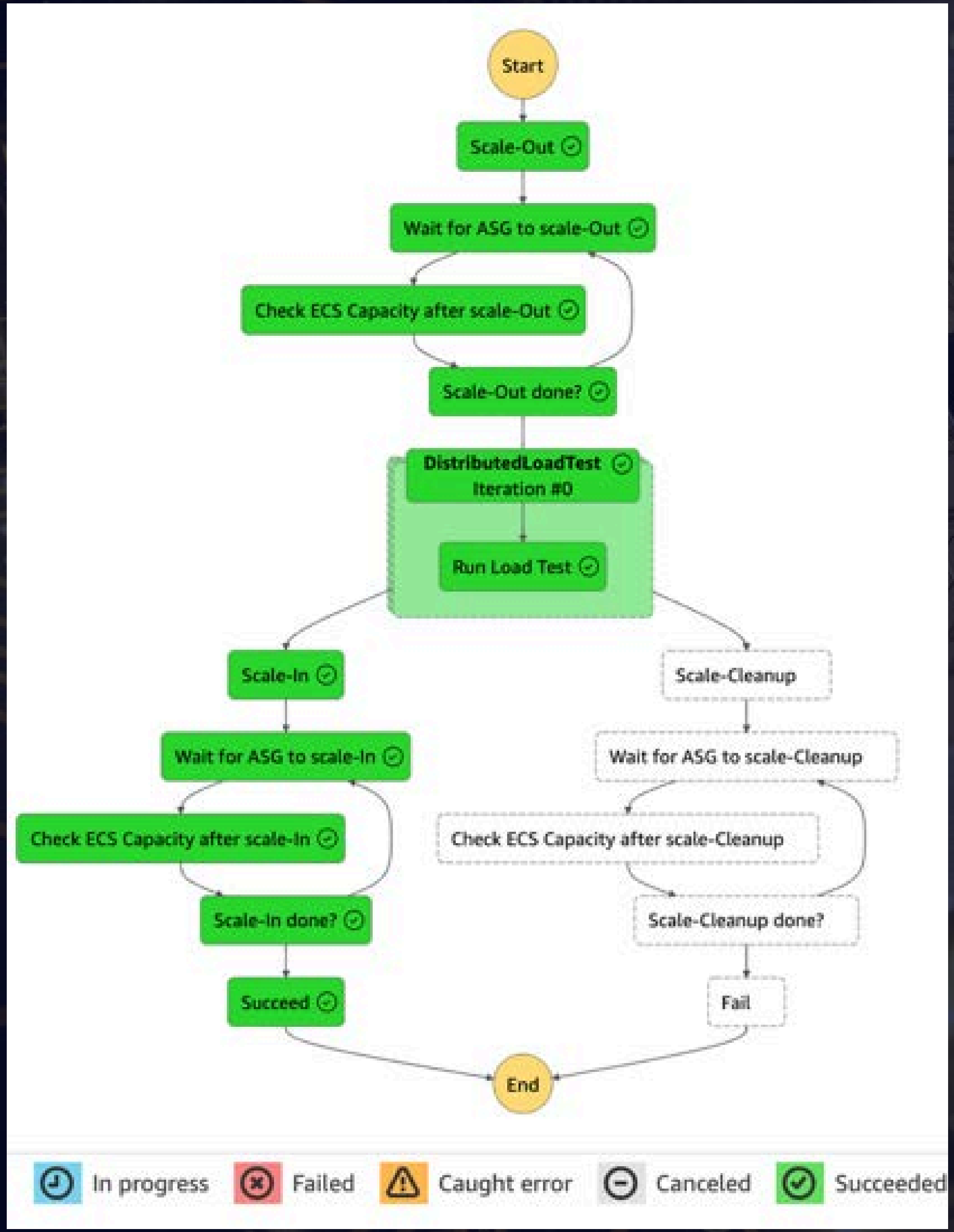
Usability

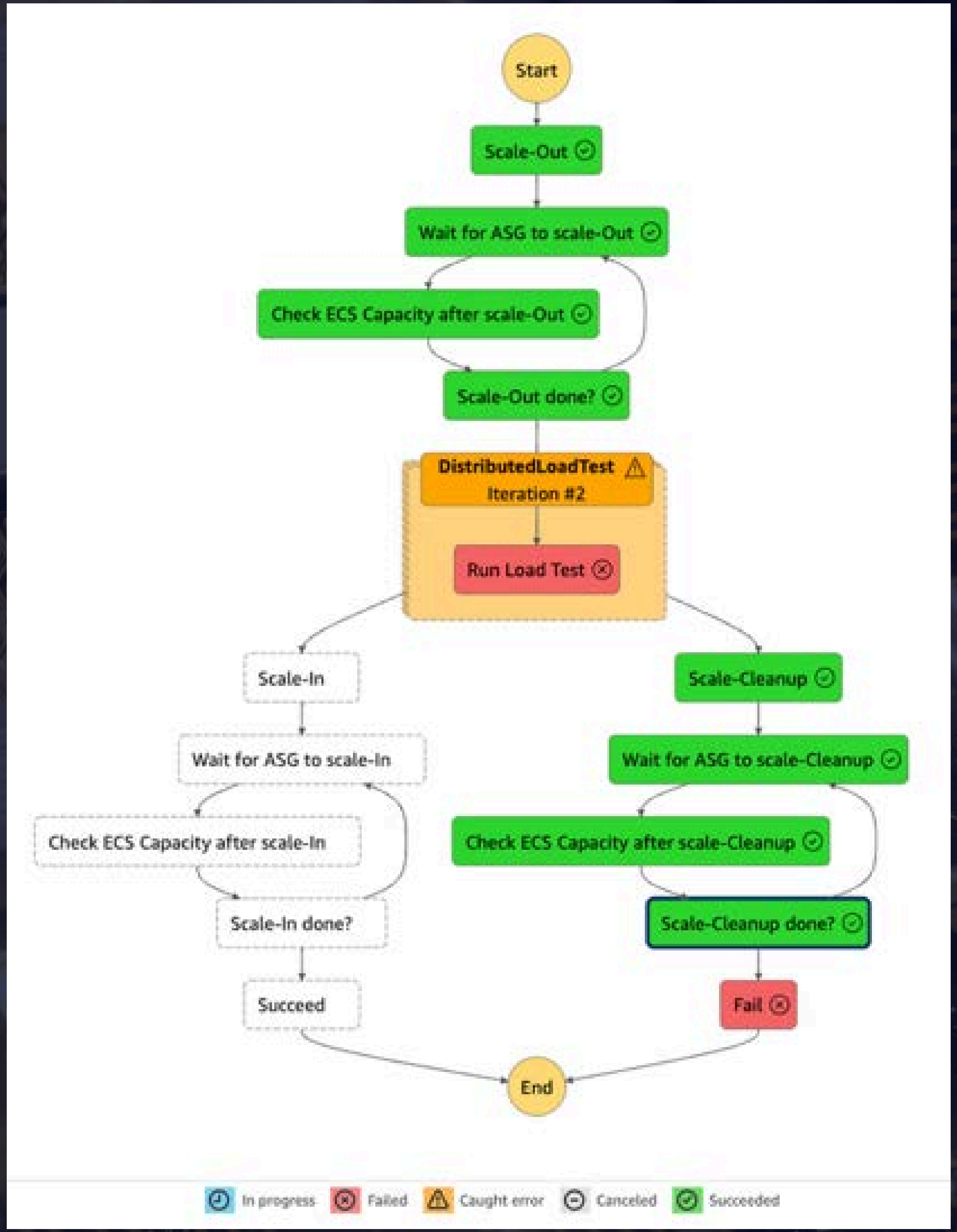
:

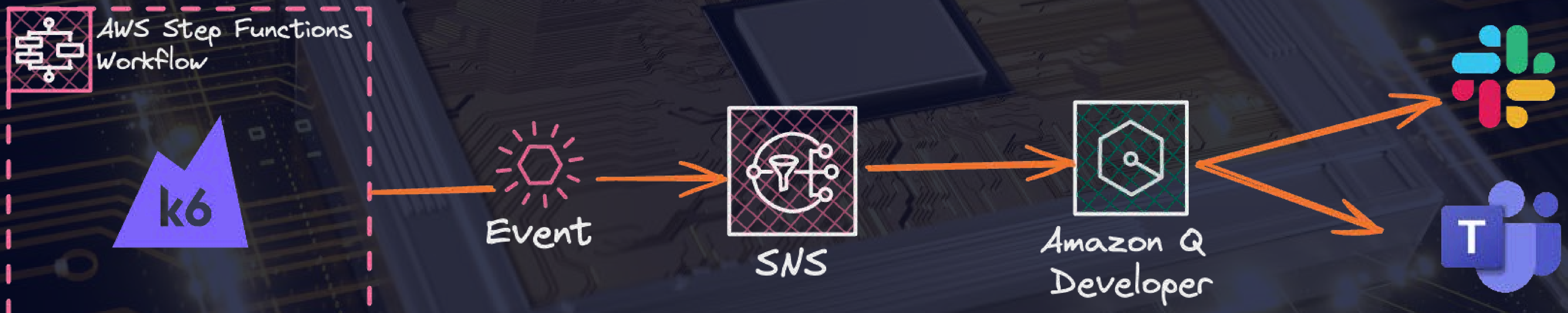
Loadtesting for “everyone”

Console Home
75%

AWS CLI


Start
Scale-Out
Wait for ASG to scale-Out
Check ECS Capacity after scale-Out
Scale-Out done?
DistributedLoadTest
Iteration #0
Run Load Test
Scale-In
Wait for ASG to scale-In
Check ECS Capacity after scale-In
Scale-In done?
Succeed
Scale-Cleanup
Wait for ASG to scale-Cleanup
Check ECS Capacity after scale-Cleanup
Scale-Cleanup done?
Fail
End
In progress
Failed
Caught error
Canceled
Succeeded

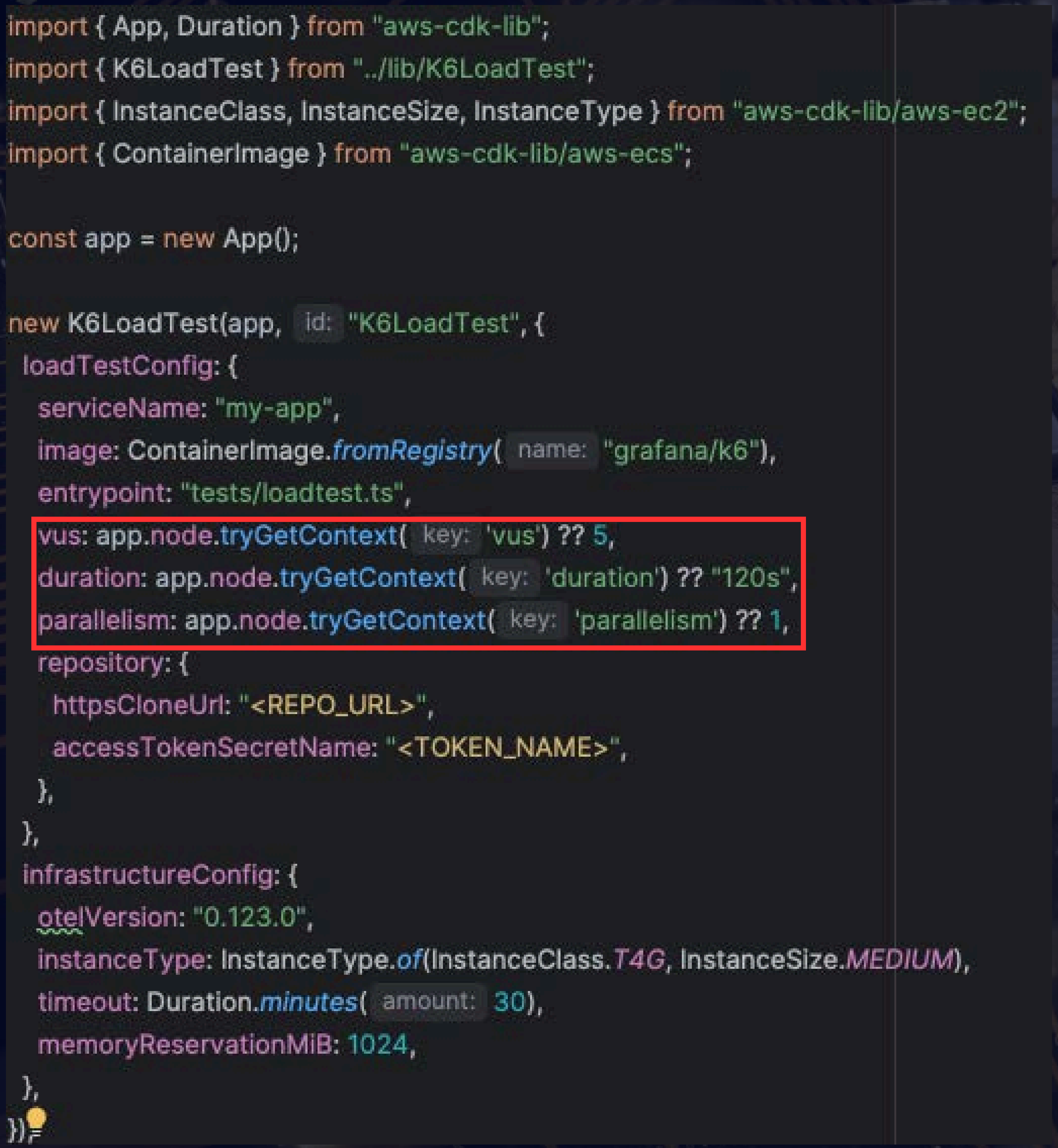

Start
Scale-Out
Wait for ASG to scale-Out
Check ECS Capacity after scale-Out
Scale-Out done?
DistributedLoadTest
Iteration #2
Run Load Test
Scale-In
Wait for ASG to scale-In
Check ECS Capacity after scale-In
Scale-In done?
Succeed
Scale-Cleanup
Wait for ASG to scale-Cleanup
Check ECS Capacity after scale-Cleanup
Scale-Cleanup done?
Fail
End
In progress
Failed
Caught error
Canceled
Succeeded


AWS Step Functions Workflow
k6
Event
SNS
Amazon Q Developer
T

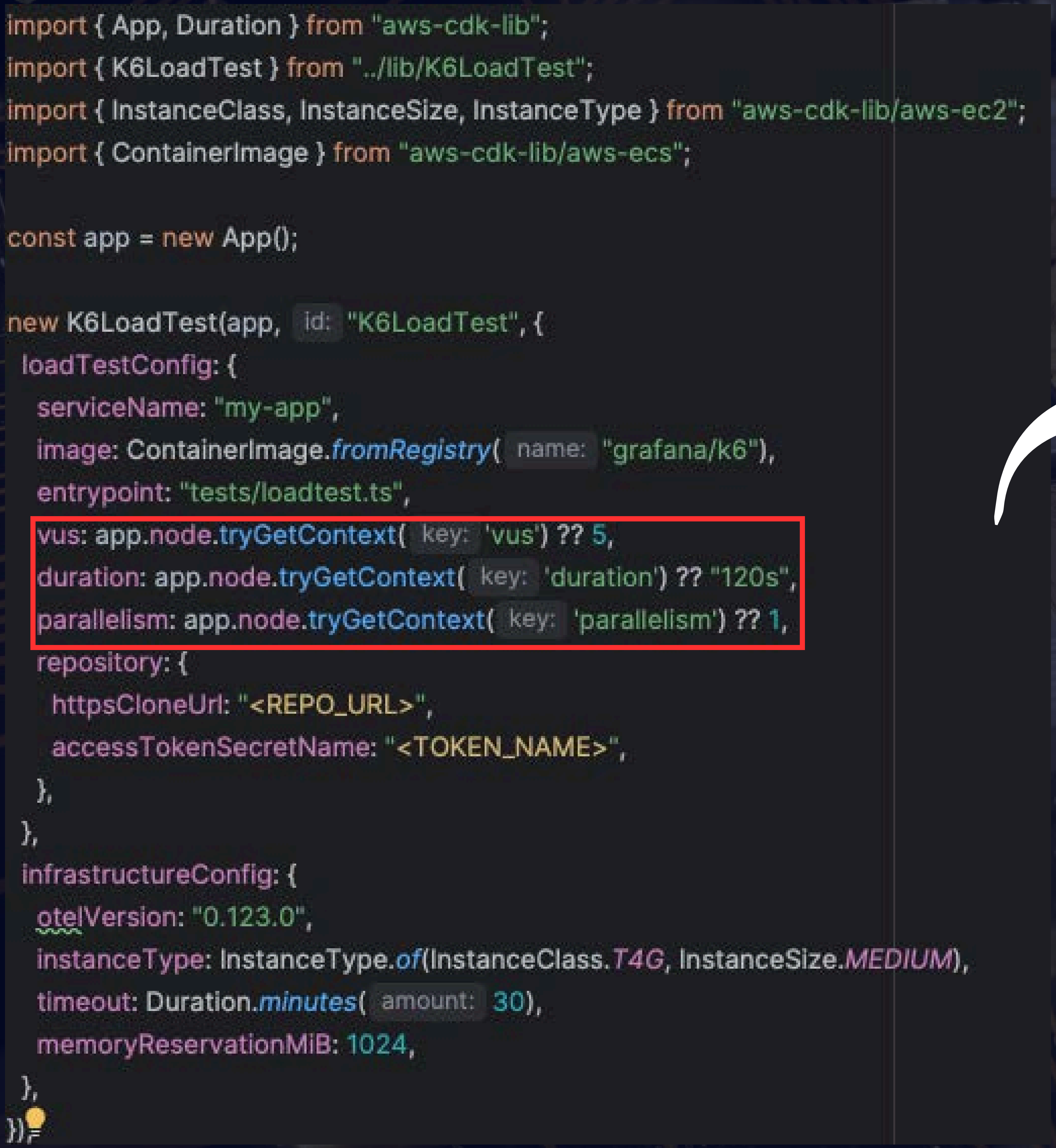
Convenience

:

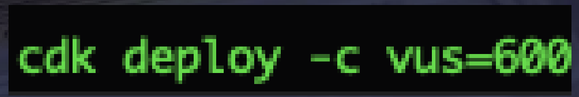
zero config/ zero shot deployment



```
import { App, Duration } from "aws-cdk-lib";
import { K6LoadTest } from "../lib/K6LoadTest";
import { InstanceClass, InstanceSize, InstanceType } from "aws-cdk-lib/aws-ec2";
import { ContainerImage } from "aws-cdk-lib/aws-ecs";

const app = new App();

new K6LoadTest(app, id: "K6LoadTest", {
  loadTestConfig: {
    serviceName: "my-app",
    image: ContainerImage.fromRegistry( name: "grafana/k6"),
    entrypoint: "tests/loadtest.ts",
    vus: app.node.tryGetContext( key: 'vus') ?? 5,
    duration: app.node.tryGetContext( key: 'duration') ?? "120s",
    parallelism: app.node.tryGetContext( key: 'parallelism') ?? 1,
    repository: {
      httpsCloneUrl: "<REPO_URL>",
      accessTokenSecretName: "<TOKEN_NAME>",
    },
  },
  infrastructureConfig: {
    otelVersion: "0.123.0",
    instanceType: InstanceType.of(InstanceClass.T4G, InstanceSize.MEDIUM),
    timeout: Duration.minutes( amount: 30),
    memoryReservationMiB: 1024,
  },
});
```



```
import { App, Duration } from "aws-cdk-lib";
import { K6LoadTest } from "../lib/K6LoadTest";
import { InstanceClass, InstanceSize, InstanceType } from "aws-cdk-lib/aws-ec2";
import { ContainerImage } from "aws-cdk-lib/aws-ecs";

const app = new App();

new K6LoadTest(app, id: "K6LoadTest", {
 loadTestConfig: {
  serviceName: "my-app",
  image: ContainerImage.fromRegistry( name: "grafana/k6"),
  entrypoint: "tests/loadtest.ts",
  vus: app.node.tryGetContext( key: 'vus') ?? 5,
  duration: app.node.tryGetContext( key: 'duration') ?? "120s",
  parallelism: app.node.tryGetContext( key: 'parallelism') ?? 1,
  repository: {
   httpsCloneUrl: "<REPO_URL>",
   accessTokenSecretName: "<TOKEN_NAME>",
  },
 },
 infrastructureConfig: {
  otelVersion: "0.123.0",
  instanceType: InstanceType.of(InstanceClass.T4G, InstanceSize.MEDIUM),
  timeout: Duration.minutes( amount: 30),
  memoryReservationMiB: 1024,
 },
});
```



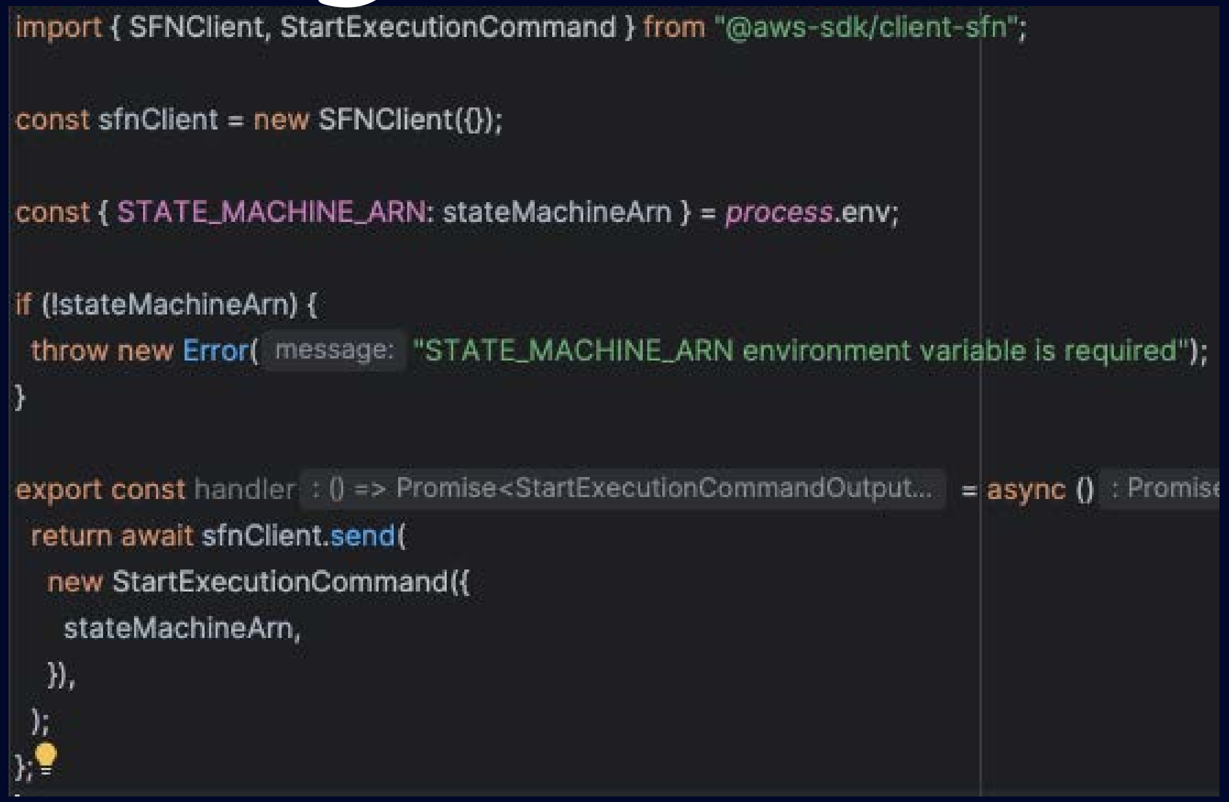
```
cdk deploy -c vus=600
```

How to run test on deploy?

1. Deploy
2. Run

How to run test on deploy?

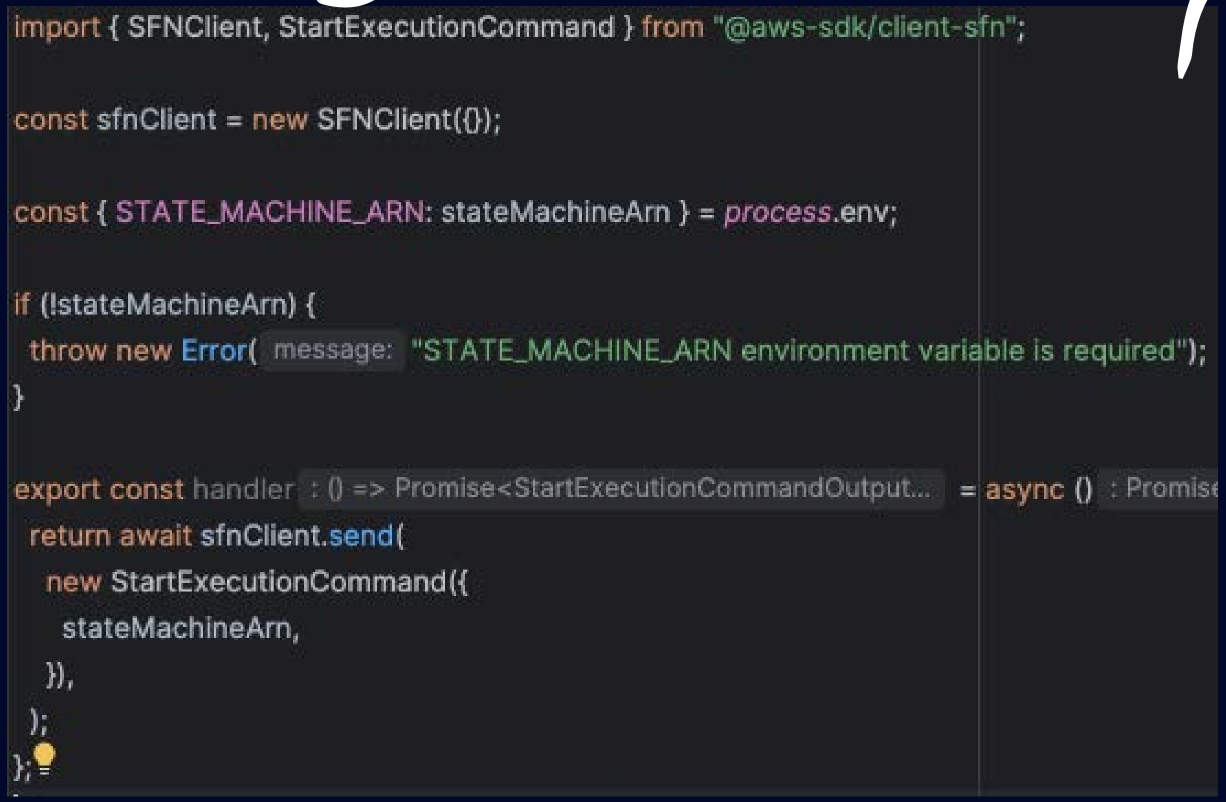
1. Deploy
2. Run



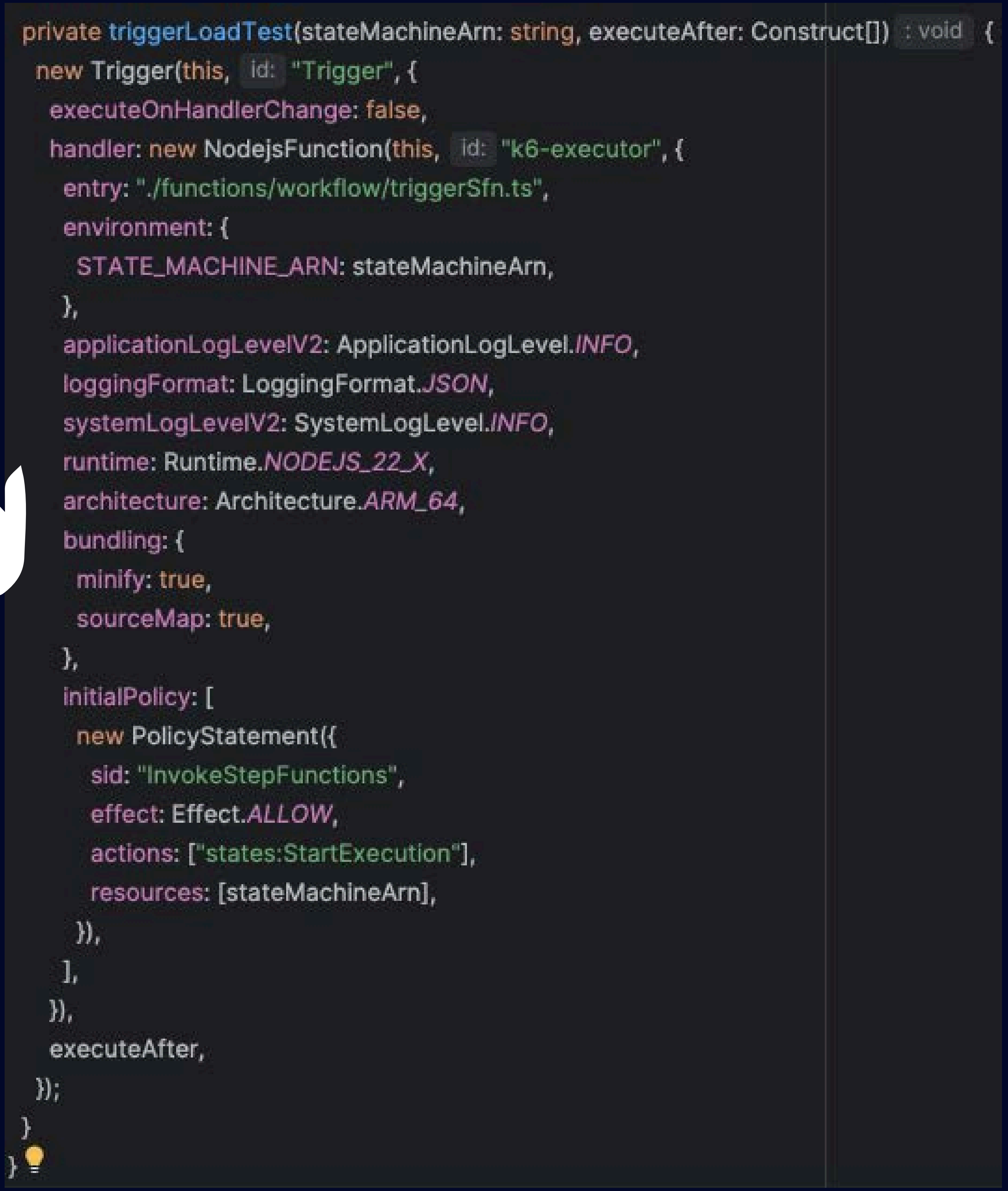
```
import { SFNClient, StartExecutionCommand } from "@aws-sdk/client-sfn";

const sfnClient = new SFNClient({});

const { STATE_MACHINE_ARN: stateMachineArn } = process.env;

if (!stateMachineArn) {
  throw new Error( message: "STATE_MACHINE_ARN environment variable is required");
}

export const handler : () => Promise<StartExecutionCommandOutput...  = async () : Promise
  return await sfnClient.send(
    new StartExecutionCommand({
      stateMachineArn,
    }),
  );
};
```

How to run test on deploy?

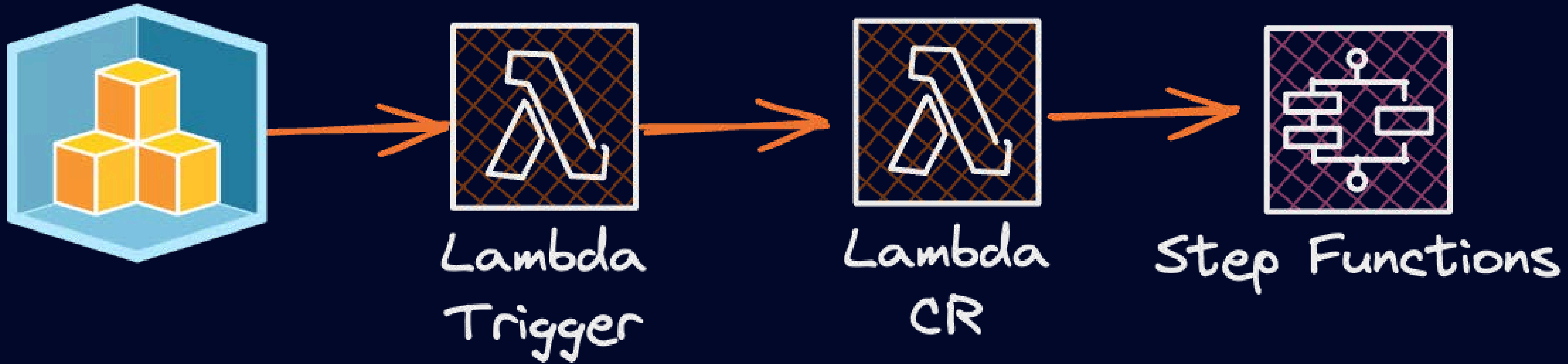
1. Deploy
2. Run



```
import { SFNClient, StartExecutionCommand } from "@aws-sdk/client-sfn";

const sfnClient = new SFNClient({});

const { STATE_MACHINE_ARN: stateMachineArn } = process.env;

if (!stateMachineArn) {
  throw new Error( message: "STATE_MACHINE_ARN environment variable is required");
}

export const handler : () => Promise<StartExecutionCommandOutput... = async () : Promise
  return await sfnClient.send(
    new StartExecutionCommand({
      stateMachineArn,
    }),
  );
};
```



```
private triggerLoadTest(stateMachineArn: string, executeAfter: Construct[]) : void {
  new Trigger(this, id: "Trigger", {
    executeOnHandlerChange: false,
    handler: new NodejsFunction(this, id: "k6-executor", {
      entry: "./functions/workflow/triggerSfn.ts",
      environment: {
        STATE_MACHINE_ARN: stateMachineArn,
      },
      applicationLogLevelV2: ApplicationLogLevel.INFO,
      loggingFormat: LoggingFormat.JSON,
      systemLogLevelV2: SystemLogLevel.INFO,
      runtime: Runtime.NODEJS_22_X,
      architecture: Architecture.ARM_64,
      bundling: {
        minify: true,
        sourceMap: true,
      },
      initialPolicy: [
        new PolicyStatement({
          sid: "InvokeStepFunctions",
          effect: Effect.ALLOW,
          actions: ["states:StartExecution"],
          resources: [stateMachineArn],
        }),
      ],
    }),
    executeAfter,
  });
}
}
```


Lambda
Trigger
Lambda
CR
Step Functions

Functions

Effort

:

3 days focus

quite some coffee

some headache

reservix
dein ticketportal

GitHub

FEEDBACK

THANK YOU

reservix
dein ticketportal