
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

BOA317-R

Building containerized apps: From development to production

Donnie Prakoso

Principal Developer Advocate, ASEAN
AWS

Efe Karakus

Senior Software Dev Engineer, Amazon ECS
AWS

Objectives

For you

- Gain practical knowledge on developing containerized apps
- Understand the dynamics of operating containerized apps
- Learn best practices

For us

- Provide you with a good experience in a hands-on workshop
- Get direct feedback to improve AWS Copilot and Amazon ECS
- Learn from your feedback on how to improve this workshop

In this workshop, we . . .

- Use Amazon ECS and AWS Fargate operated with AWS Copilot
- Simulate how we ship containerized applications
- Learn various use cases with demos
- Do hands-on modules

List of modules

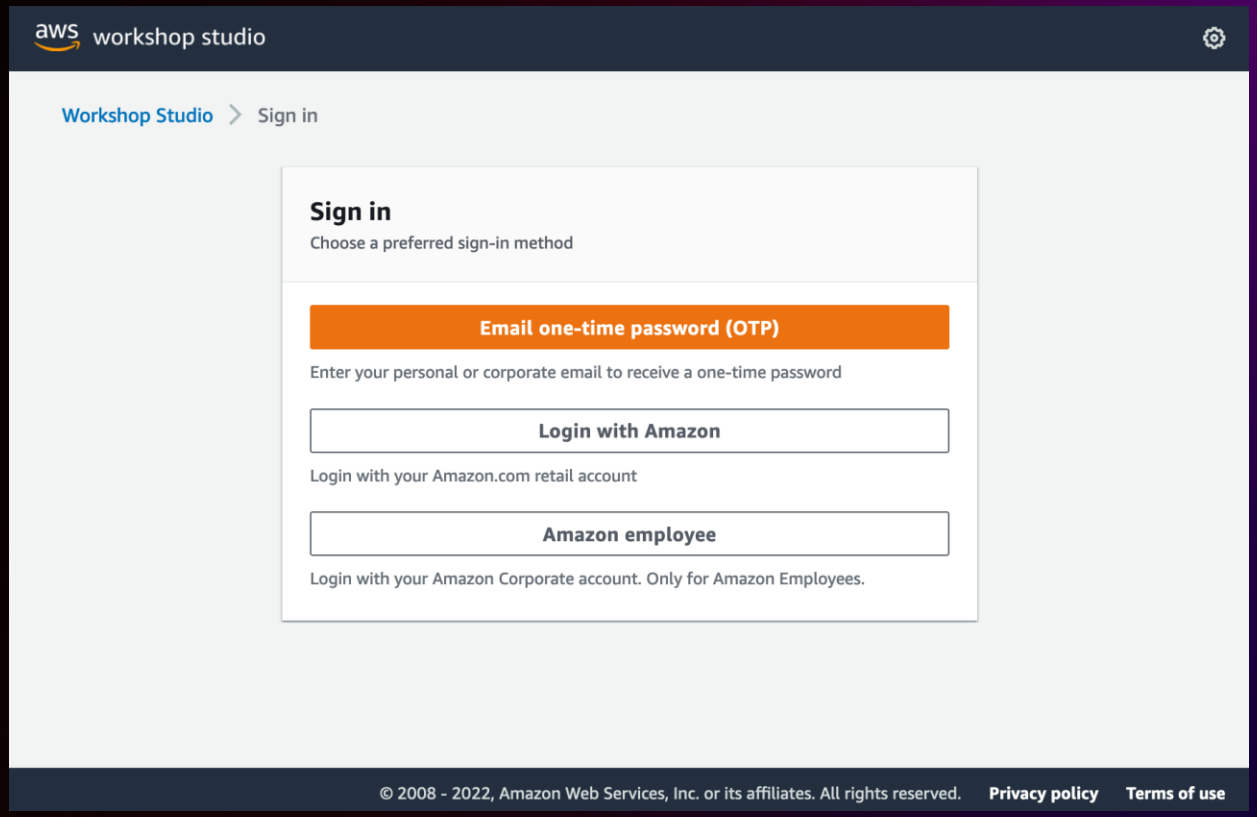
Hands-on modules	Extra modules
Module 1: Deploying API	Module 4: Implementing pub/sub
Module 2: Configuring multiple environments	Module 5: Building release pipeline
Module 3: Load-balanced private API	Module 6: Implementing observability and monitoring

Getting started with this workshop

- As a participant, you will have access to an AWS account with any optional pre-provisioned infrastructure and IAM policies needed to complete this workshop.
- The AWS account will only be available for the duration of this workshop. You will lose access to the account thereafter.
- The optional pre-provisioned infrastructure will be deployed to a specific region. Check your workshop content to determine whether other regions will be used.
- Be sure to review the terms and conditions of the event. Do not upload any personal or confidential information to the account.

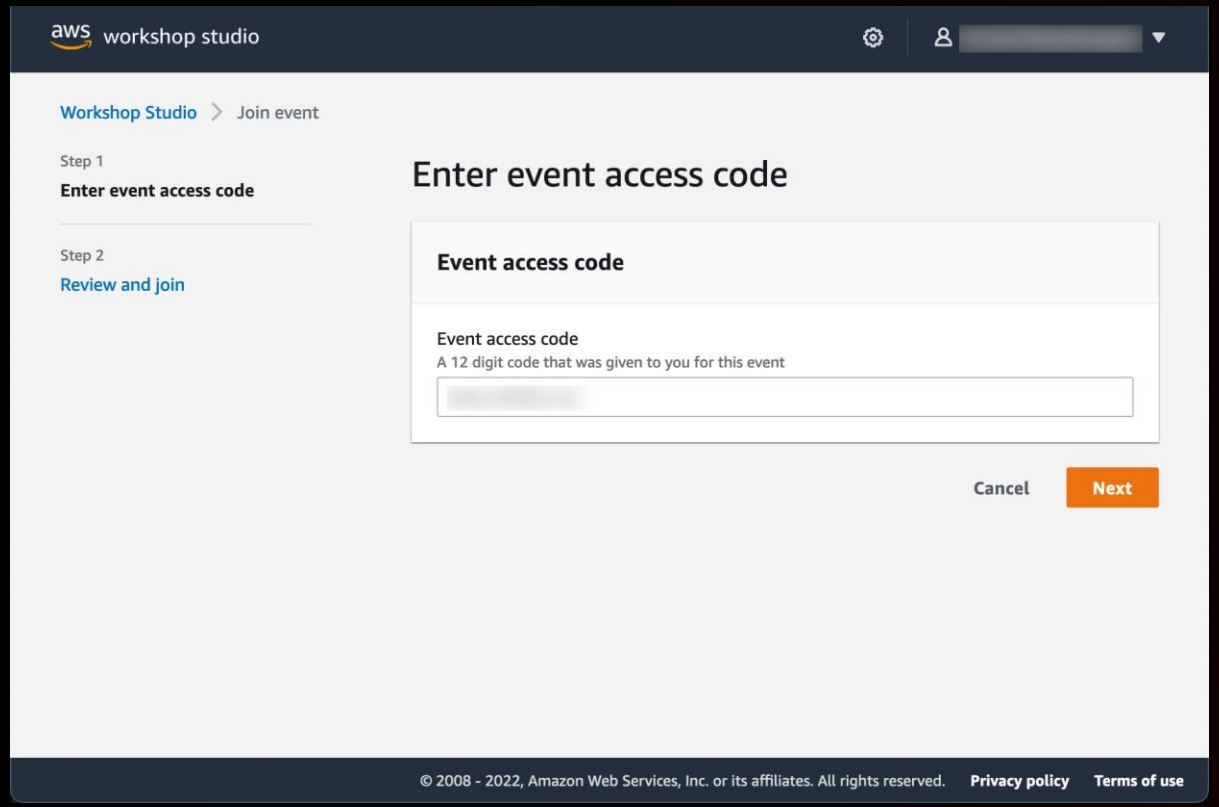
Step 1: Sign in via your preferred method

https://catalog.us-east-1.prod.workshops.aws/join?access-code=97d2-0744ab-db



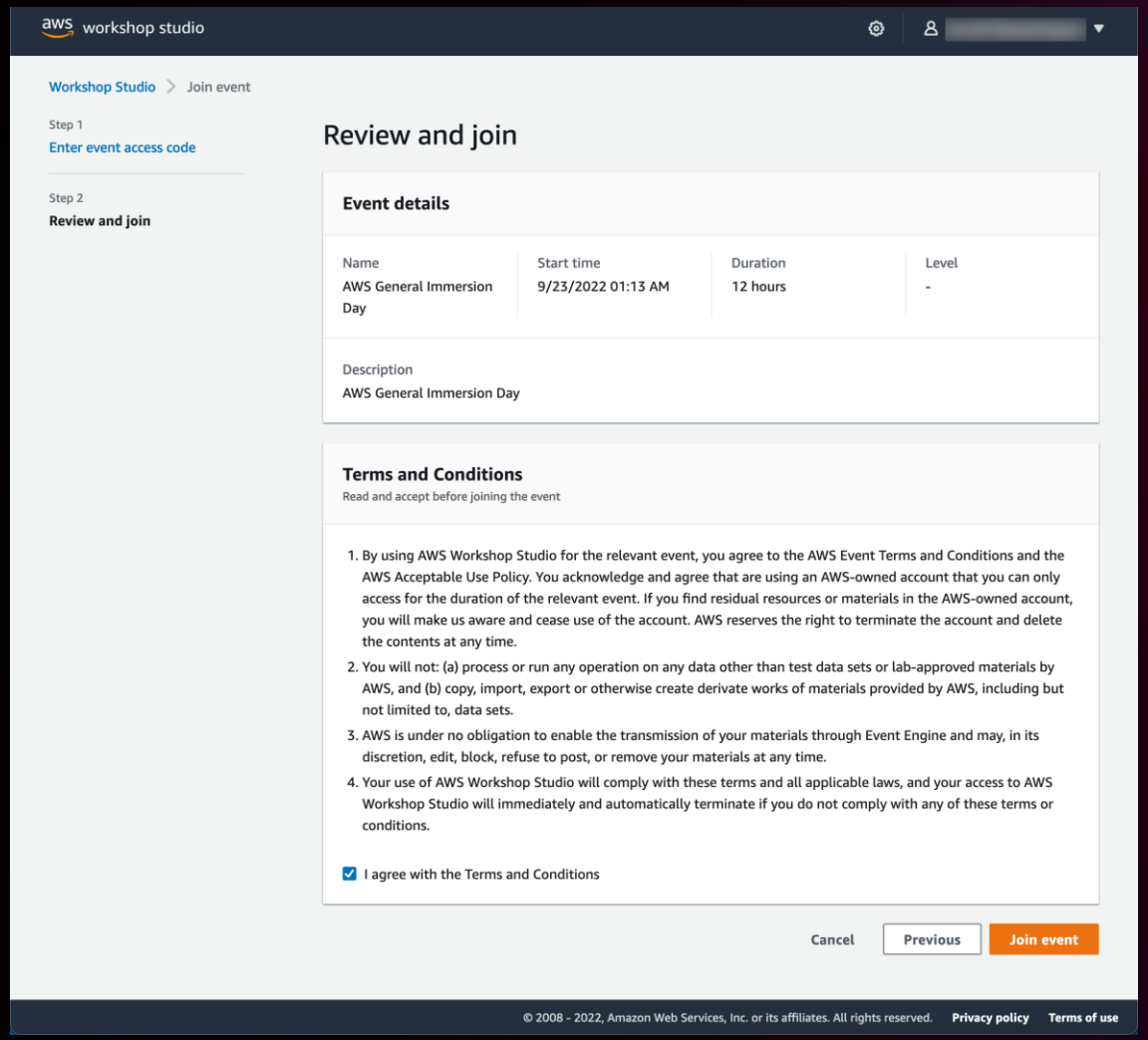
Step 2: Enter event access code

Enter the 12 digit event access code. If you were given a one-click join link, you can skip this step.



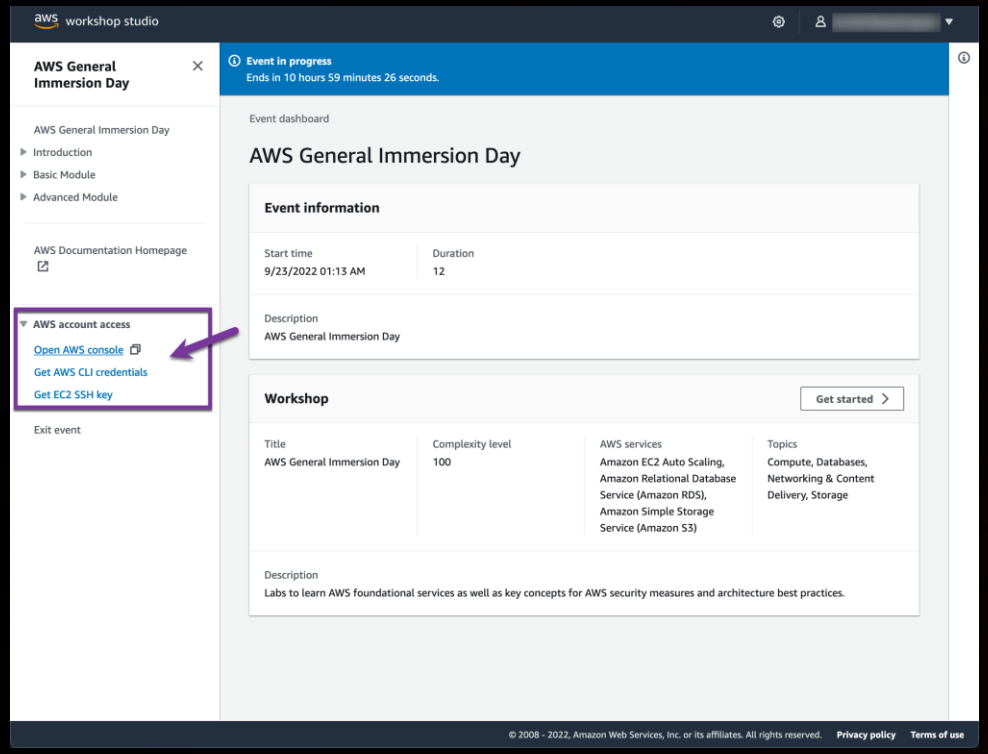
Access code: 97d2-0744ab-db

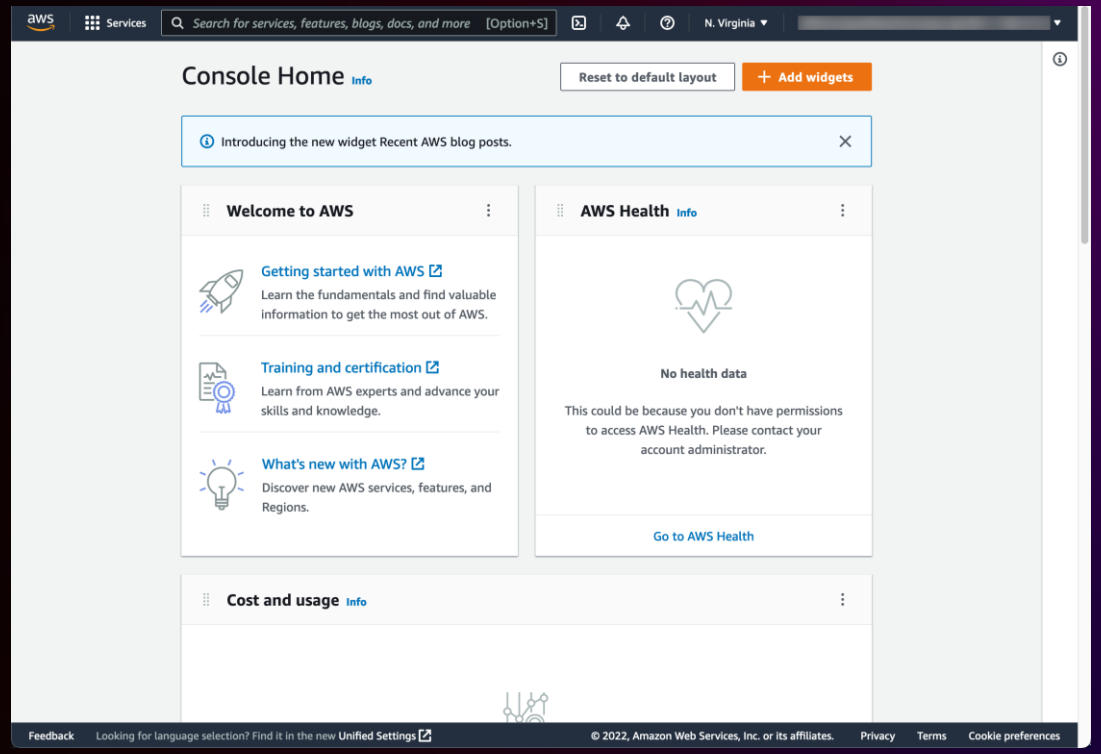
Step 3: Review terms and join event



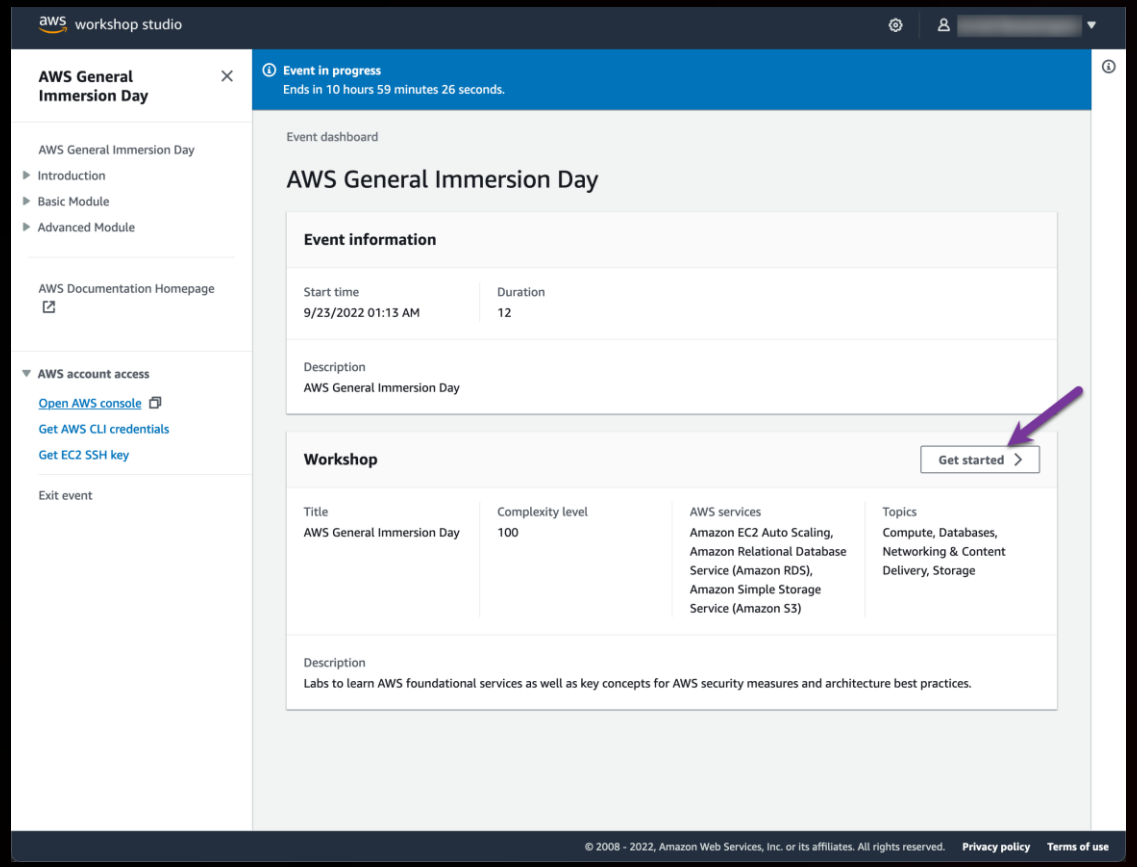
Step 4: Access AWS account

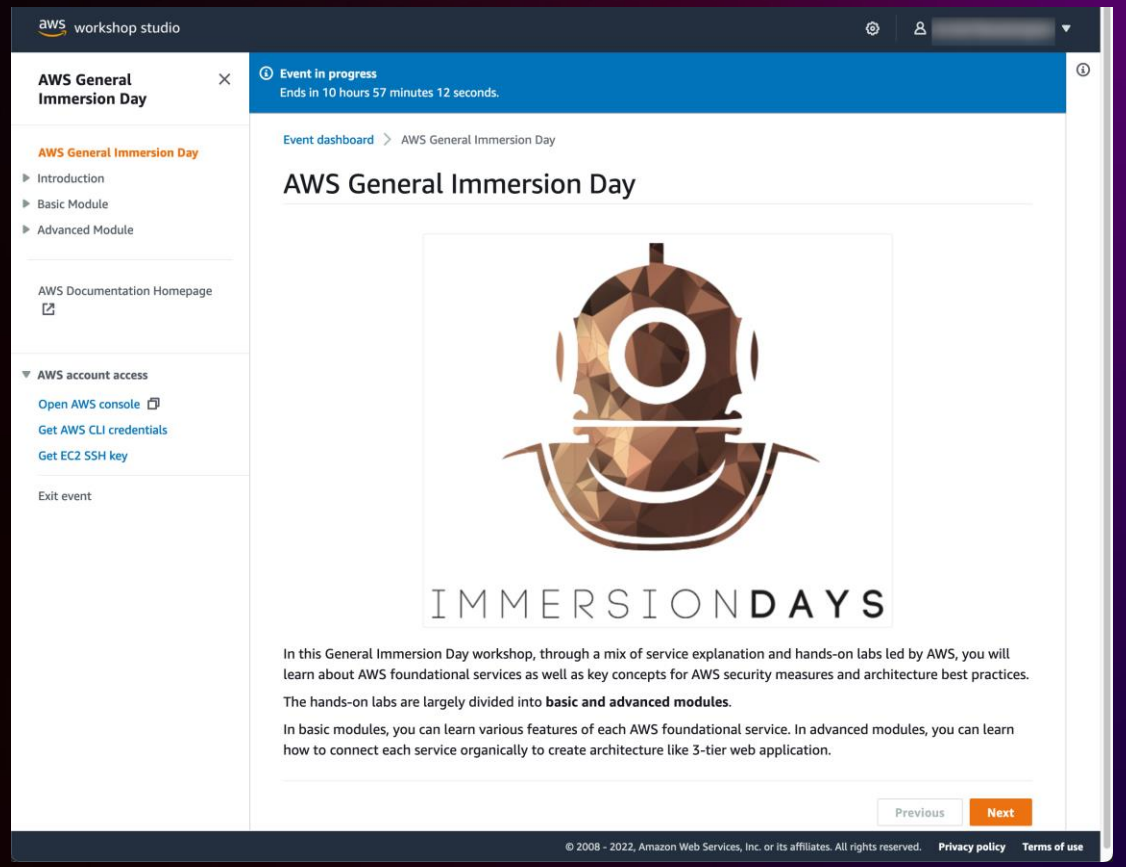
Access the AWS console, or generate AWS CLI credentials as needed.



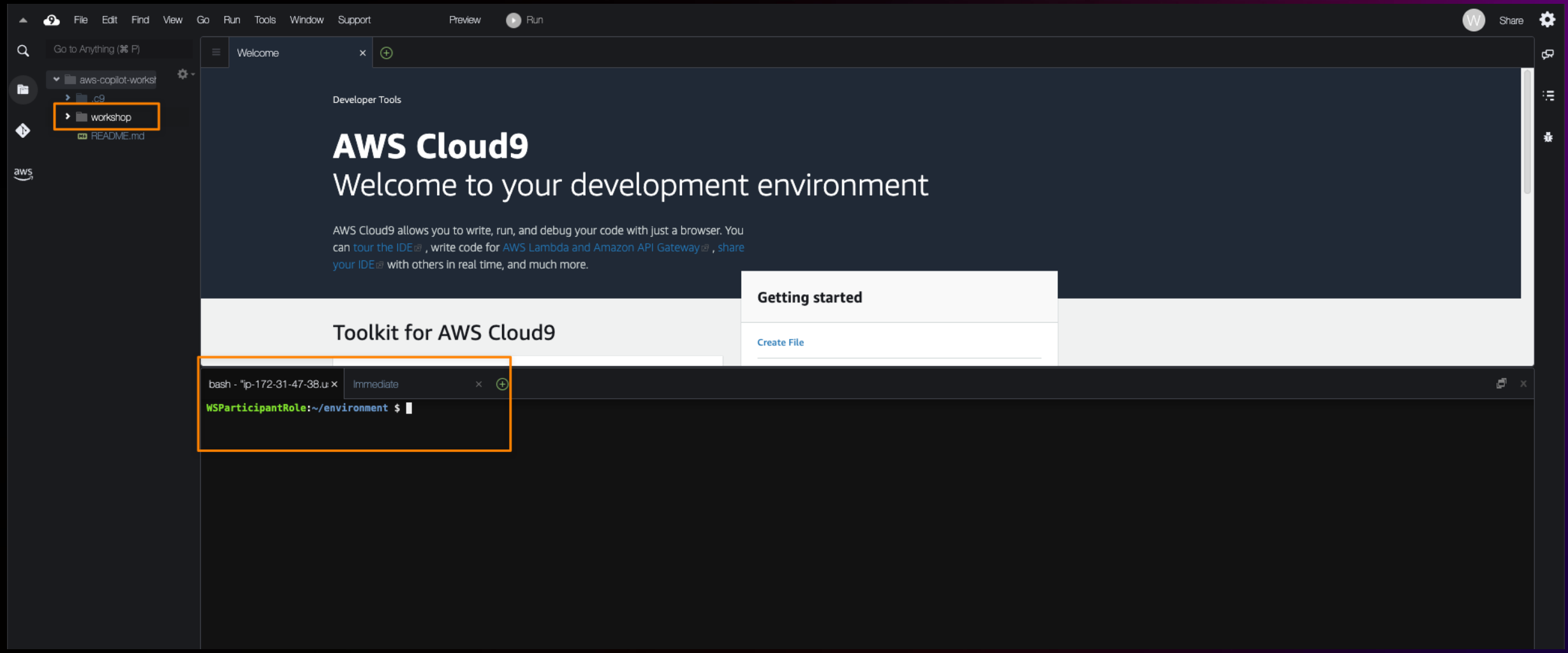


Step 5: Get started with the workshop





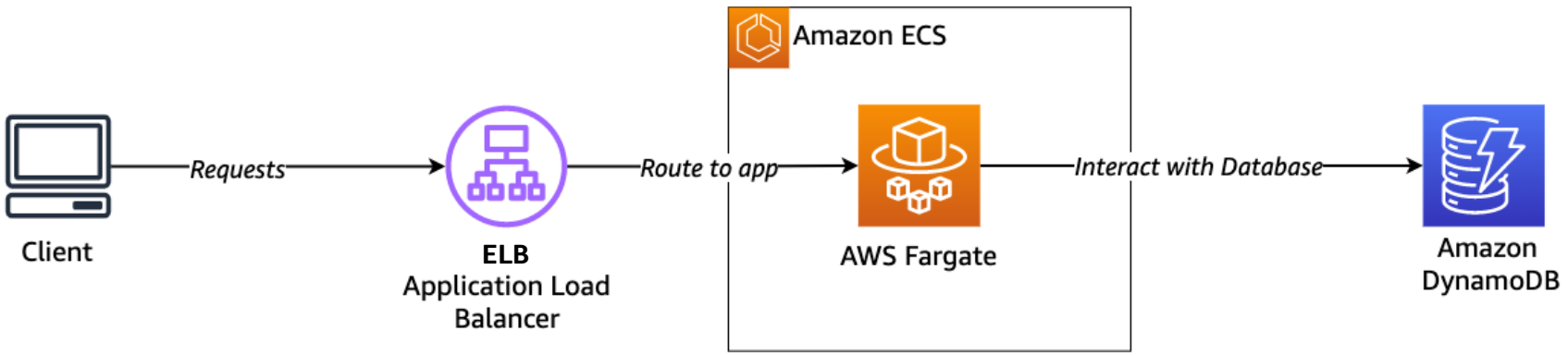
Step 6: Log in to AWS Cloud9 IDE



Workshop preparation (10 mins)

1. Go to **https://catalog.us-east-1.prod.workshops.aws/join?access-code=97d2-0744ab-db**
2. Input the event access code: **97d2-0744ab-db**
3. Get AWS CLI credentials
4. Finish **Get started with this workshop** -> **Running Workshop at an AWS Event**

Let’s start with a simple architecture



What we are going to build

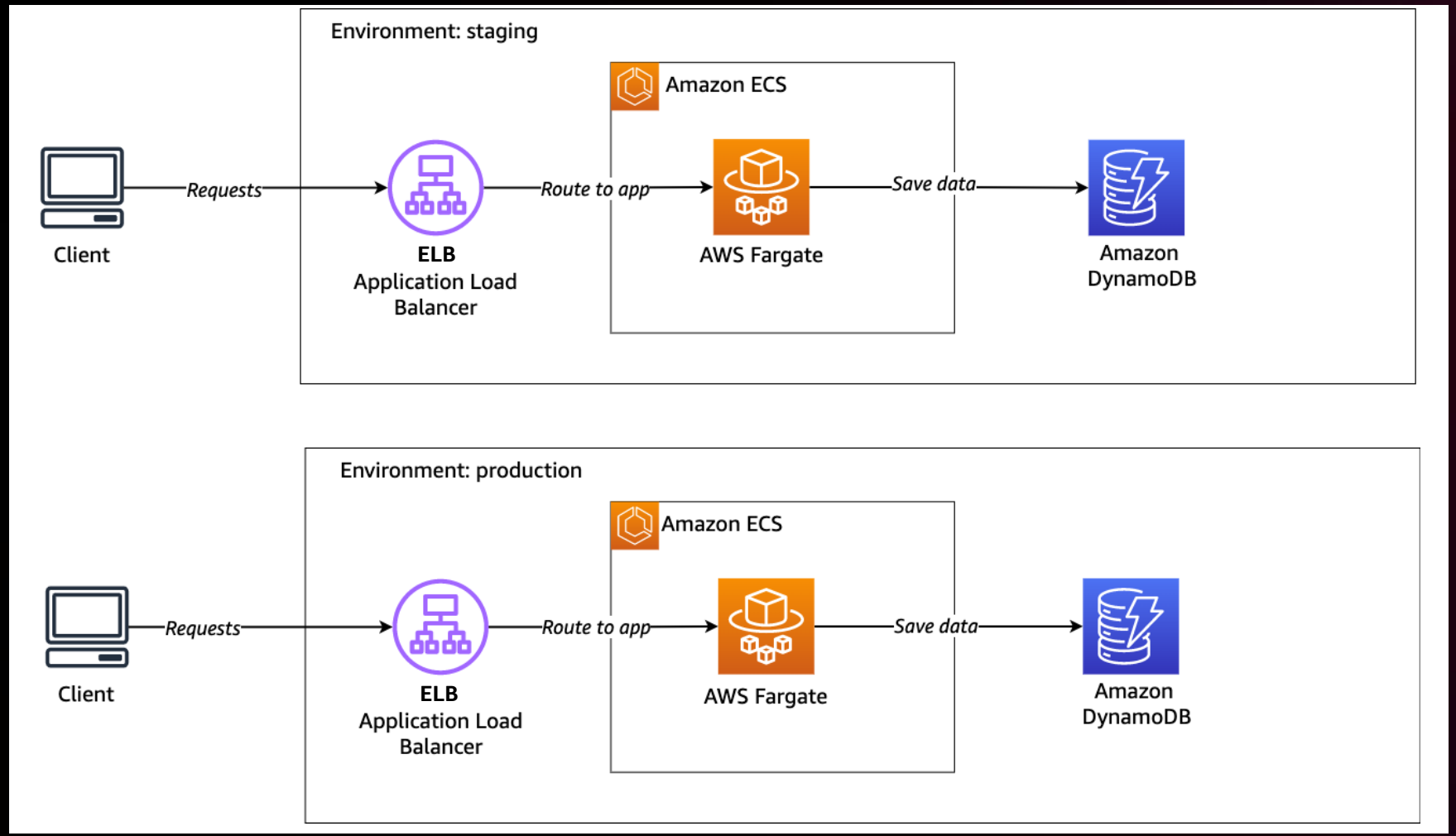
Basic architecture to interact with API and data retrieval and manipulation from/to database

Services and tools used

1. AWS Copilot
2. Elastic Load Balancing
3. Amazon ECS
4. AWS Fargate
5. Amazon DynamoDB

Start Module 1: Deploying API

But . . . I need to deploy into multiple environments



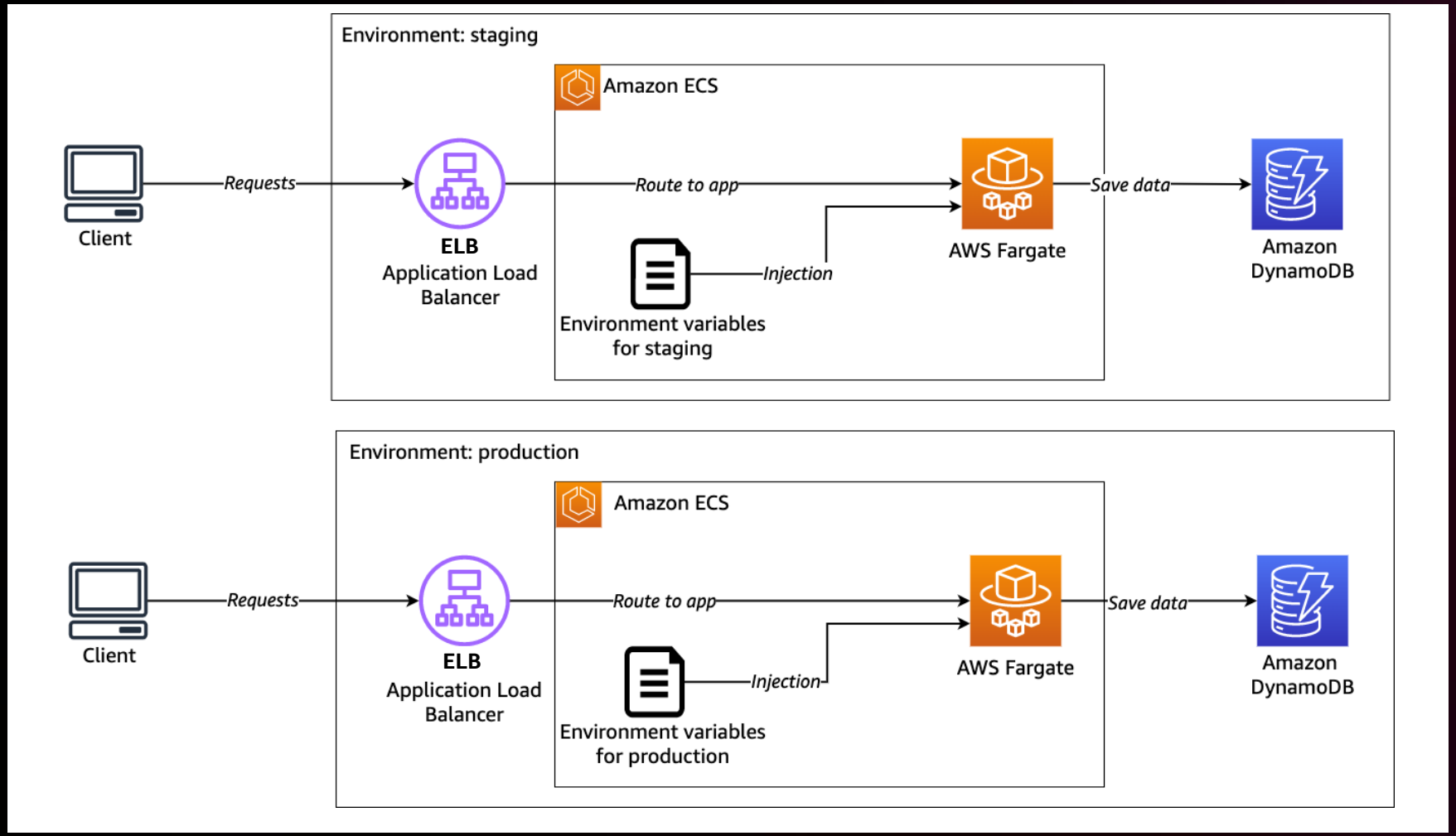
What we are going to build

Multiple environments (staging and production) and maintaining dev/prod parity for applications

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. Amazon DynamoDB

Adding environment variables



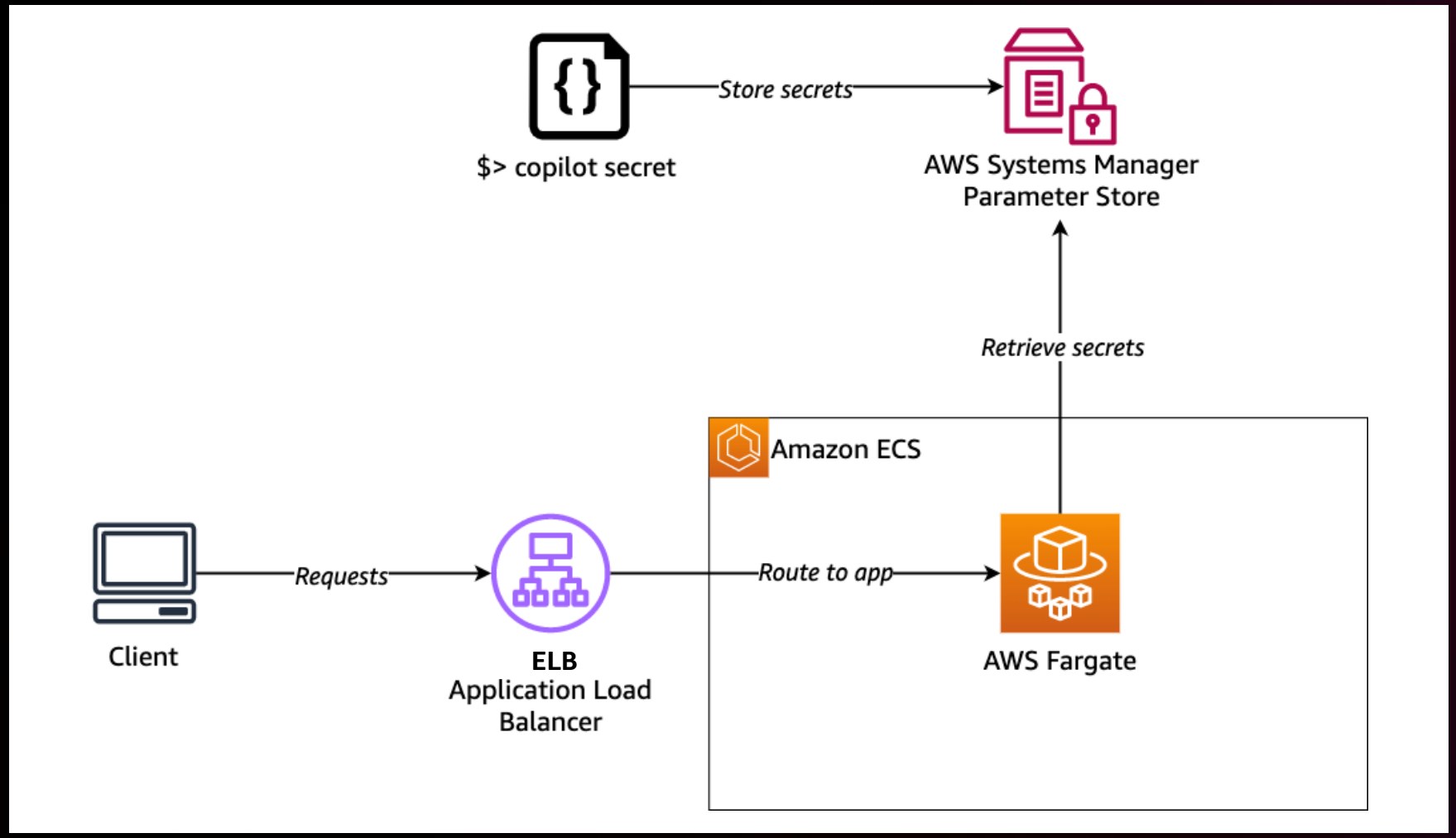
What we are going to build

Add global and separate environment variable(s) for each environment

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. Amazon DynamoDB

Configuring secrets



What we are going to build

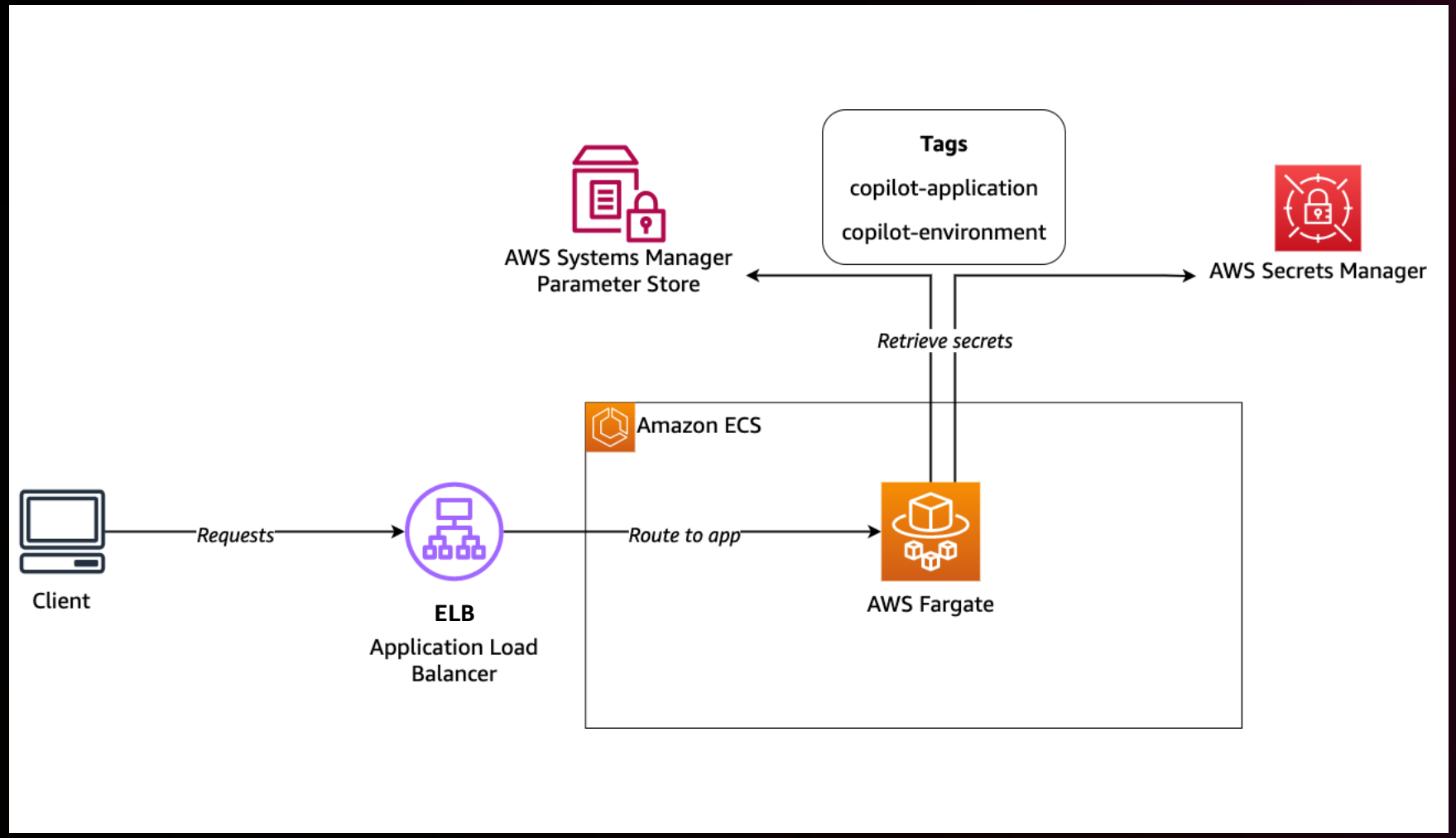
Add secret(s) with AWS Copilot and retrieve secret(s) from Parameter Store, a capability of AWS Systems Manager

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. Parameter Store

Start Module 2: Configuring multiple environments

Importing existing secrets



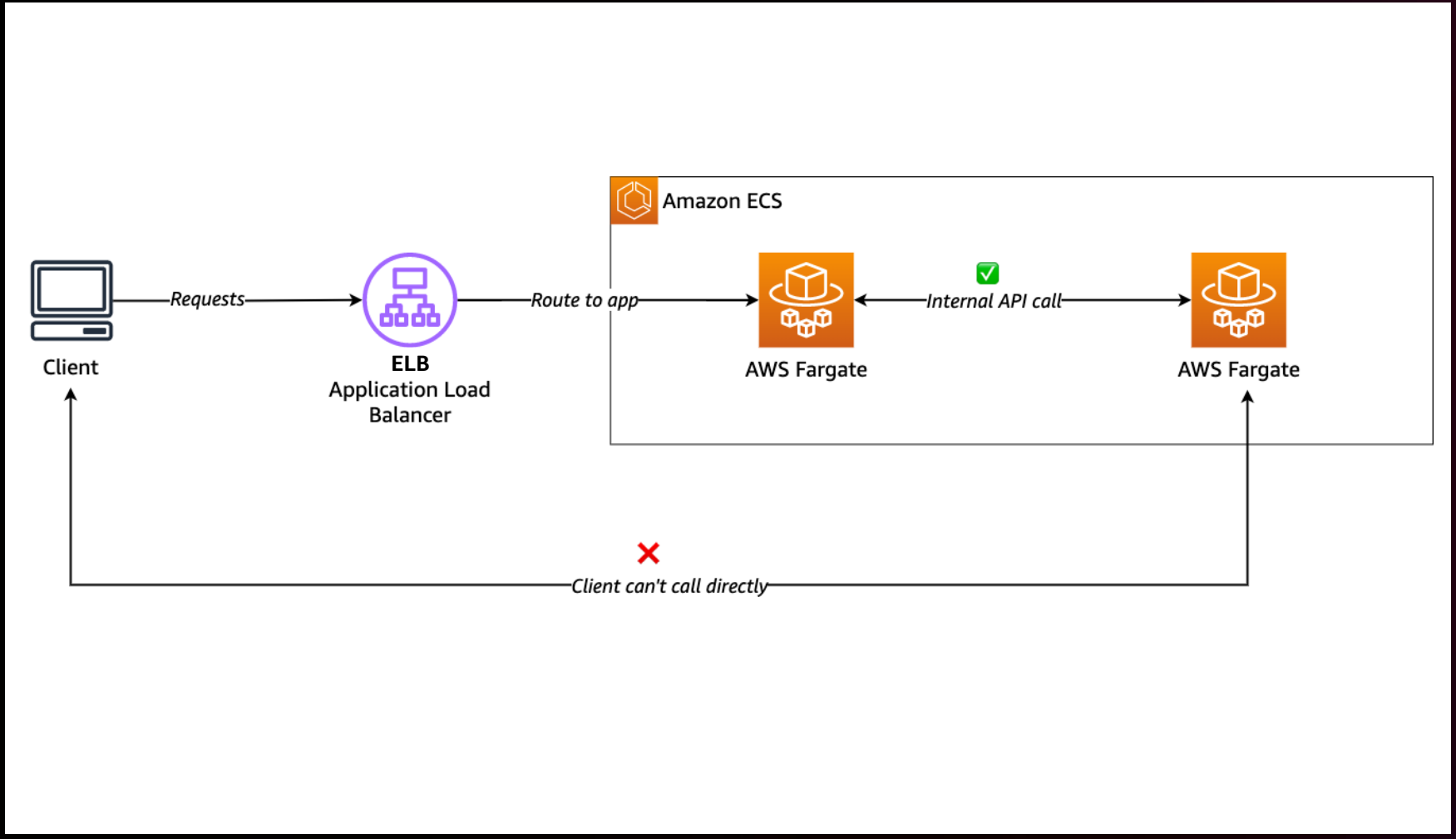
What we are going to build

Import existing secret(s) and retrieve secret(s) from application

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. Parameter Store
6. AWS Secrets Manager

Private API using Service Discovery



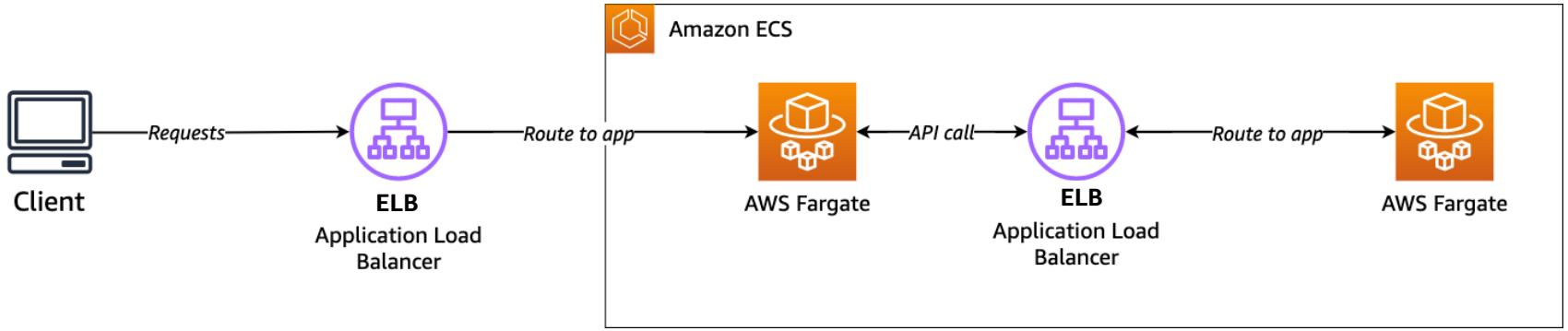
What we are going to build

Private API that isn't accessible from public and can only be accessed by internal application

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. Amazon ECS Service Discovery

Load-balanced private API



What we are going to build

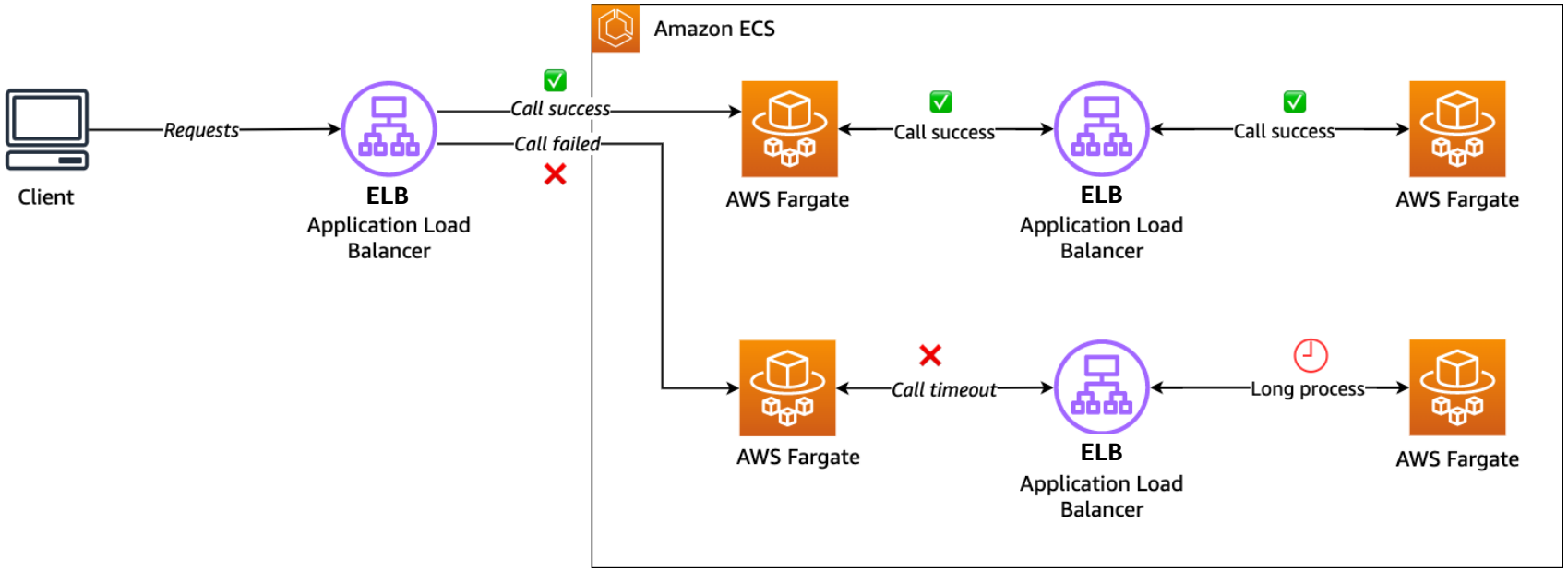
Add internal load balancer to access services to provide access from internal application for service communication

Services and tools used

1. AWS Copilot
2. ELB + internal ELB
3. Amazon ECS
4. AWS Fargate

Start Module 3:
Load-balanced private API

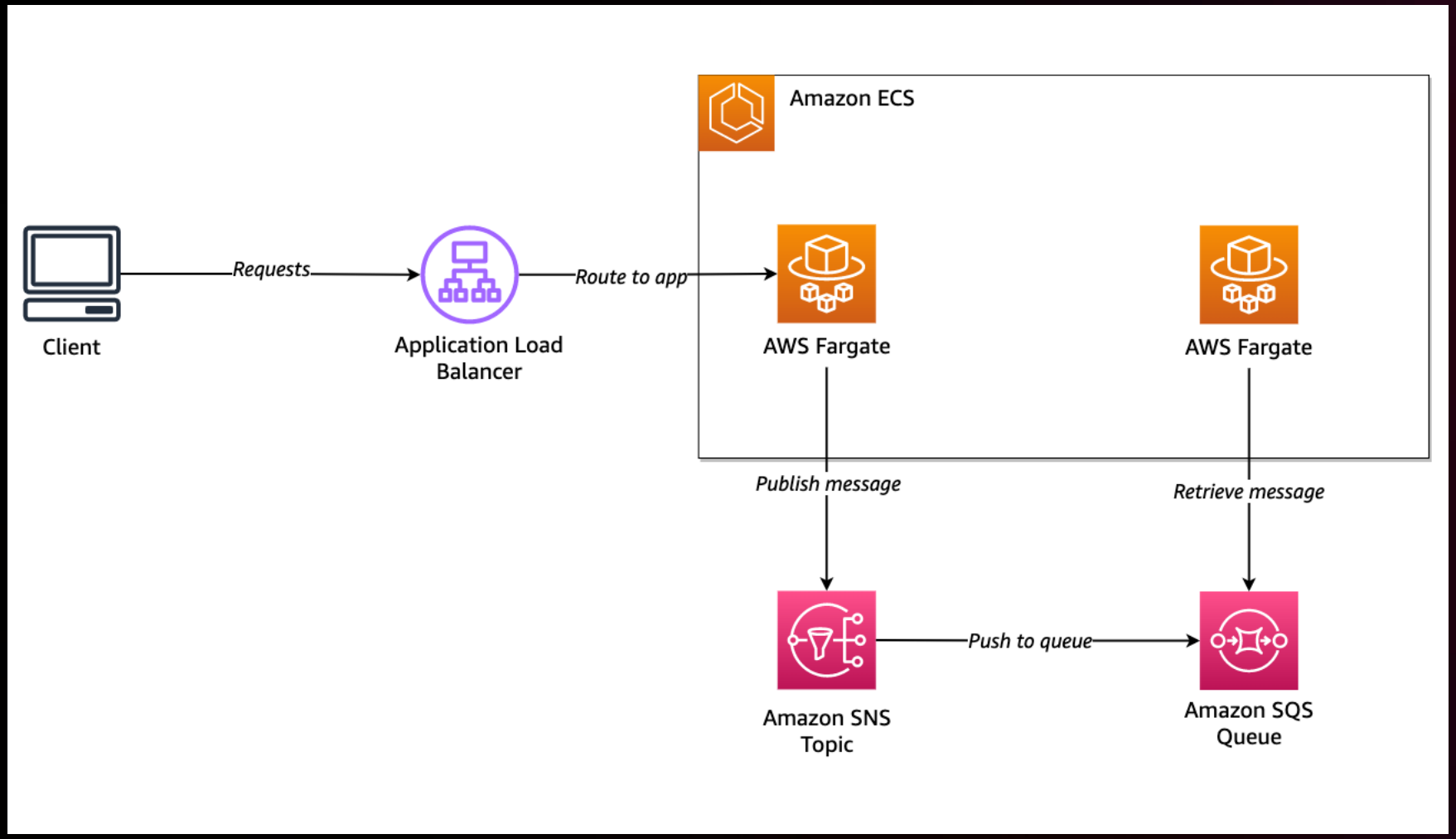
But . . . my application is very slow



Challenges

It's a common misconception that microservices consist **only** of a set of HTTP APIs. Due to the nature of synchronous call, it increases the interdependency within services and potentially introduces problems in application performance.

Implementing pub/sub



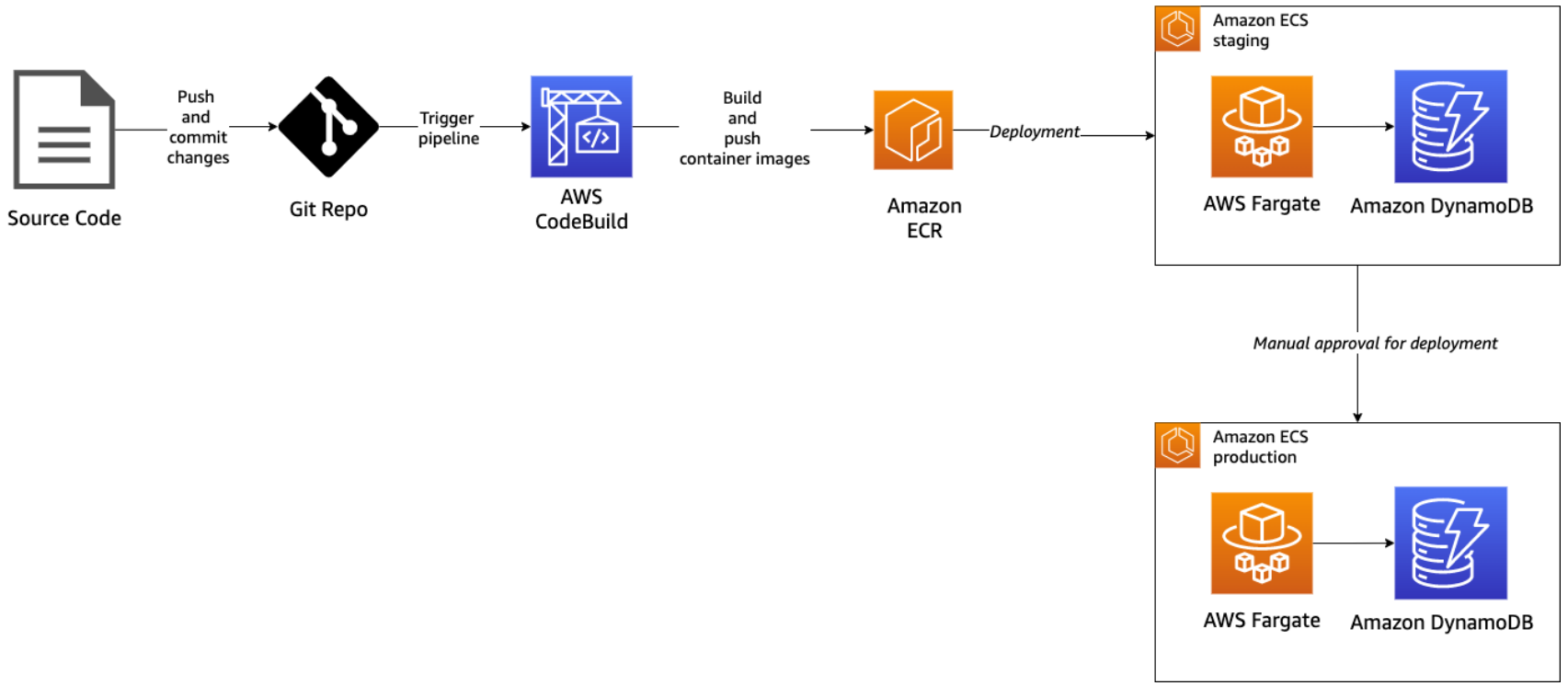
What we are going to build

Rearchitect internal services communication by leveraging asynchronous communication, implementing pub/sub pattern to decouple services

Services and tools used

1. AWS Copilot
2. Amazon ELB
3. Amazon ECS
4. AWS Fargate
5. Amazon SNS
6. Amazon SQS

Building release pipeline



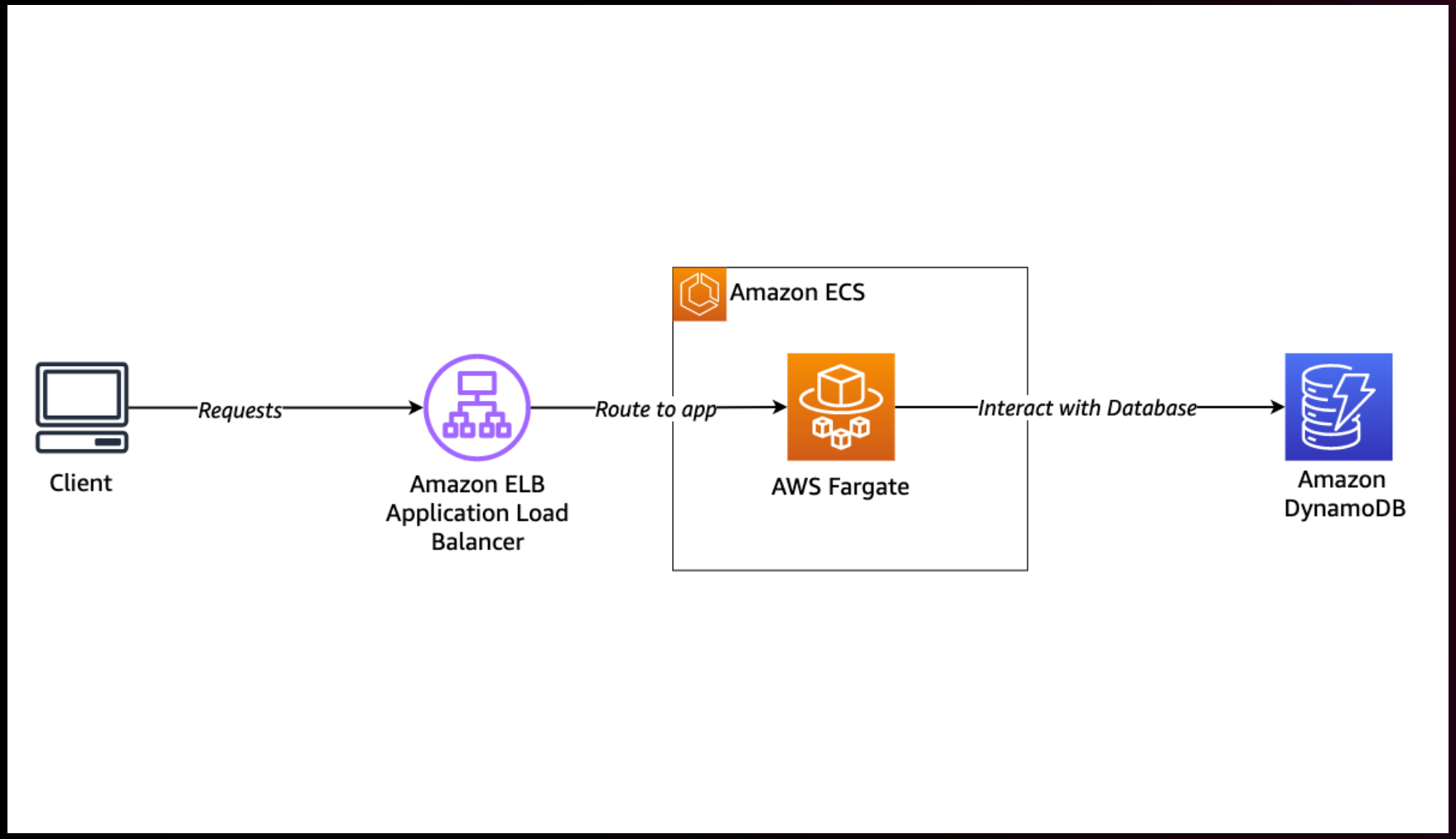
What we are going to build

Implement CI/CD for automatic deployment into application environment(s)

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. Amazon ECR
6. AWS CodePipeline
7. AWS CodeBuild

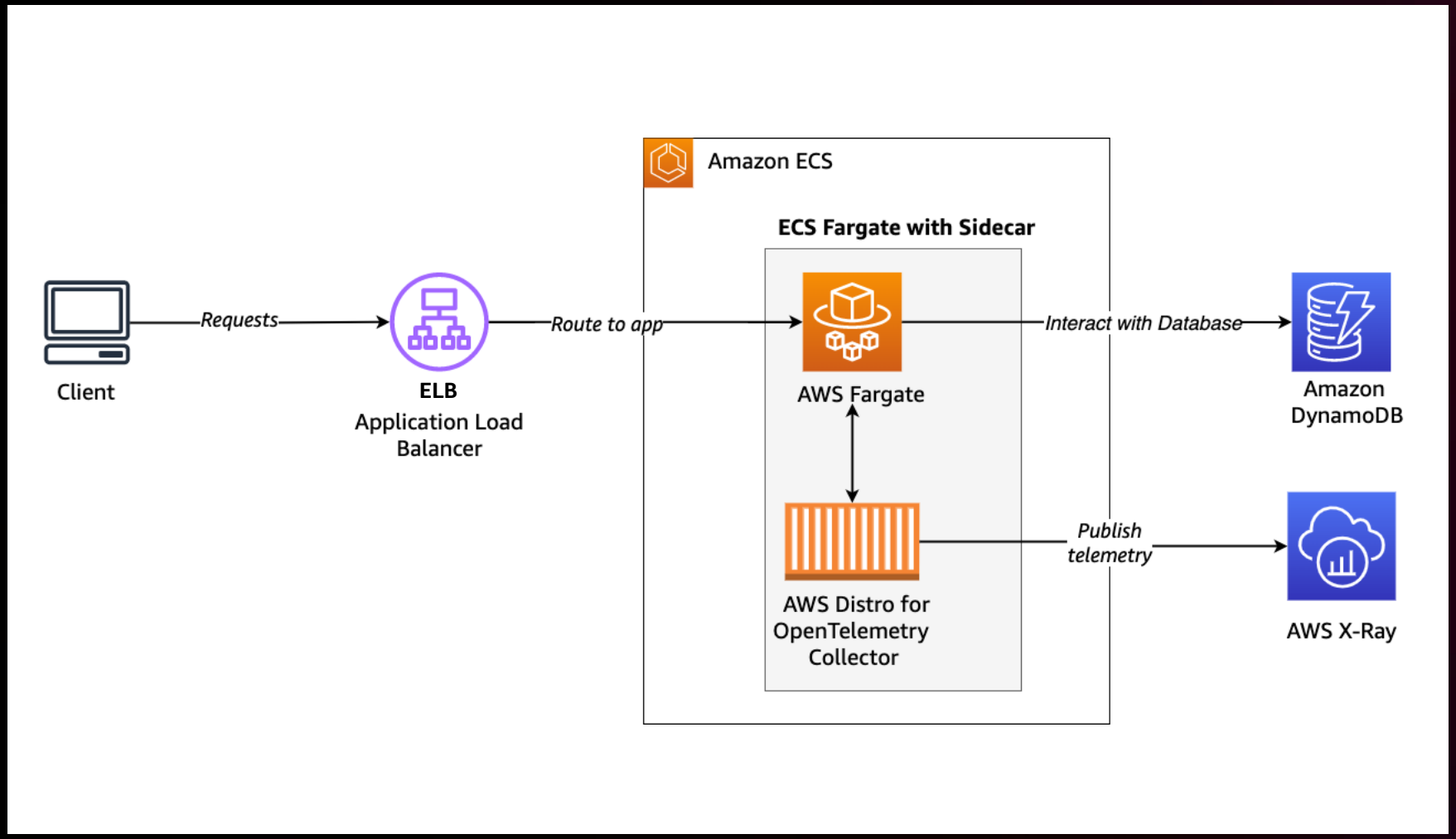
What's happening with my application?



Observability and monitoring

It's critical to implement observability and monitoring from an early stage to understand what's going on instead of guessing. Observability and monitoring is a combination of three foundational pillars: logs, metrics, and traces.

Implementing observability



What we are going to build

Use sidecar pattern to send telemetry data alongside the services

Services and tools used

1. AWS Copilot
2. ELB
3. Amazon ECS
4. AWS Fargate
5. AWS Lambda
6. AWS Distro for OpenTelemetry Collector
7. AWS X-Ray
8. Amazon CloudWatch

Start Module 6: Implementing observability and monitoring

Continue the workshop

https://catalog.workshops.aws/aws-copilot-workshop

Thank you!

Donnie Prakoso
@donnieprakoso
(Twitter | LinkedIn)

Efe Karakus
@efekarakus (Twitter)

go.donnie.id/youtube

Please complete the session survey in the **mobile app**