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

DAT205

Build your first graph application with Amazon Neptune

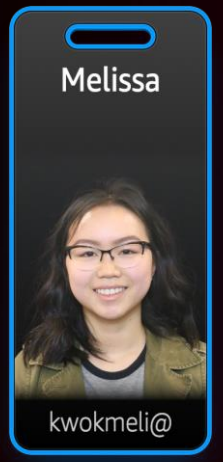
Kelvin Lawrence

Sr. Prin. Graph Architect
Amazon Neptune, AWS

Taylor Riggan

Sr. Graph Architect
Amazon Neptune, AWS

Meet the team



Agenda

- What are we building?
- Graph fundamentals
- Let’s build!

What are we building?

Travel agency

You've just inherited a travel agency that helps vacationers decide on travel routes.

Vacation consultants for the travel agency provide customers with a set of routes that take them from one destination to another, sometimes including extra cities along the way and limiting routes based on total distances.

The challenge

Today, the consultants use a relational database to help them plan routes. So, queries that need to find different paths from Austin to Auckland, for example, look a little like this:

```
select a.code ,r.dist, a2.code, r2.dist,a3.code,
r2.dist+r.dist as total from airports a
join routes r on a.id = r."from"
join airports a2 on a2.id = r."to"
join routes r2 on a2.id = r2."from"
join airports a3 on a3.id = r2."to"
where a.code = 'AUS'
  and a3.code = 'AKL'
order by total asc
limit 4
```

routes table

id	from	to	dist
1010	3	8	190
1054	3	13	1230
1123	3	23	1500
1189	13	63	6512
1189	23	63	6525

airports table

id	label	code	city
3	airport	AUS	Austin
8	airport	DFW	Dallas
13	airport	LAX	Los Ang
23	airport	SFO	San Fr
63	airport	AKL	

Actually, this query doesn't give you variable length paths, only one-stop routes…

The challenge

This is the real SQL query using Common Table Expressions (CTE)

```
WITH RECURSIVE graphtraversal AS (
    SELECT e.src, e.dest, 1 as depth, concat_ws('->',n1.iata,n2.iata) as path
    FROM airports n1
    INNER JOIN routes e on n1.id = e.src  INNER JOIN airports n2 ON n2.id =
e.dest
    WHERE n1.iata = 'AUS'
  UNION
    SELECT r.src, p.dest, r.depth + 1 as depth, concat_ws('->', r.path,
n.iata)
    FROM graphtraversal r INNER JOIN routes p on p.src = r.dest

      INNER JOIN airports n ON p.dest = n.id
    WHERE r.depth < 4
)
SELECT r.depth as total_hops, r.dest as destination, r.path
FROM graphtraversal r INNER JOIN airports n ON r.dest = n.id
WHERE n.iata = 'AKL';
```

The objective

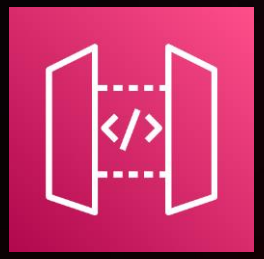
Your consultants are tired of writing and managing long, complex SQL queries.

You've just learned about graph databases and believe they will be a better fit for the highly connected air routes data your consultants work with and the types of queries that they write.

So you decide to create a web application that is backed by Amazon Neptune.

AWS services you will be using today

Amazon CloudFront



Amazon API Gateway

AWS Lambda

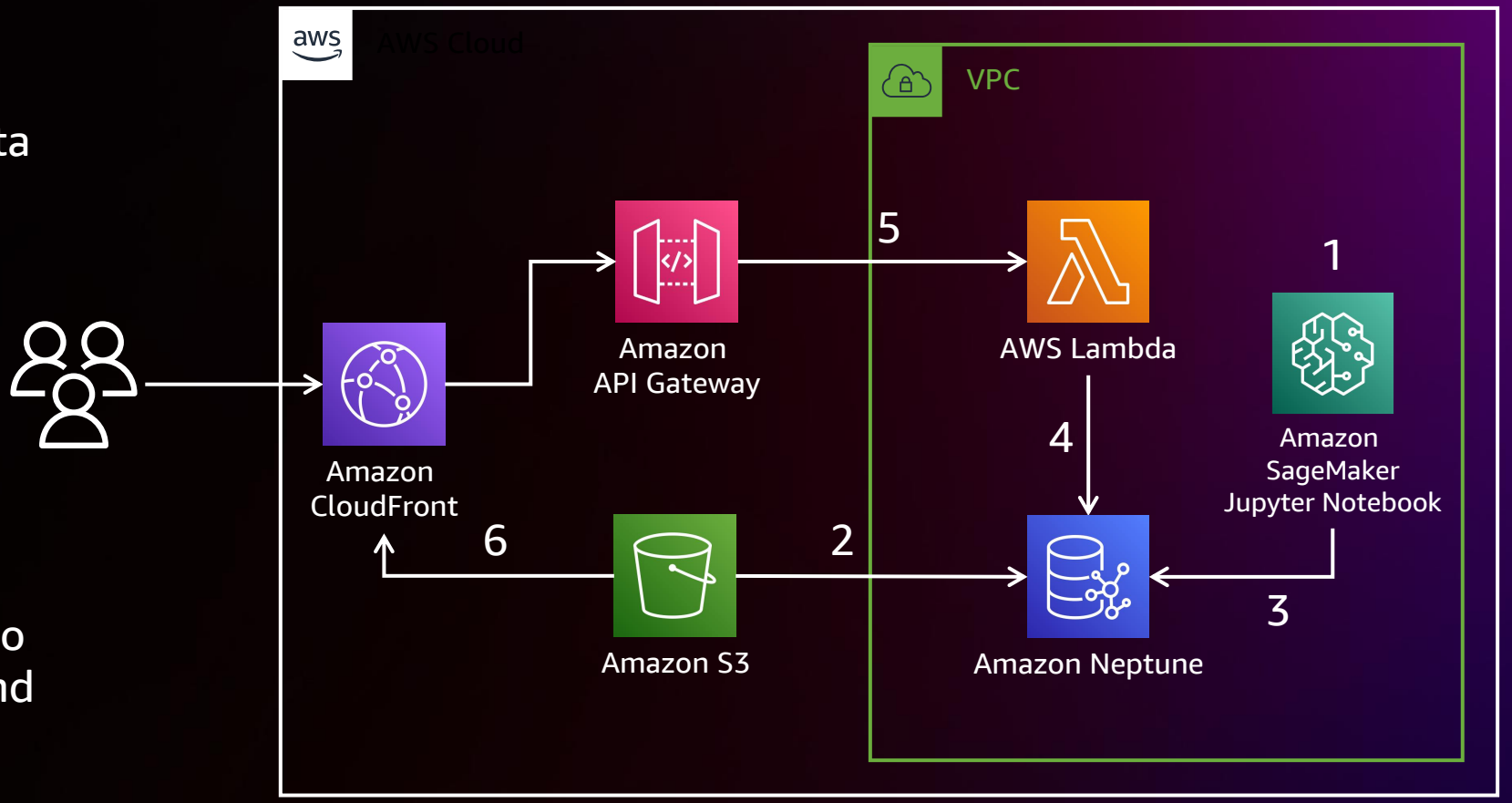
Amazon SageMaker

Amazon Neptune

Amazon S3

The architecture

1. Extract and transform the data
2. Bulk load the data
3. Design your queries
4. Configure AWS Lambda to submit queries
5. Configure Amazon API Gateway to add a “front door” to your back end from the front end
6. Layer on the front end



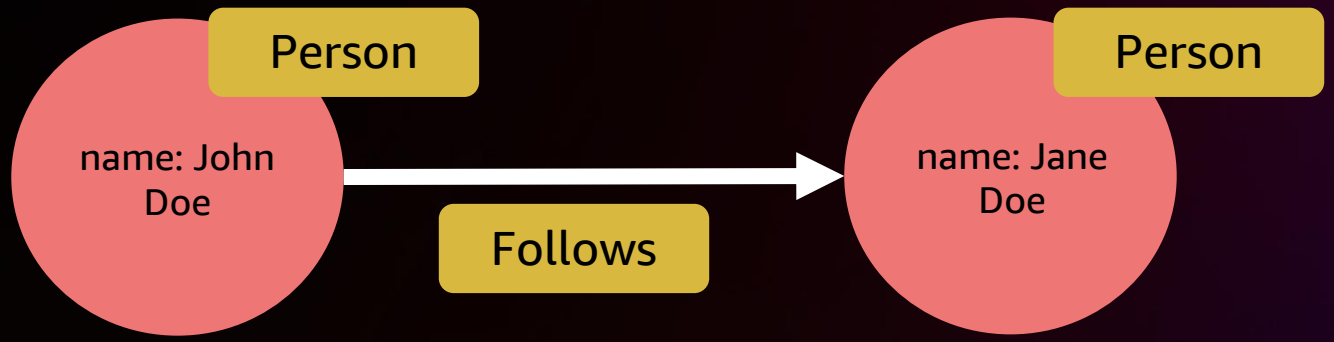
Graph fundamentals

What is a graph?

- Graphs are purpose-built to store and navigate relationships
- Nodes represent real-world objects
- Edges store relationships between objects

What is a property graph?

- Each node or edge is like a “container” that can hold properties and label(s)
- Labels are used for categorizing types of entities
- Properties are used to specify attributes of a particular instance of a node or edge type



Why use a graph?

Social networking

Recommendations

Knowledge graphs

Fraud detection

Life sciences

Network and IT operations

These use cases all:

- Have a highly connected data domain
- Navigate (variably) connected structures
- Filter or compute results based on the strength, weight, or quality of relationships

Let’s build!

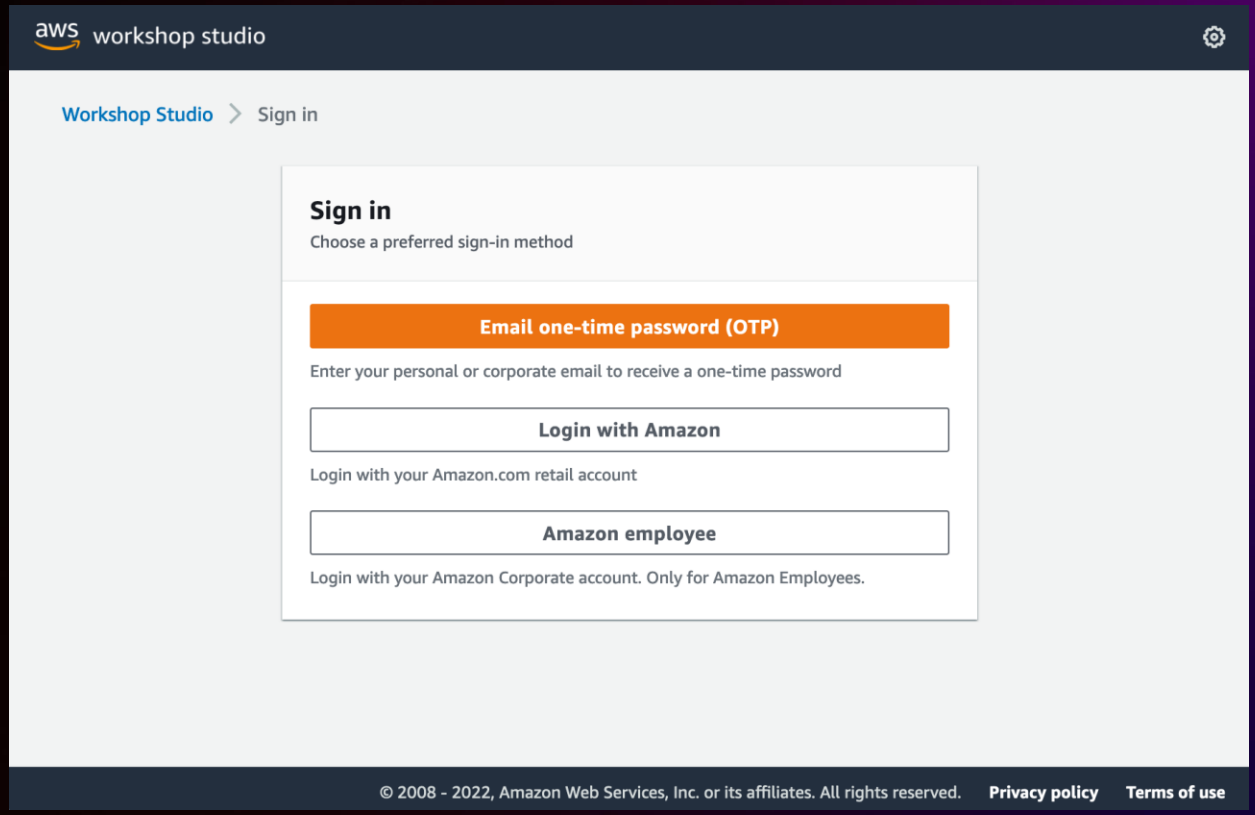
Getting started with this workshop

- As a participant, you will have access to an AWS account with 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 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 in the account.

Step 1: Sign in via your preferred method

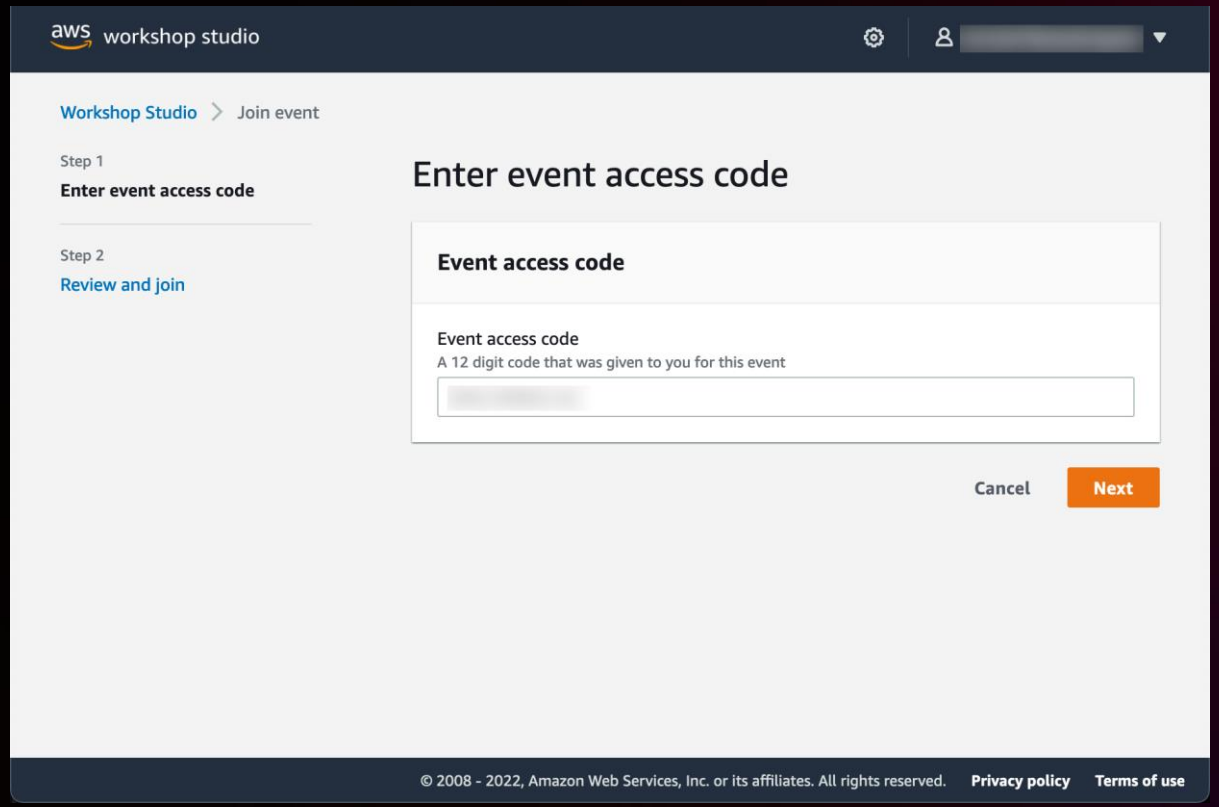
https://catalog.us-east-1.prod.workshops.aws/join

Or shortened: https://s12d.com/dat205

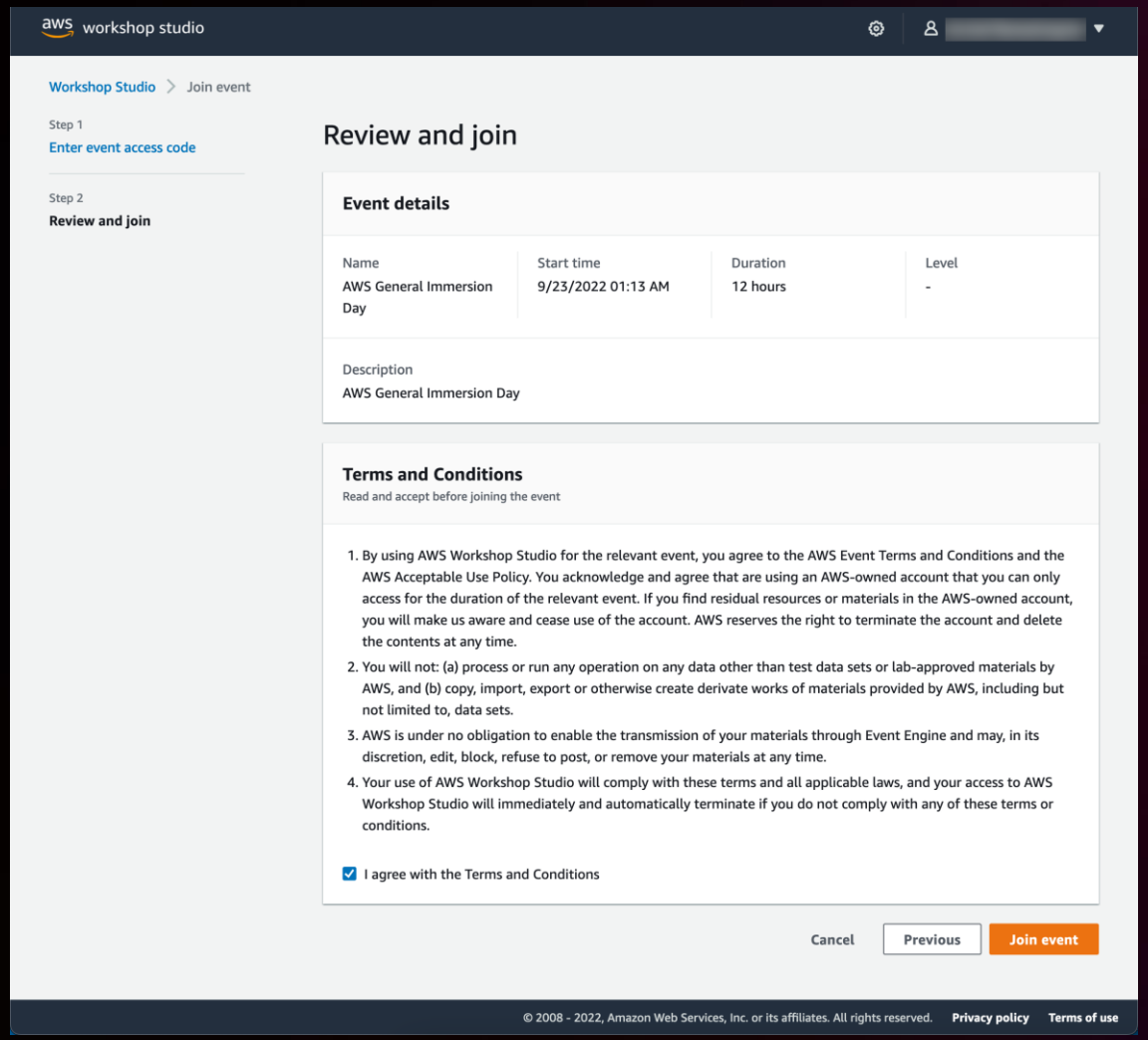


Step 2: Enter event access code

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

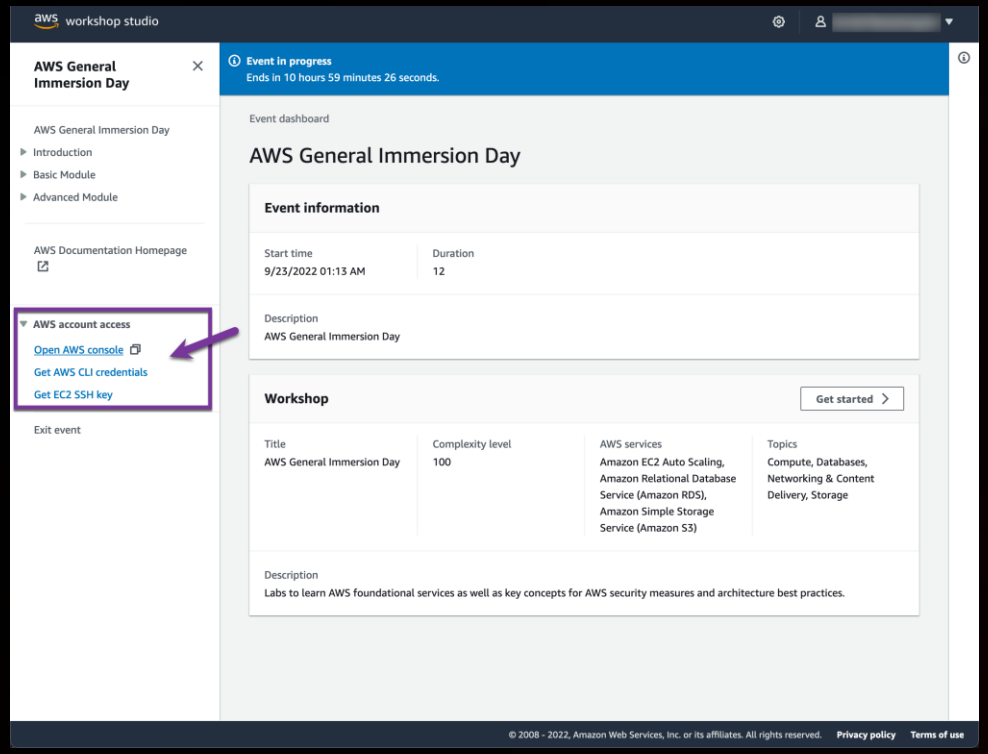


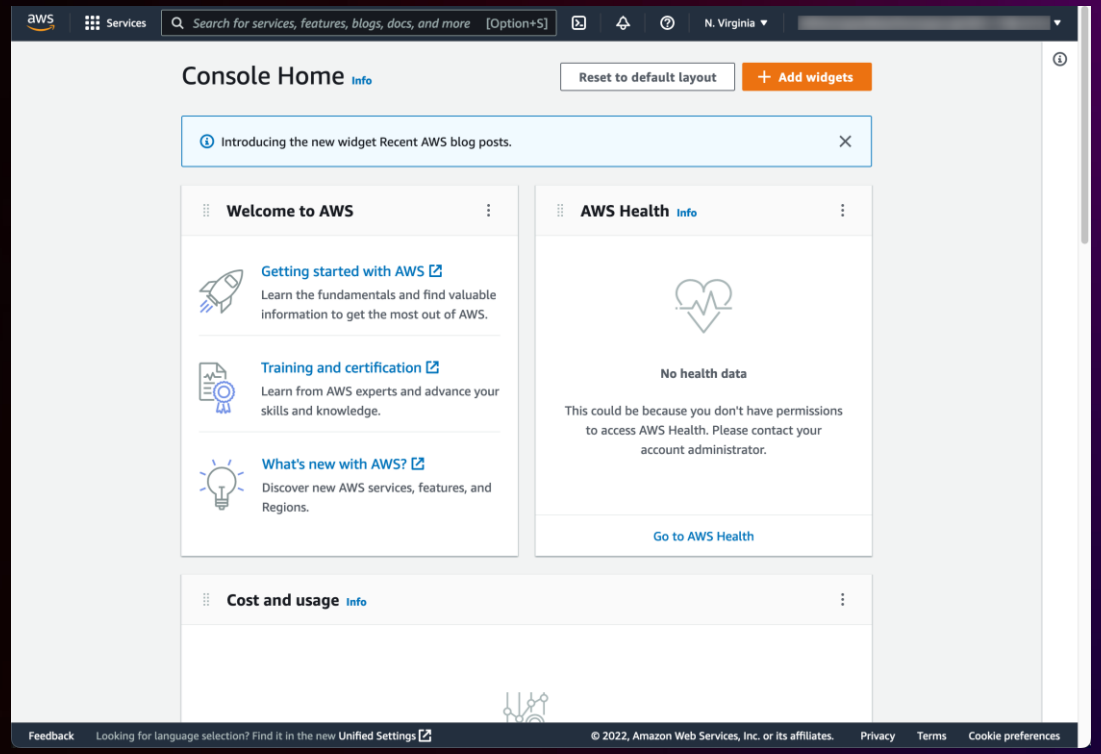
Step 3: Review terms and join event



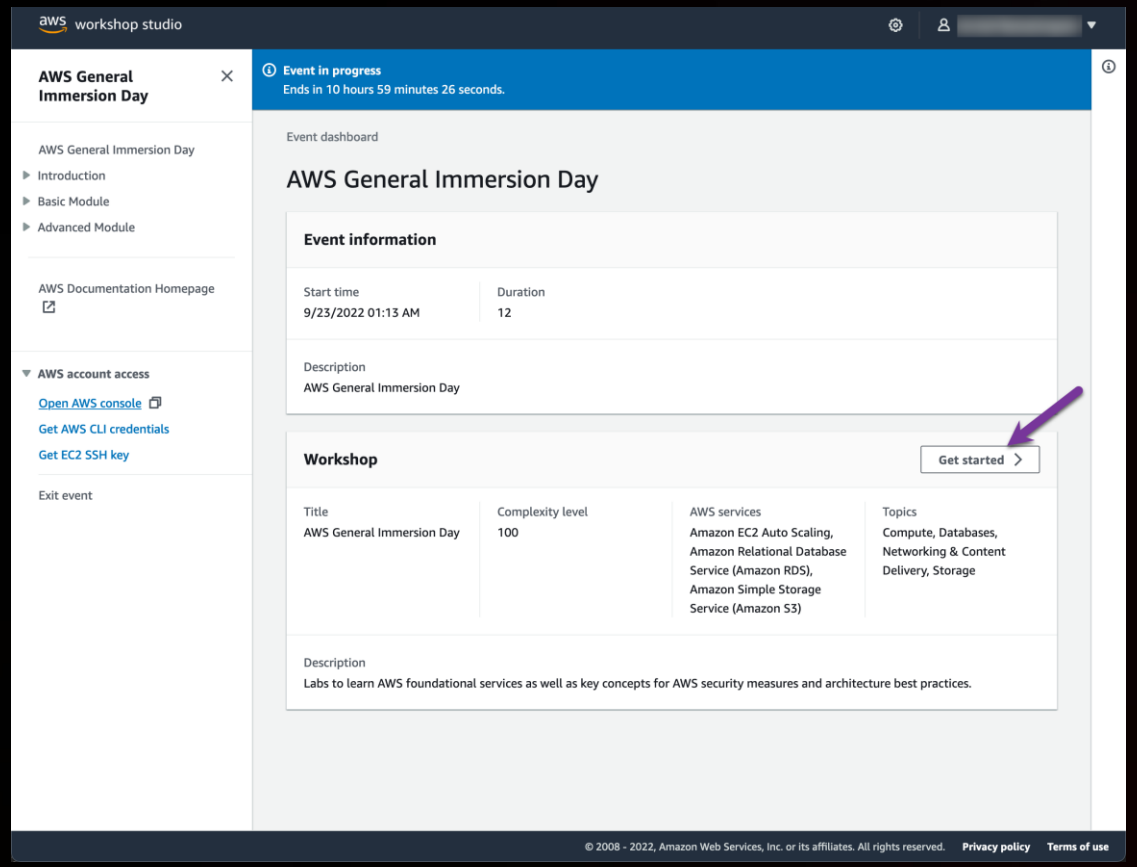
Step 4: Access AWS account

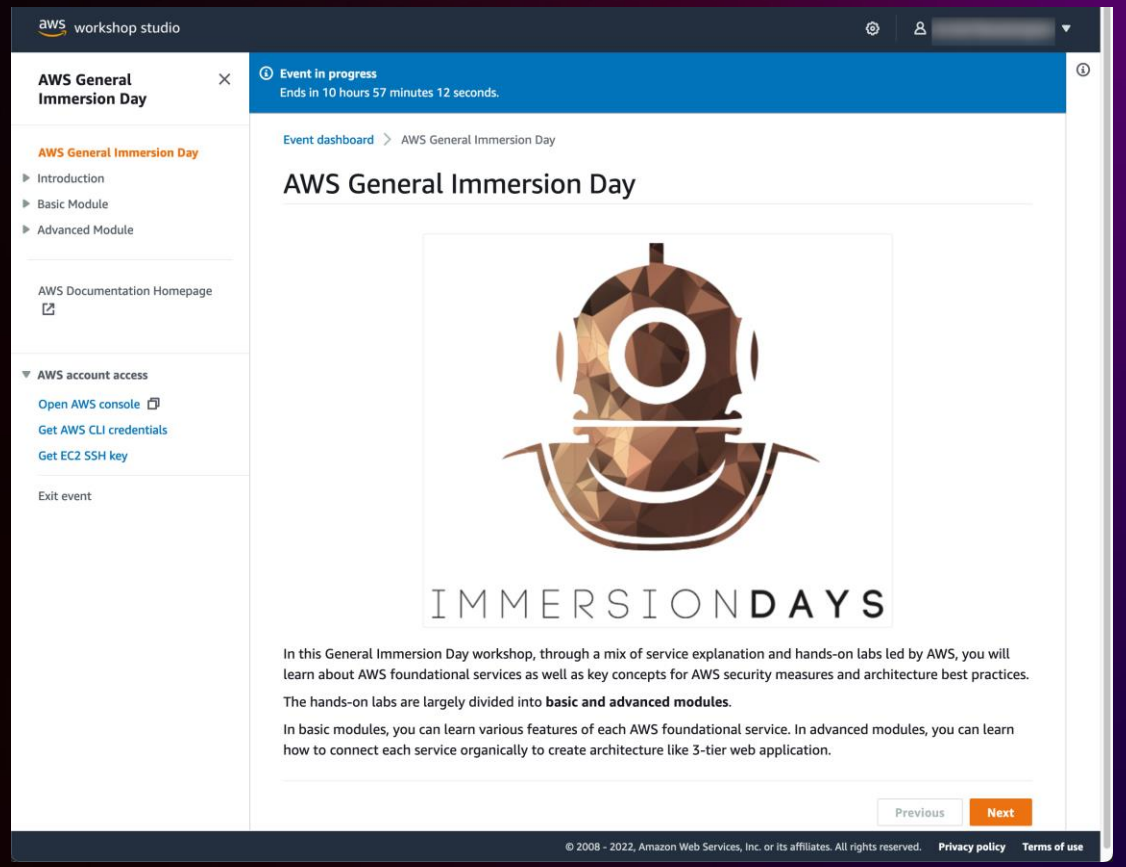
Access the AWS Console or generate AWS CLI credentials as needed.





Step 5: Get started with the workshop





Thank you!

Kelvin Lawrence
lakelvin@amazon.com
@gfxman

Taylor Riggan
triggan@amazon.com
@triggan

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