

# BUILDING MODERN APPS IN MINUTES IN THE CLOUD



Unubold Tumenbayar

Instructor at MIU

AWS Certified Solutions Architect Professional



# AGENDA

The Cloud  
GraphQL  
AWS Amplify

# What is the cloud?

Everything in this world is a system. A system has inputs, does processing, and returns outputs just like a function.

Cloud is a bunch of services. Services are essentially just an API that takes your input, does something behind the scene, and as a result, you get what you want.

```
const AWS = require("aws-sdk");
const dynamodb = new AWS.DynamoDB({ apiVersion: "2012-08-10" });
const scanParams = { TableName: tableName };
const allRecords = await dynamodb.scan(scanParams).promise();
```

# Benefits of the Cloud

- With the cloud, you **developer do less and achieve more**. The cloud provider (they are experts) deal with the technical problems. Developers contribute their time to develop business logic code. As a result, business grows faster.
- Clients won't see any interruptions. Because your applications **are always available** in multiple datacenter (or regions) and more **reliable**.
- There are a whole of bunch services that make your apps more **secure, performant**.



# Cloud Models



	IaaS (Infrastructure as a Service)	CaaS (Container as a Service)	FaaS (Function as a Service)
App runs in	virtual machines	container	the cloud
You manage	a lot of things (networking, OS, library, environment, ...)	a few things (your image)	only your business code
Scales in	a couple of minutes	a minute	a second
Costs	a lot (charges every minute + additional cost)	Depends if cluster is on EC2 or serverless	least expensive

# Where to get started with the cloud?



- Pick one cloud provider and learn [the most important core services](#). Once you learned one cloud, you can learn others easily. Because concepts behind the scene are similar.
- Don't directly jump on the IaaS model (virtual machines). You won't really get the benefits of the cloud. Start your side project using serverless services and tools like AWS Amplify. It is cost-effective, easy to start, development is faster, and so on. Once you got the taste of the serverless services. You will never go back to managing virtual machines and configuring servers.
- [Do the certificates](#). Companies appreciate that. Start with the cloud practitioner.

# What is GraphQL?



It is a query language for APIs. Gives clients the power to ask for exactly what they need and nothing more. Benefits are:

- Cleaner and less complex front-end code.
- Best practice is to do as less as API calls from your front-end. With GraphQL, you can get all the required data on the page in one shot. Hence, it improves performance.
- You can specify what attributes you need, so small data is transferred over the network.

# GraphQL components

- Schema - A description of the data.
- Resolvers - A resolver is a function that's responsible for populating the data.







# DEMO

GraphQL in action



# What is AWS Amplify?



Comprehensive tool to build full-stack web and mobile apps faster. Easy to start, easy to scale. Components are:

- Database (DynamoDB) – NoSQL DB.
- Object storage (S3) – Image, video, file storage.
- Custom logic (Lambda) – Serverless function.
- User management (Cognito) – Sign in, sign up, JWT token generation, MFA.
- API (AWS AppSync or API Gateway) – GraphQL or REST API.
- CDN (CloudFront) – Deliver static contents faster all over the world.

It also creates a CI/CD pipeline for you.

# NoSQL DB and DynamoDB

+



- If your app needs hyper scaling, use NoSQL. Supports trillions of transactions a day.
- More performant. 2-digit millisecond. You can add an accelerator, so latency is microsecond.
- Faster development. No schema. Less management. Just create a table.
- Cheaper.
- Highly available and fault-tolerant as it runs on many servers in the cloud (distributed system).
- Newer NoSQL DBs support transactions.
- Other DynamoDB features are Global tables, Streaming. Export data to another data source (such as Data Lake). Normalize and analyze with SQL using analytical services.

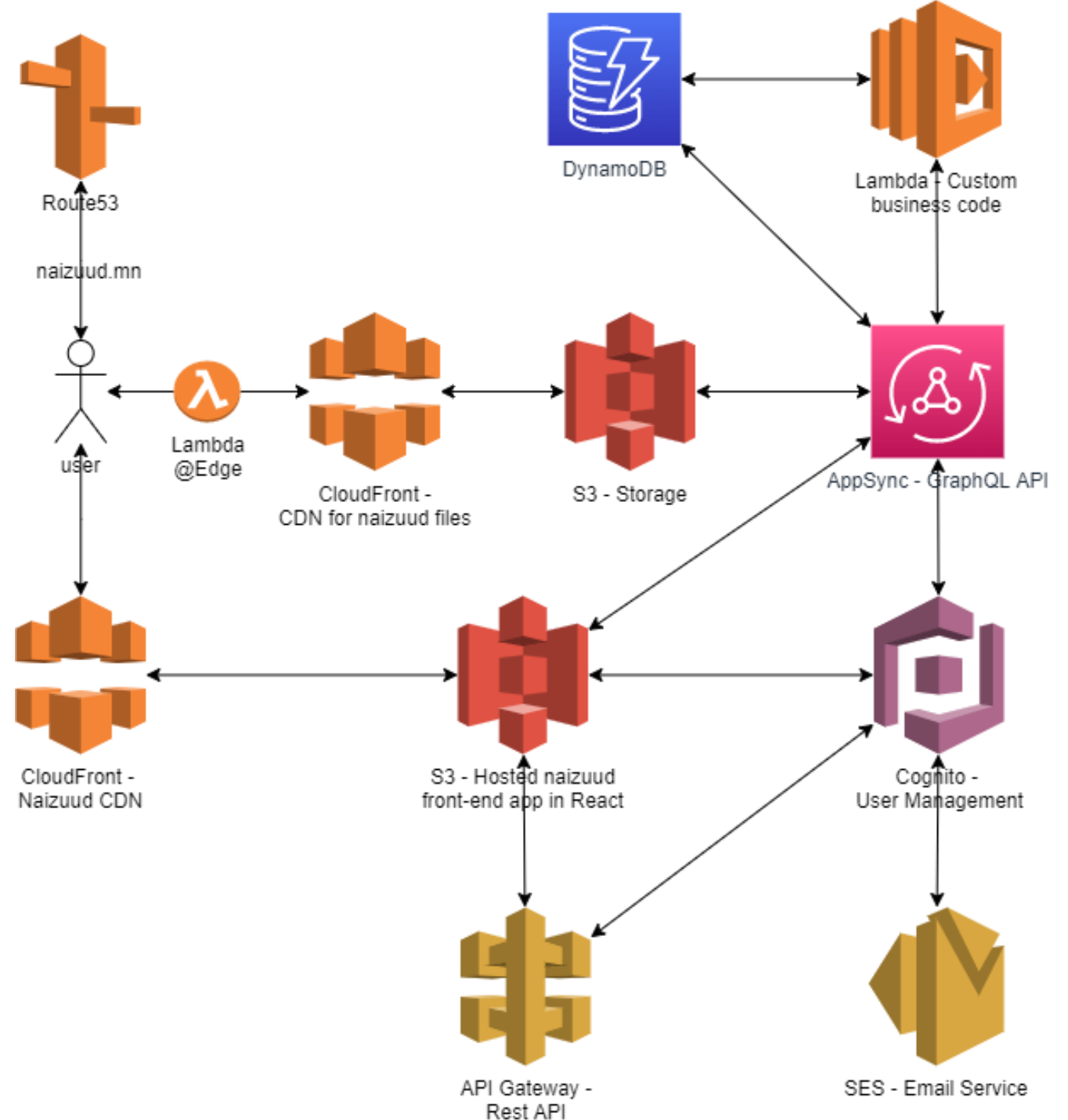


# DEMO



Building a demo app using AWS Amplify

- Architecture for a typical application created with AWS Amplify.
- Serverless. Low management overhead.
- Secure.
- Highly scalable.
- Cost effective.
- Performant.



# React



- Spread operator, destructuring, and array functions are commonly used.
- Components are isolated, independent and, reusable HTML code.
- State – Data inside the component. Treat as immutable.
- Props – Data passing from parent to child. Immutable. Can be functions. You probably need global state management tools like React Context or Redux. Or simply use LocalStorage.
- HOC adds features to your component.
- Need to call an API? Do it in componentDidMount.

# Best practices in React

+



- Use hooks and functional components.
- Utilize PureComponents, overwrite `componentShouldUpdate`, `React.memo` to prevent unnecessary re-rendering.
- Use `LocalStorage` to cache it on the client-side and eliminate number of API calls.
- Use composition to reuse your code.
- Make as less as API calls if possible from your front-end app.
- Set and show successful states first. Then make the API call asynchronously.



# Summary

- Cloud is inevitable and has many benefits. Keep learning and using it.
- Serverless and low-code software development is a great choice for small businesses and side projects.
- Infrastructure as Code (IaC) is the wand to create resources in the cloud and govern it properly.
- Component-driven libraries and frameworks are trending when building front-end apps.



+



o



.



# THANK YOU

Unubold Tumenbayar

<https://www.linkedin.com/in/unuboldtumenbayar/>