



Unleashing the Power of Serverless

Building Scalable & Cost-Effective
Applications with GraphQLGen and
AWS Lambda





Durgaprasad Budhwani

Chief Technologist @ Tech9

WHAT IS SERVERLESS ?



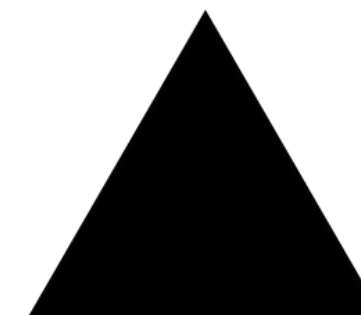
AWS



Azure



Google



Vercel

ADVANTAGES OF SERVERLESS ARCHITECTURE



Reduced infrastructure management



Scalability and automatic scaling



Cost-effective pricing model based on usage



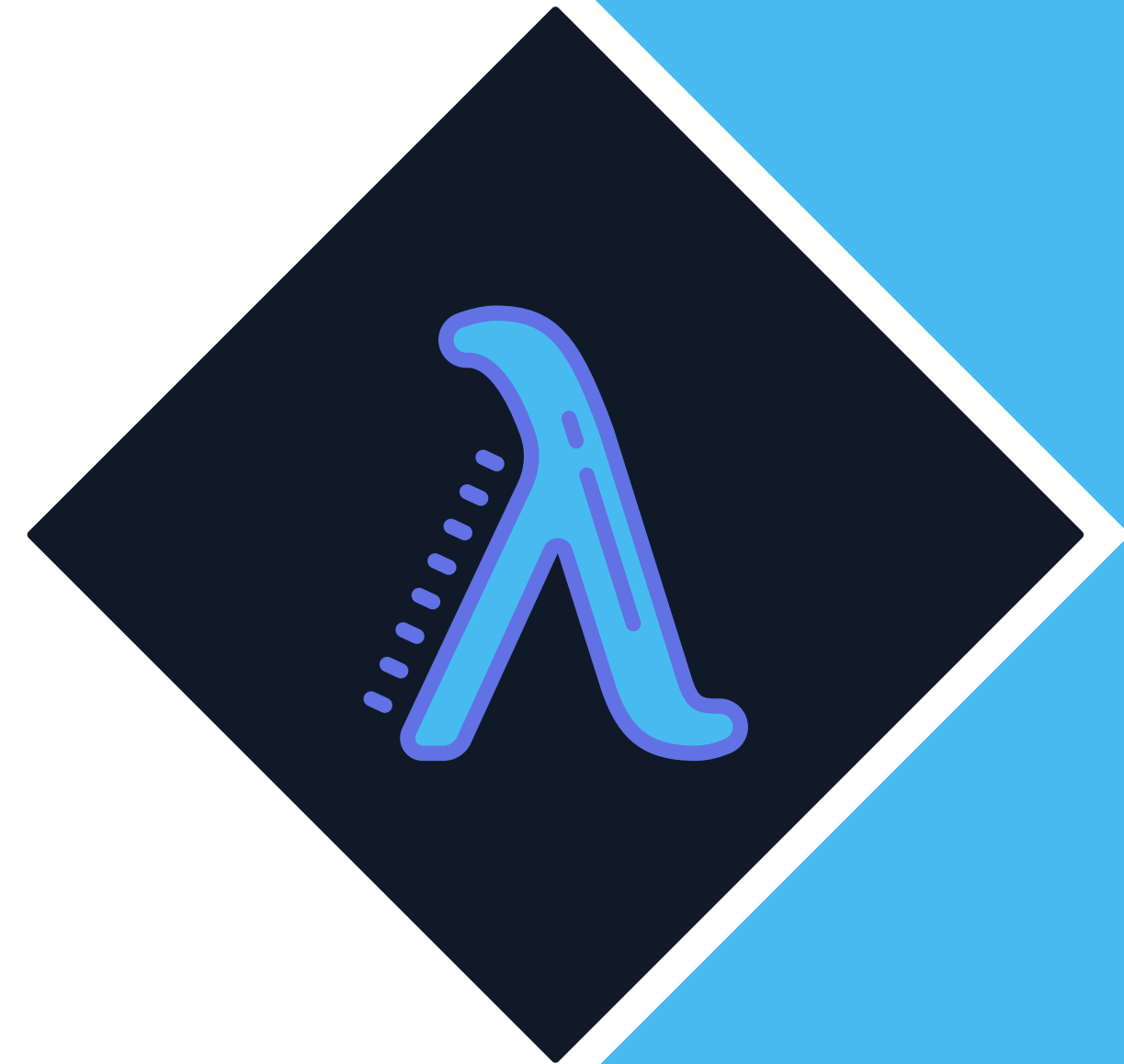
Improved developer productivity



Improved time-to-market

WHAT IS AWS LAMBDA?

AWS Lambda is a serverless compute service provided by Amazon Web Services.





HELLO WORLD IN GOLANG

```
package main

import "github.com/gin-gonic/gin"

func main() {
    // Create a new Gin router
    r := gin.Default()

    // Define a route handler for the "GET /" endpoint
    r.GET("/", func(c *gin.Context) {
        c.JSON(200, gin.H{
            "message": "Hello, World!",
        })
    })

    // Start the server on port 8080
    r.Run(":8080")
}
```

HELLO WORLD IN GOLANG FOR AWS LAMBDA ?

```
package main

import (
    "context"

    "github.com/aws/aws-lambda-go/events"
    "github.com/aws/aws-lambda-go/lambda"
    "github.com/awslabs/aws-lambda-go-api-proxy/gin"
    "github.com/gin-gonic/gin"
)

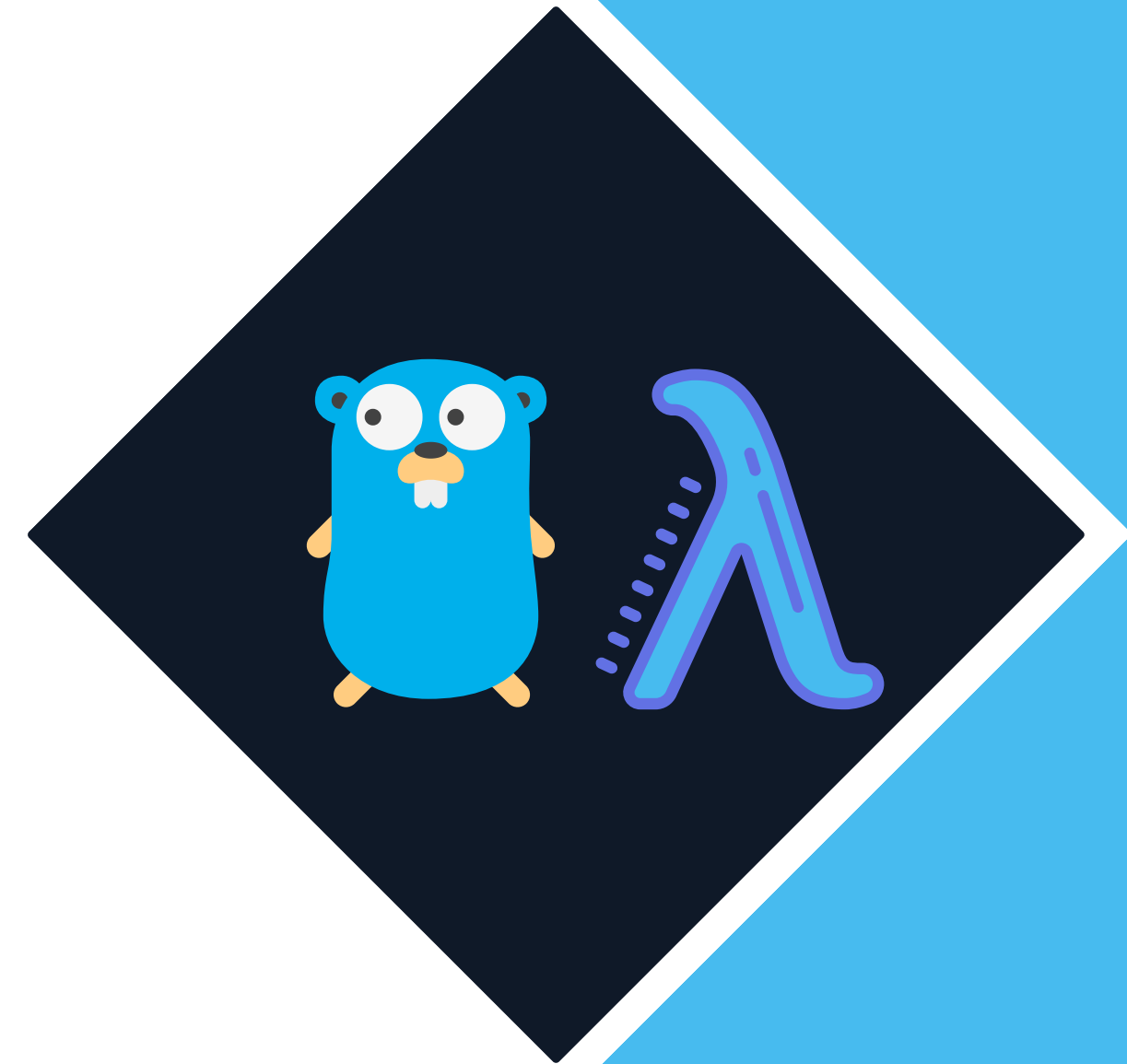
var ginLambda *ginadapter.GinLambda

func init() {
    r := gin.Default()
    r.GET("/", func(c *gin.Context) {
        c.JSON(200, gin.H{
            "message": "Hello Serverless!",
        })
    })
}

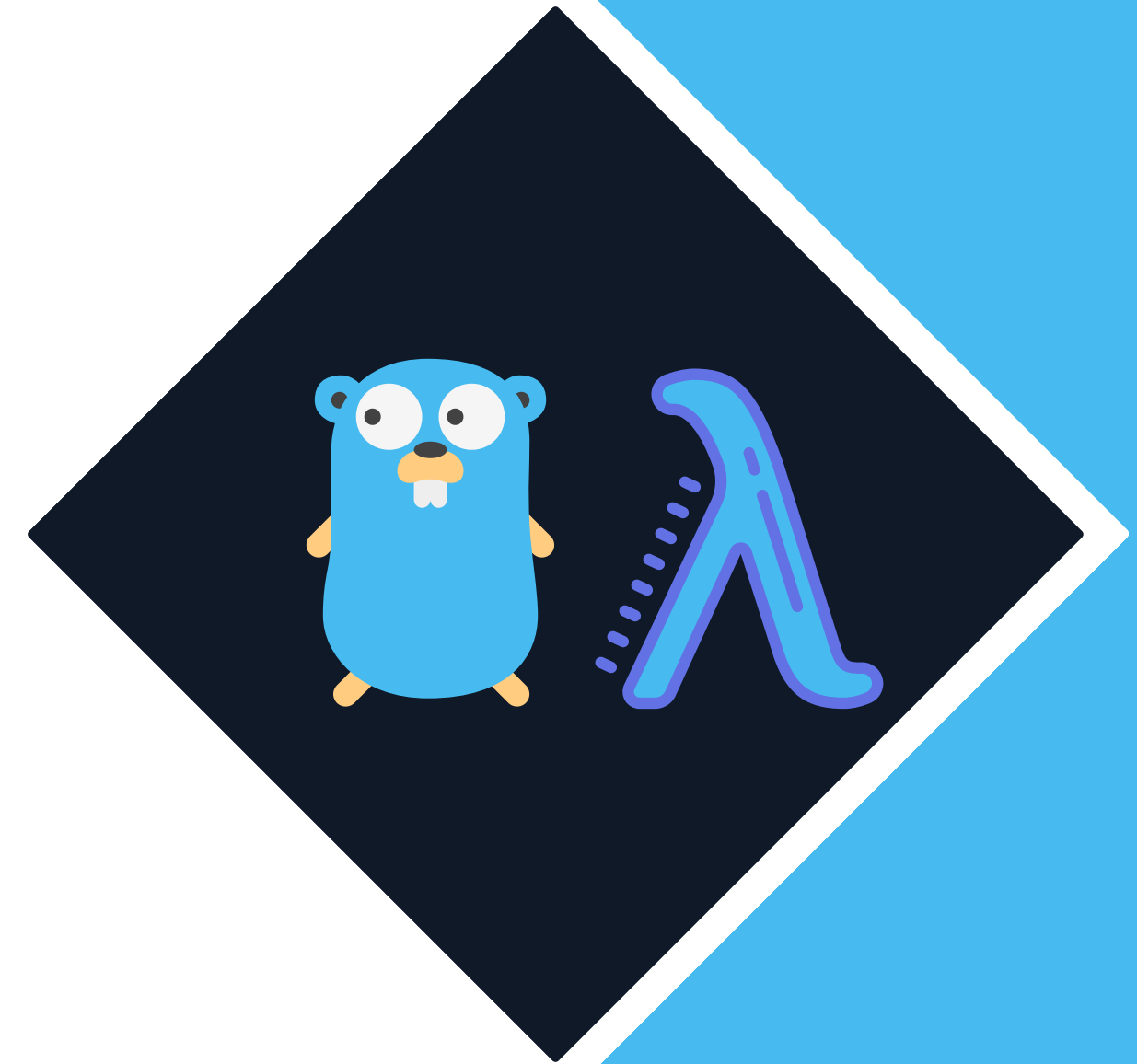
ginLambda = ginadapter.New(r)

func Handler(ctx context.Context, req events.APIGatewayProxyRequest) (
    events.APIGatewayProxyResponse, error) {
    // If no name is provided in the HTTP request body, throw an error
    return ginLambda.ProxyWithContext(ctx, req)
}

func main() {
    lambda.Start(Handler)
}
```



CODE SHARING DEMO



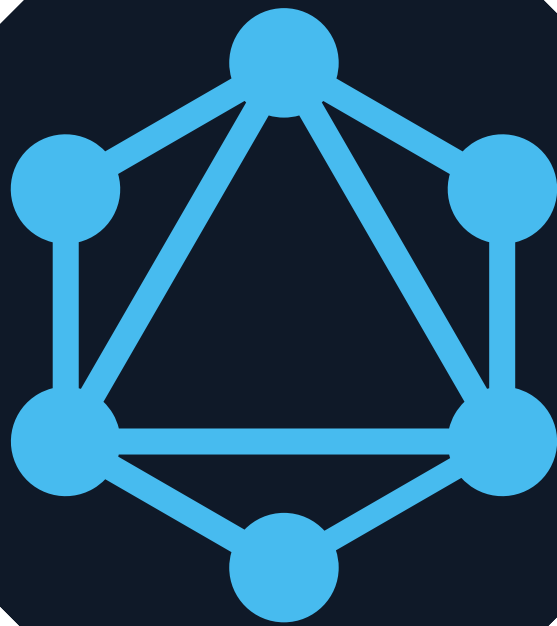
SAM FOR DEPLOYMENT



sam init

WHAT IS GRAPHQL?

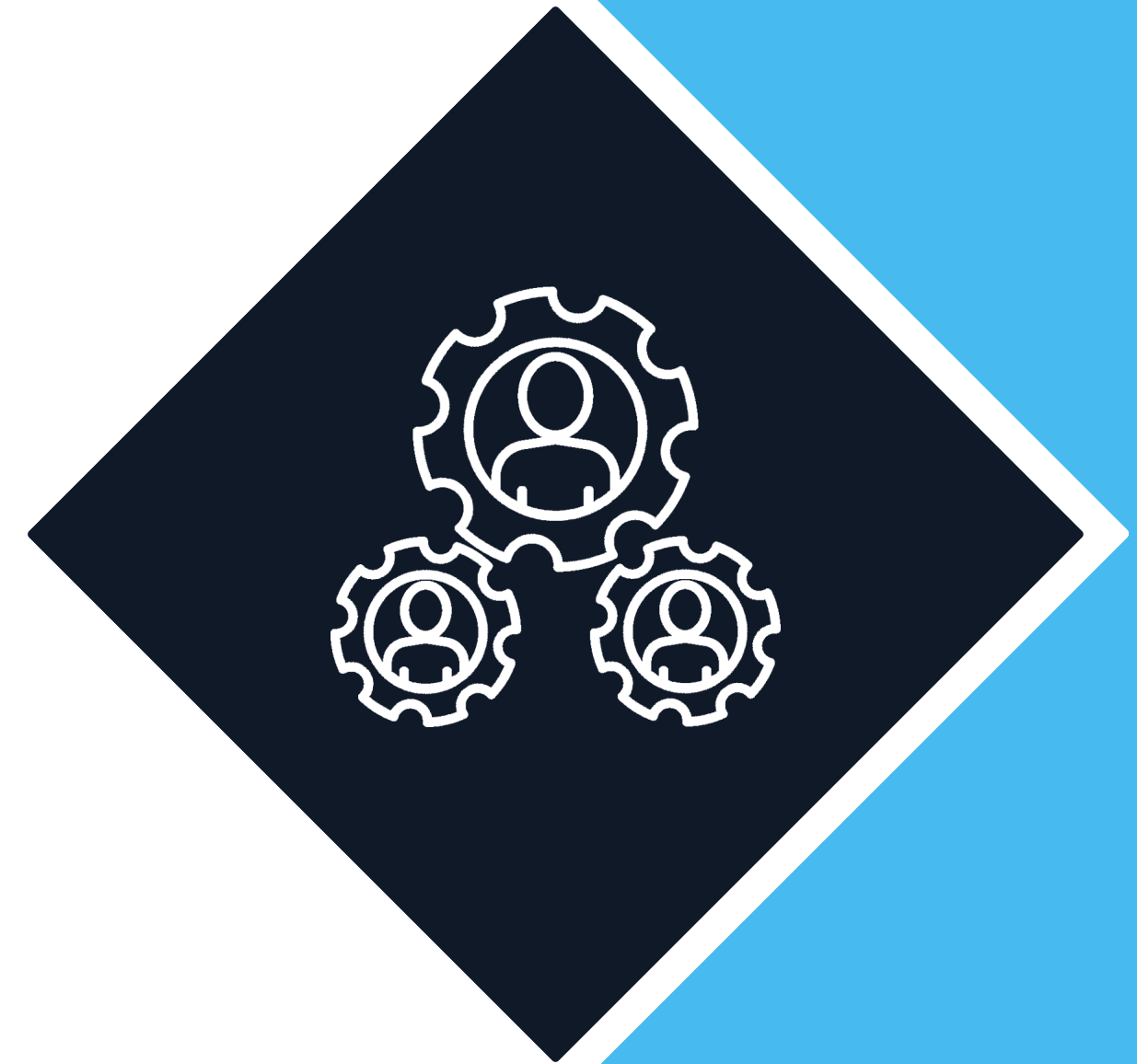
GraphQL is a query language for APIs that was designed to address the limitations of REST APIs.

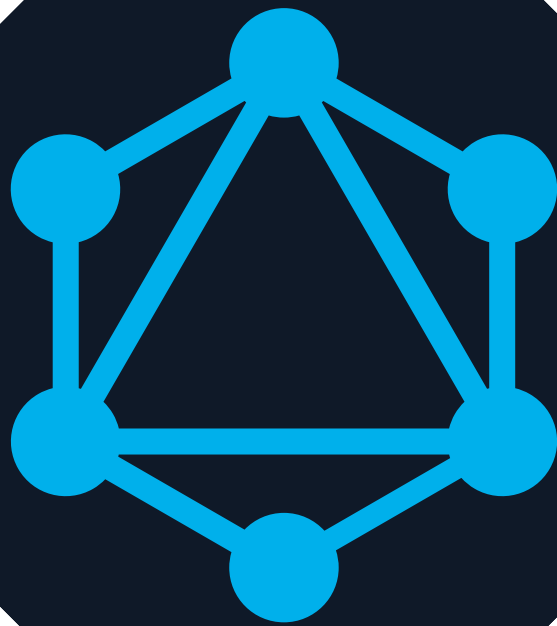


HOW DOES GRAPHQL WORK?

Schema

Client
Query/Mutation





GRAPHQL SCHEMA

```
type Query {  
  todos: [Todo!]!  
}
```

```
type Mutation {  
  createTodo(input: CreateTodoInput!): Todo!  
  updateTodo(id: ID!, input: UpdateTodoInput!): Todo!  
  deleteTodo(id: ID!): ID!  
}
```

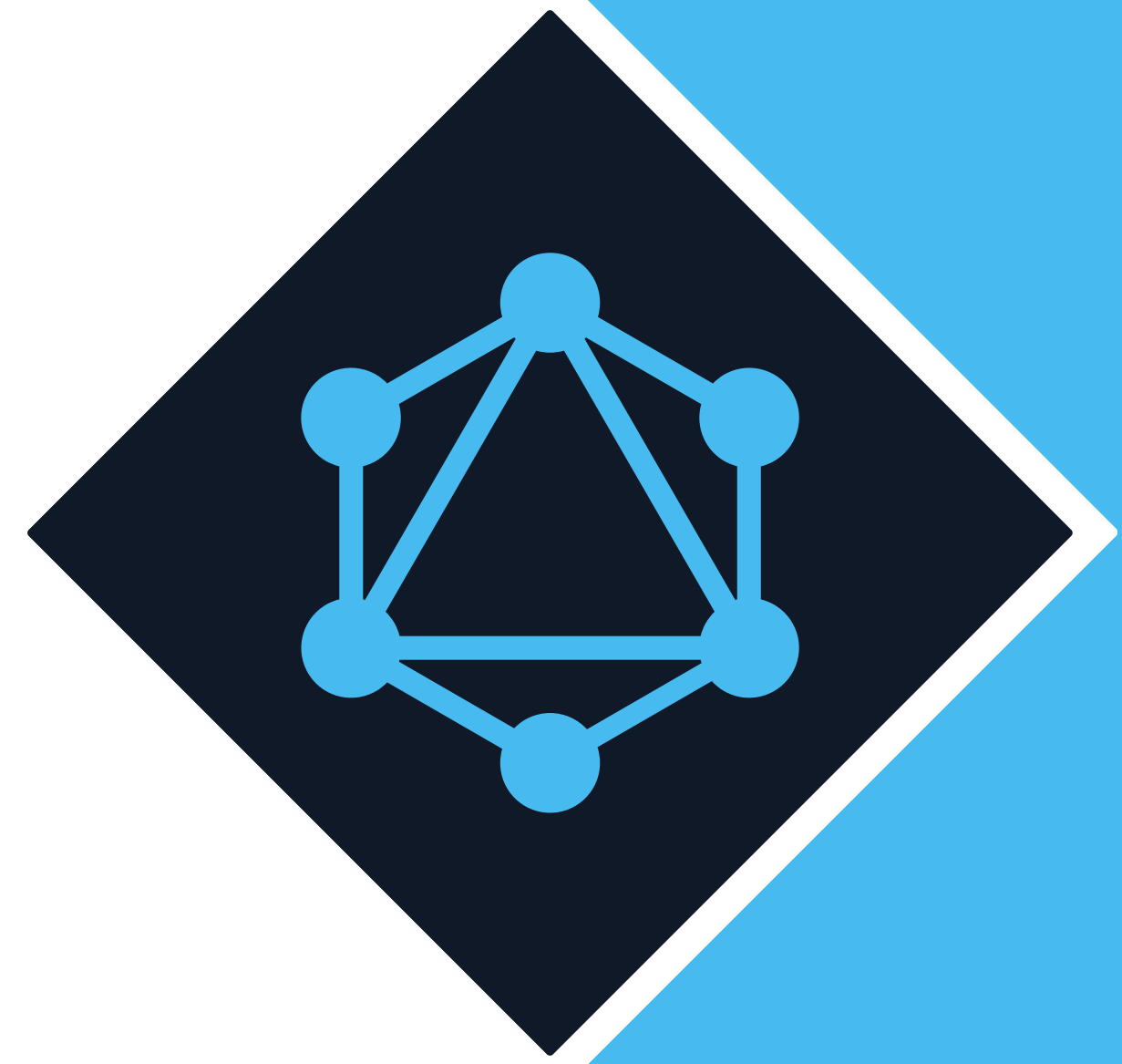
```
type Todo {  
  id: ID!  
  title: String!  
  description: String  
  completed: Boolean!  
}
```

```
input CreateTodoInput {  
  title: String!  
  description: String  
}
```

```
input UpdateTodoInput {  
  title: String  
  description: String  
  completed: Boolean  
}
```

GRAPHQL RESOLVER

```
{  
  todos {  
    id  
    title  
    completed  
  }  
}
```



WHAT IS GQLGEN?

Gqlgen is a Go library for building GraphQL APIs. It generates type-safe server code based on a GraphQL schema and functions that you define.



GQLGEN INITIALIZATION

```
go run github.com/99designs/gqlgen init
```



Thank you

