실시간 채팅 자음 퀴즈 앱 만들기

AWS IAM, AWS API Gateway, AWS Lambda, AWS DynamoDB

AUSG 2기, 김예본

발표자 소개

- 고려대학교 컴퓨터학과
- 안드로이드 개발
- Java, Python
- 前 dodotdo 안드로이드 개발자
- SW Maestro 8기
- AUSG 2기 오거나이저

오늘 함께 할 내용

시작하기 전에...

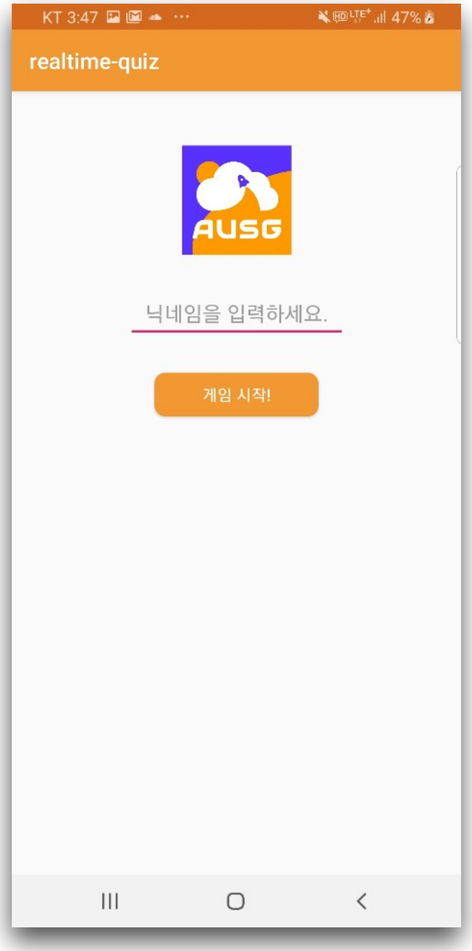
- 우리가 만들어 볼 것
- AWS Service 구조
- DynamoDB 데이터
- 데이터 이동 및 동작
- 자주 겪는 오류
- 개선점 (a.k.a 꼼수 대잔치)

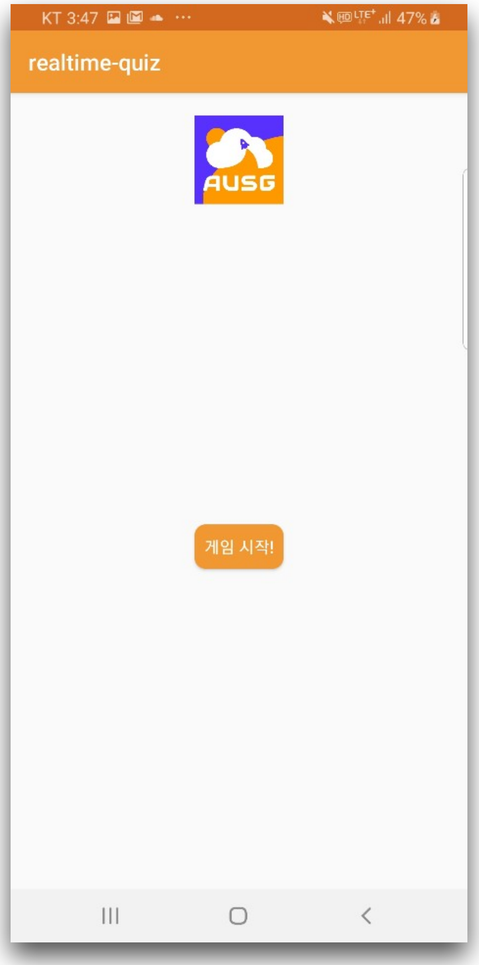
AWS 서버 개발

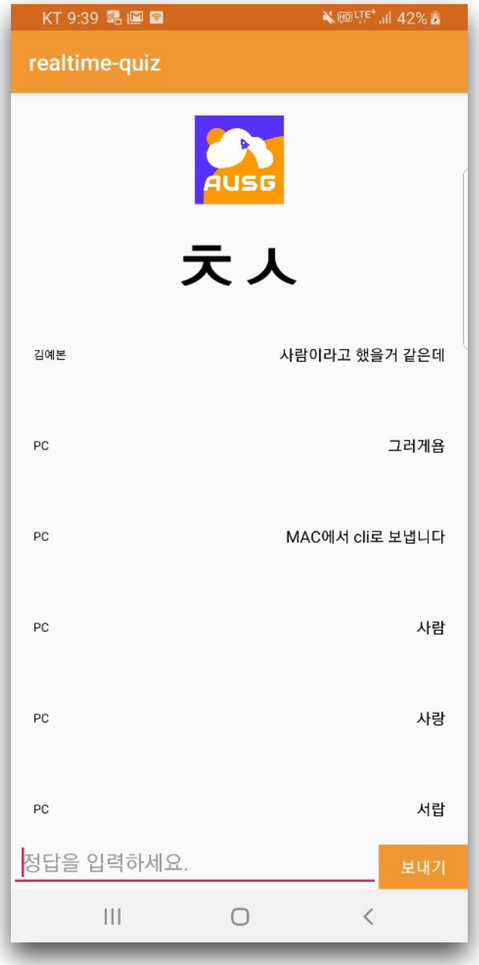
- 서버 코드 설명

우리가 만들어 볼 것

우리가 만들어 볼 것

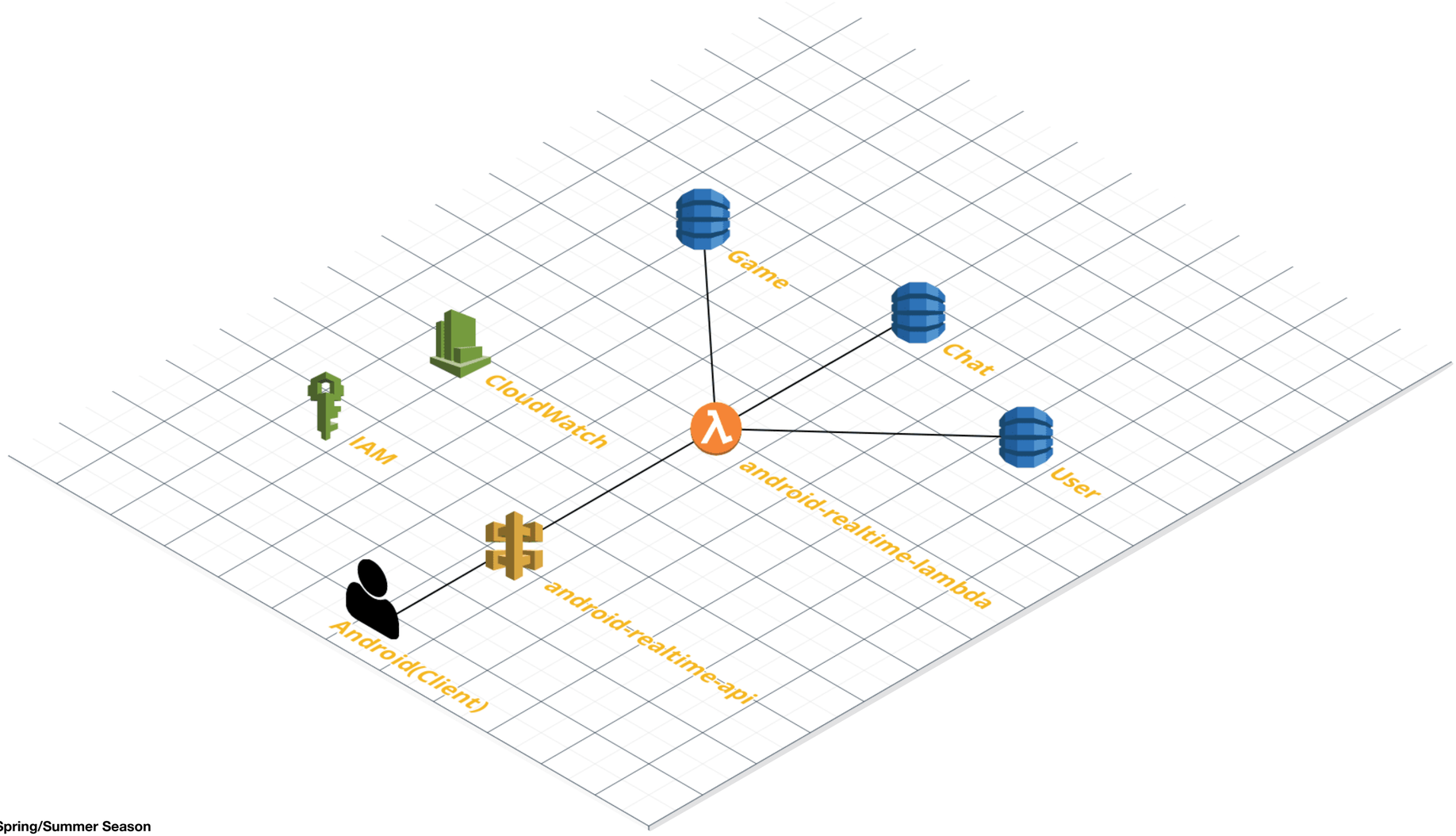

KT 3:47
47%
realtime-quiz
AUSG
닉네임을 입력하세요.
게임 시작!


KT 3:47
47%
realtime-quiz
AUSG
게임 시작!


KT 9:39
42%
realtime-quiz
AUSG
ㅊㅅ
김예본 사람이라고 했을거 같은데
PC 그러게욤
PC MAC에서 cli로 보냅니다
PC 사람
PC 사랑
PC 서랍
정답을 입력하세요.
보내기

AWS Service 구조

AWS 서비스 구조


Game
Chat
User
CloudWatch
IAM
android-realtime-lambda
android-realtime-api
Android(Client)

DynamoDB 데이터

DynamoDB 데이터

User 테이블

DynamoDB 데이터

Chat 테이블

스캔: [표] Chat: timestamp

스캔 | [표] Chat: timestamp

필터 추가

검색 시작

timestamp	content	room	username
2019-06-24T17:00:20.921Z	aa	general	aa
2019-06-24T17:02:23.000Z	aa	general	aa
2019-06-24T17:04:23.970Z	aa	general	aa
2019-06-24T17:06:07.810Z	aa	general	aa
2019-06-24T17:08:12.080Z	aa	general	aa

DynamoDB 데이터

Game 테이블

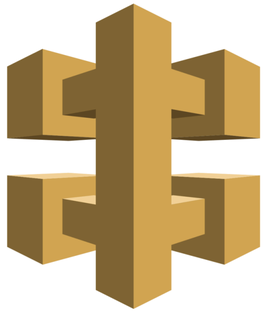
데이터 이동 및 동작

데이터 이동 및 동작

Client

채팅 및 정답 메시지 :
{“username” : “tester”, “content” : “test”}

채팅 및 정답 메시지 :
{“id":1,"nowWordIdx":1,"answeredWords":", , 강아지","nowConsonant":"ㄱㅇㅇ"}



Server

자주 겪는 오류

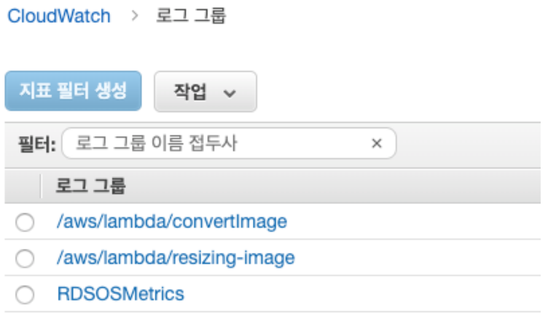
자주 겪는 오류

Python Lambda

botocore.exceptions.UnknownServiceError: Unknown service: ' Apigatewaymanagementapi' Valid service names are: acm, apigateway, application-autoscaling, appstream, athena, autoscaling, batch, budgets, ce, clouddirectory, cloudformation, cloudfront, cloudhsm, cloudhsmv2, cloudsearch, cloudsearchdomain, cloudtrail, cloudwatch, codebuild, codecommit, codedeploy, codepipeline, codestar, cognito-identity, cognito-idp, cognito-sync, config, cur, datapipeline, dax, devicefarm, directconnect, discovery, dms, ds, dynamodb, dynamodbstreams, ec2, ecr, ecs, efs, elasticache, elasticbeanstalk, elastictranscoder, elb, elbv2, emr, es, events, firehose, gamelift, glacier, glue, greengrass, health, iam, importexport, inspector, iot, iot-data, kinesis, kinesisanalytics, kms, lambda, lex-models, lex-runtime, lightsail, logs, machinelearning, marketplace-entitlement, marketplacecommerceanalytics, meteringmarketplace, mgh, mobile, mturk, opsworks, opsworkscm, organizations, pinpoint, polly, pricing, rds, redshift, rekognition, resourcegroupstaggingapi, route53, route53domains, s3, sdb, servicecatalog, ses, shield, sms, snowball, sns, sqs, ssm, stepfunctions, storagegateway, sts, support, swf, waf, waf-regional, workdocs, workspaces, xray

자주 겪는 오류

권한 문제



Cloud Watch에서 Log가 생성되지 않는다면?

자주 겪는 오류

권한 문제

WebSocket에서 Broadcast가 되지 않는다면?

자주 겪는 오류

Lambda return 형식

```
✘ ...Desktop/chatMessage/NewBotoVersion  🐍 system
$ wscat -c wss://4zqvmzxfdc.execute-api.ap-northeast-2.amazonaws.com/dev
error: Unexpected server response: 502
> %
```

```
'statusCode': 200,
'headers': {},
'body': JSON.stringify(data)
```

자주 겪는 오류

Async, await

```
async function deleteFromDyDB(params) {
    await ddb.delete(params, function(err, data) {
      if (err) {
        console.error("Unable to delete item from '" + params.TableName + "' Table. Error JSON:", JSON.stringify(err, null, 2))
      } else {
        console.log("Deleted item from '" + params.TableName + "' Table:", JSON.stringify(data, null, 2))
      }
    }).promise()
}
```

개선점 (a.k.a 꼼수 대잔치)

개선점

단어 리스트 저장 위치

```
var word = [
    "강아지", "고양이", "선풍기", "가방", "서랍", "책상", "방향", "영어", "의자", "사진"
]
```

VS

In Lambda 함수

AWS Elastic Search

개선점

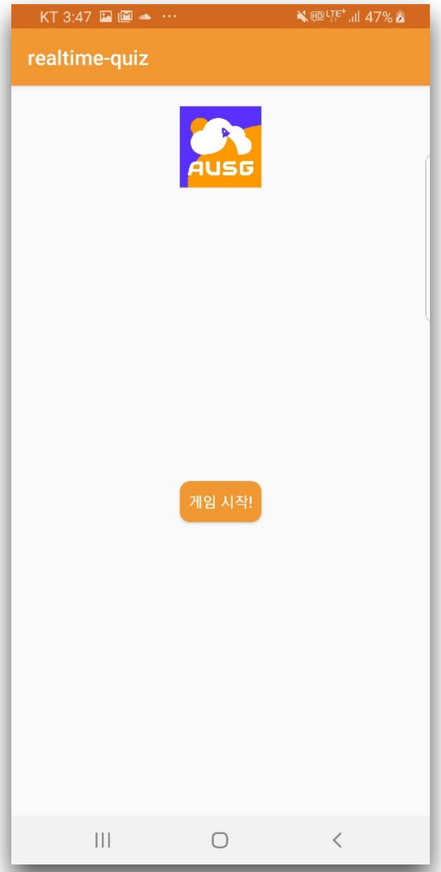
한 초성에 하나의 단어만 존재한다고 가정

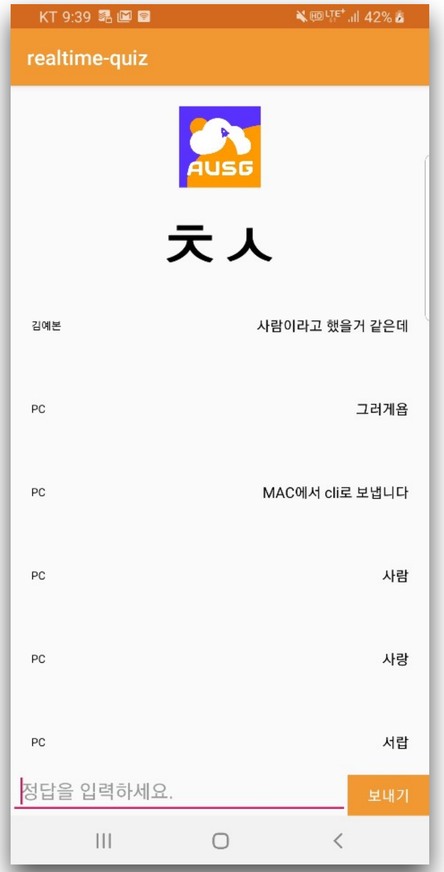
사람
서랍
사랑
서리
.
.
.

불가능

개선점

시작하기 버튼의 존재





개선점

게임 Room 생성 개념이 없음

```
if (eventType === "CONNECT") {
  console.log("Connect Requested")

  let params = {
      TableName: "User",
      Item: {
          connectionId: connectionId,
          answerCnt: 0
      }
  }
  await putToDyDB(params)

  if(await isOne()) {
    game_params.Item.nowWordIdx = 0
    game_params.Item.nowConsonant = getConsonant(word[0])
    game_params.Item.answeredWords = ", "
    await putToDyDB(game_params)
  }
```

서버 코드 설명

서버 코드 설명

$connect

```
if (eventType === "CONNECT") {
  console.log("Connect Requested")

  let params = {
      TableName: "User",
      Item: {
          connectionId: connectionId,
          answerCnt: 0
      }
  }
  await putToDyDB(params)

  if(await isOne()) {
    game_params.Item.nowWordIdx = 0
    game_params.Item.nowConsonant = getConsonant(word[0])
    game_params.Item.answeredWords = ", "
    await putToDyDB(game_params)
  }
```

서버 코드 설명

$disconnect

```
} else if (eventType === "DISCONNECT") {
  console.log("Disconnect Requested")

  let params = {
    TableName: "User",
    Key: {
      connectionId: connectionId
    }
  }

  await deleteFromDyDB(params)
```

서버 코드 설명

$disconnect

```
} else if (eventType === "MESSAGE") {
  let isJson = true
  var payload
  var keys
  try {
    payload = JSON.parse(event.body)
    keys = Object.keys(payload)
  }catch(e) {
    isJson = false
  }

  if(isJson === true && keys.includes("content") && keys.includes("username")) {
    let params = {
      TableName: "Chat",
      Item: {
        "room": "general",
        "content": payload.content,
        "timestamp": new Date().toISOString(),
        "username": payload.username
      }
    }
    await putToDyDB(params)
    await getGameData().then(async (result) => {
      if(isAnswer(payload.content)) {
        await addScore(connectionId)
        await updateGameData(payload.content)
      }
      await broadcastMsg(event, payload)
    })
  }

  await broadcastGameData(event)
```

서버 코드 설명

Broadcast

```
async function broadcastMsg(event, payload) {
  try {
    connectionData = await ddb.scan({ TableName: "User", ProjectionExpression: 'connectionId' }).promise()
  } catch (e) {
    return { statusCode: 500, body: e.stack }
  }

  const apigwManagementApi = new AWS.ApiGatewayManagementApi({
    endpoint: event.requestContext.domainName + '/' + event.requestContext.stage
  })

  const postCalls = connectionData.Items.map(async ({ connectionId }) => {
    try {
      await apigwManagementApi.postToConnection({ ConnectionId: connectionId, Data: JSON.stringify(payload) }).promise()
    } catch (e) {
      return { statusCode: 500, body: e.stack }
    }
  })

  try {
    await Promise.all(postCalls)
  } catch (e) {
    return { statusCode: 500, body: e.stack }
  }
}
```

감사합니다

다음에 또 봬요!

이 설문조사는 영국에서 시작되어… 🕵️

오늘 어떤 모임에 참여하셨나요? → ‘대학생’ 선택

한줄 요약: 완료하시면 부자된대요😉