



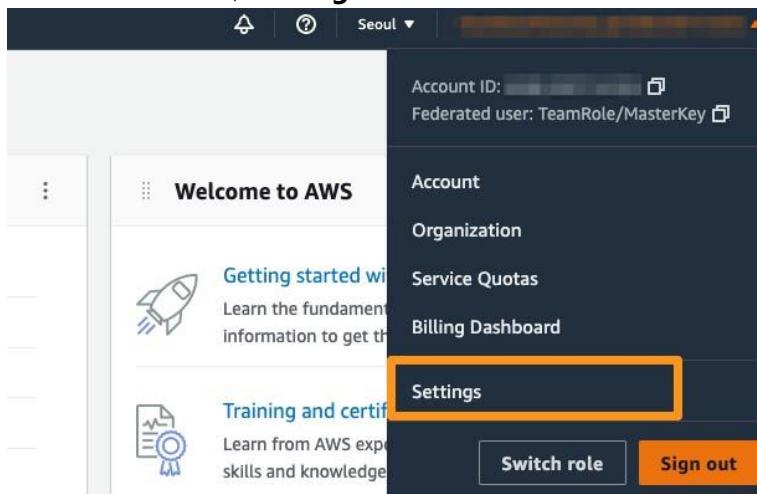
AWS IoT Application
Amplify, AppSync, Cognito
Hands-on Lab

Jan 11, 2023

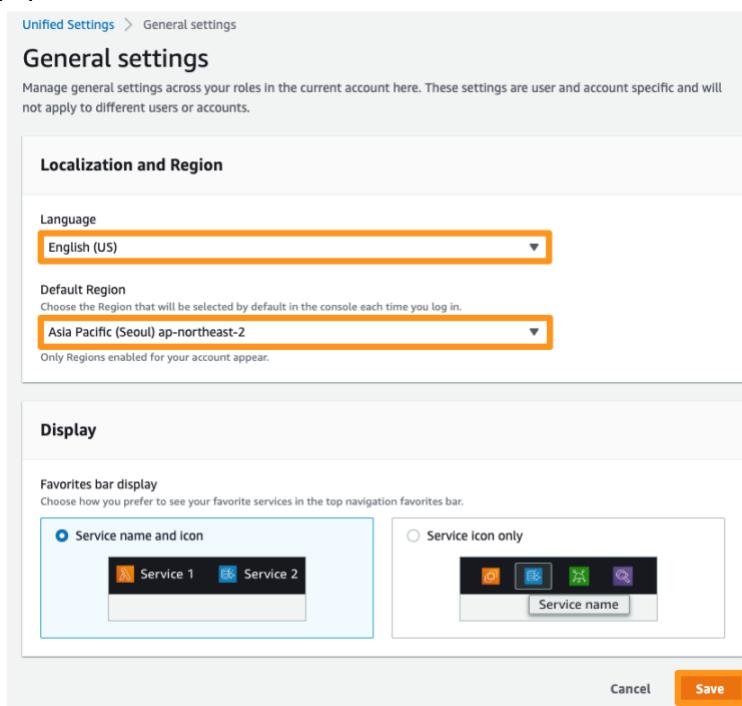
1. 사전 준비 사항

AWS 관리 콘솔(<http://console.aws.amazon.com>)에 접속 후 아래 내용을 확인해 주시기 바랍니다. 본 행사에서는 [AWS EventEngine](#) 을 사용하여 콘솔 창에 접속합니다. 연결된 링크로 접속하여 해당 가이드를 참고합니다.

- 1-1. 본 실습은 **서울 리전(ap-northeast-2)**에서 진행합니다.
- 1-2. 본 실습 문서는 영어 AWS 관리 콘솔을 기준으로 작성되었습니다. 관리 콘솔 우측 상단에 계정 정보를 클릭한 후, **Settings** 메뉴에서 원하시는 언어로 변경할 수 있습니다.



- 1-3. 뿐만 아니라 자주 사용하는 리전이 있을 경우, **Default Region** 탭에서 설정할 수 있습니다.

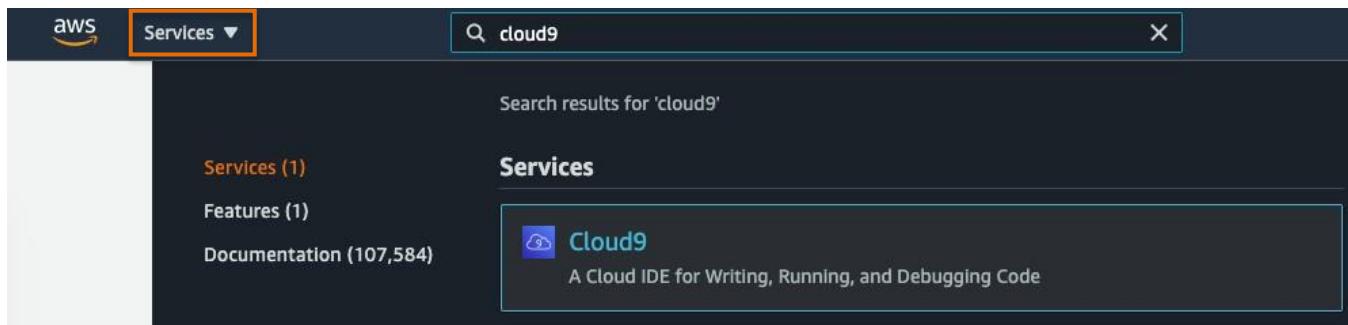


2. AWS Cloud9 으로 실습 환경 구축하기

AWS Cloud9은 브라우저만으로도 코드를 작성, 실행 및 디버깅할 수 있는 IDE입니다. 코드 편집기, 디버거 및 터미널이 포함되어 있으며 많이 사용되는 프로그래밍 언어를 위한 필수 도구가 사전에 패키징되어 제공되므로, 새로운 프로젝트를 시작하기 위해 파일을 설치하거나 개발 머신을 구성할 필요가 없다는 특징을 가지고 있습니다.

본 실습에서는 AWS 관리 콘솔을 사용하여 AWS Cloud9으로 실습 환경을 구축하고 IoT 애플리케이션 빌딩 작업을 진행합니다.

2-1. AWS 콘솔 좌측 상단에 있는 Services 메뉴를 클릭한 후, 검색창에 Cloud9을 입력한 후, 아래 화면과 같은 결과가 나오면 해당 서비스를 선택합니다.



2-2. **Create environment** 버튼을 누르고 화면과 같이 진행합니다.



Name에 **iot-workshop**을 입력하고 Next step으로 갑니다.

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.
 Limit: 60 characters

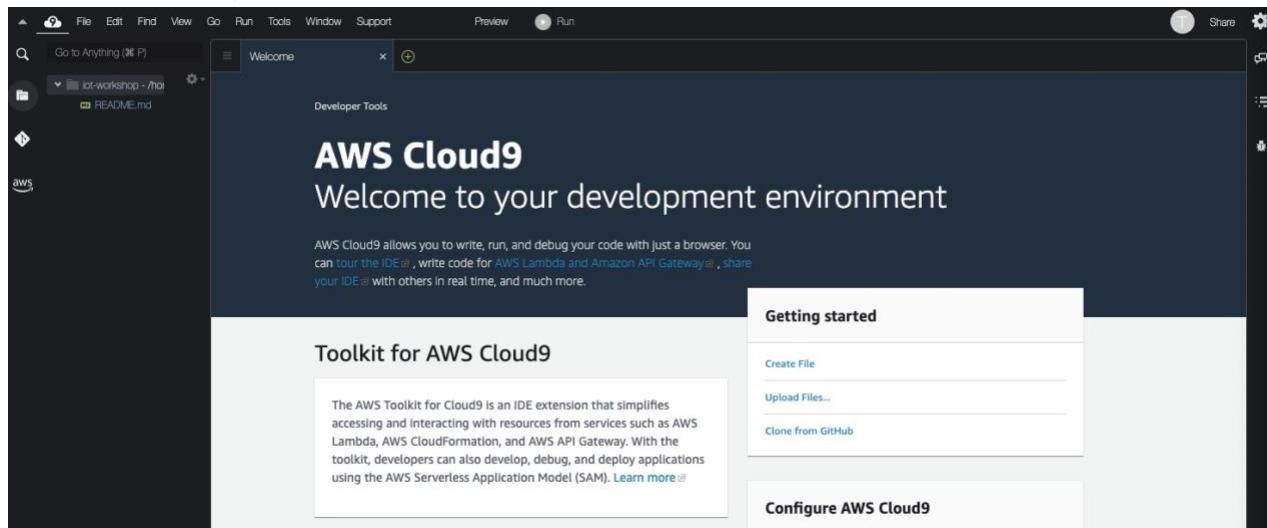
Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.
 Limit: 200 characters

Cancel **Next step**

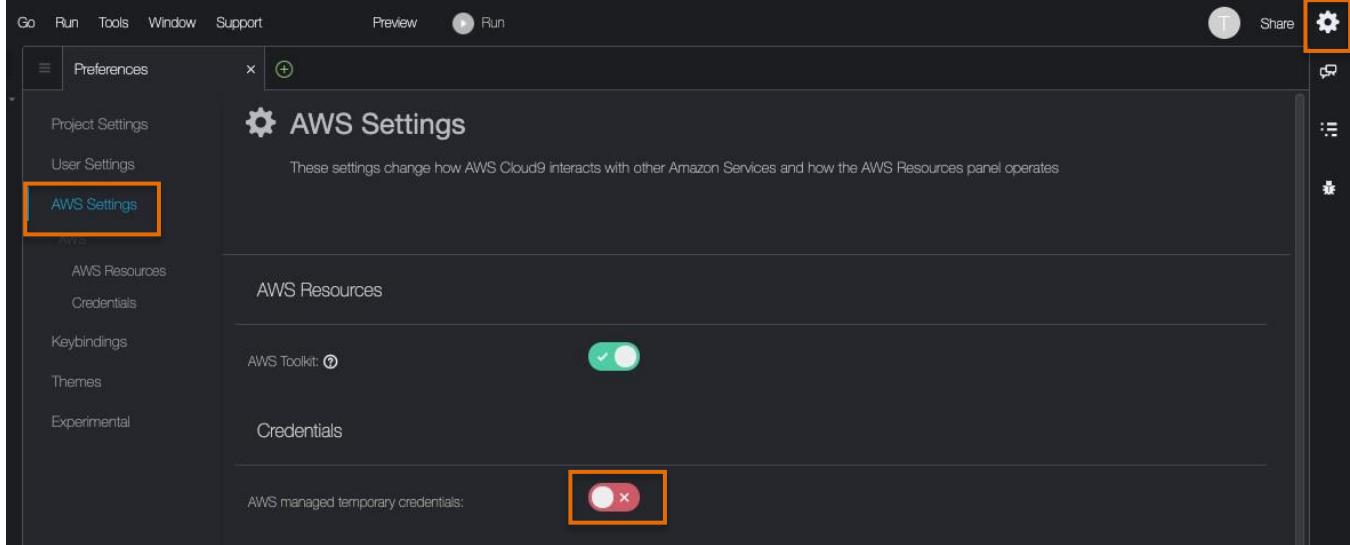
다음 페이지의 모든 설정을 아래와 같이 설정한 후, 다음 단계로 이동합니다.

Environment type	Create a new EC2 instance for environment(direct access)
Instance type	m5.large (8 GiB RAM + 2 vCPU)
Platform	Amazon Linux 2 (recommended)

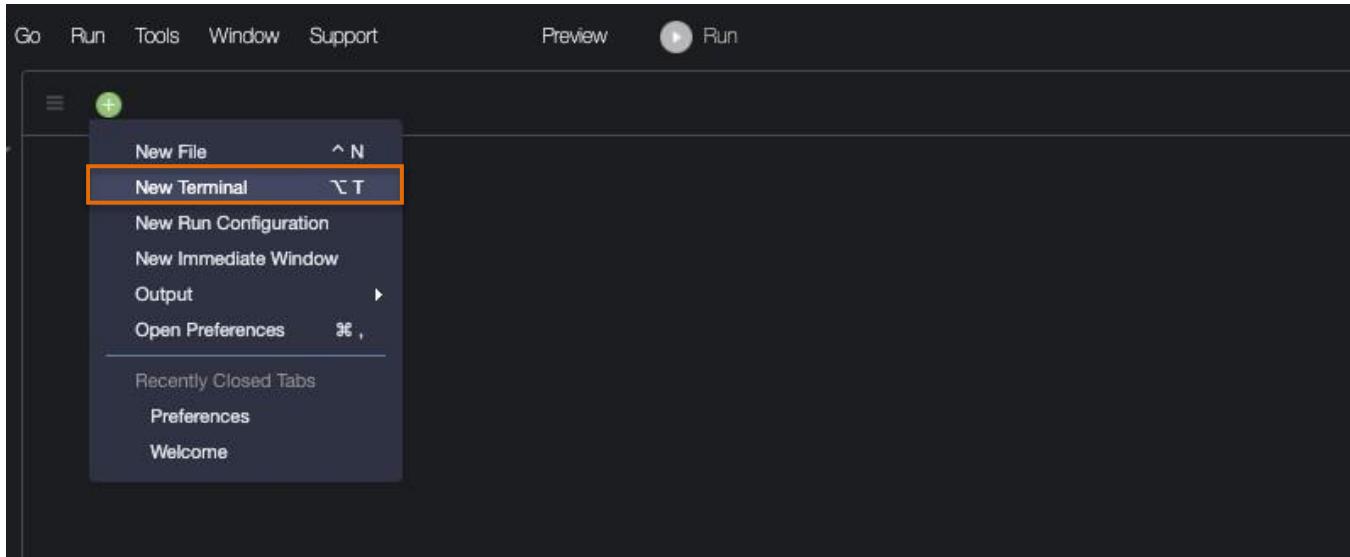
마지막 페이지에서 설정 값을 확인한 후, **Create environment** 버튼을 클릭하여 Cloud9 실습 환경을 구축합니다. 잠시 후, Cloud9 IDE로 이동된 것을 볼 수 있습니다.



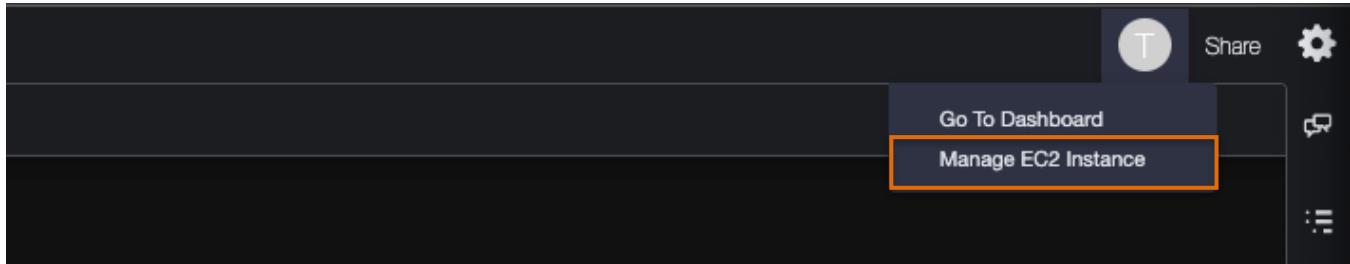
2-3. 오른쪽 상단에 있는 톱니바퀴 모양을 클릭한 후, 사이드 바에서 **AWS Settings** 메뉴를 선택합니다. Credentials에서 AWS managed temporary credentials 를 비활성화합니다. 설정이 완료되면 창을 닫습니다.



2-4. Preferences 창을 닫고 새로운 terminal 창을 엽니다.



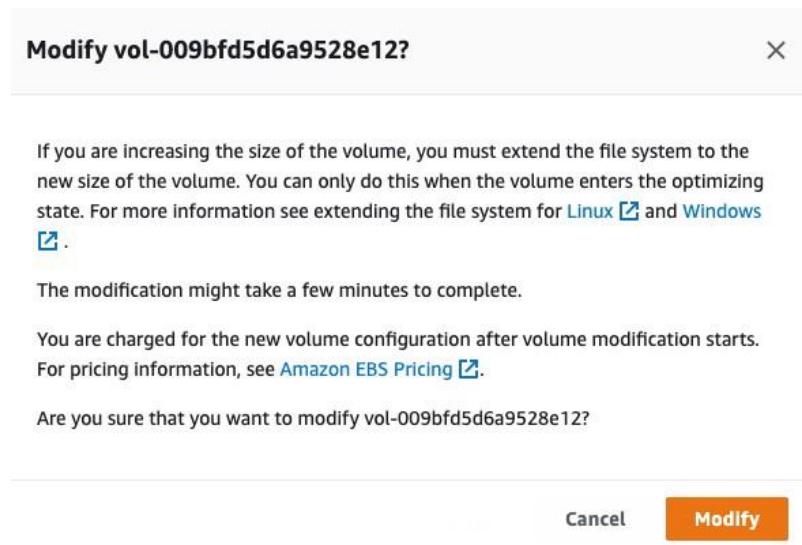
2-5. Cloud9의 볼륨은 기본적으로 10GiB가 제공됩니다. 원활한 실습 진행을 위해 디스크 저장 공간을 늘리는 작업을 수행합니다. 오른쪽 상단에 알파벳이 기입된 동그라미를 클릭한 후, **Manage EC2 Instance** 를 클릭합니다.



EC2 페이지가 나오면 사이드 바에서 Elastic Block Store > **Volumes** 를 클릭합니다. 화면에 보이는 볼륨을 선택한 후, 우측 상단 Actions > **Modify volume** 을 클릭합니다.

Name	Volume ID	Type	Size	IOPS
-	vol-009bfd5d6a9528e12	gp2	10 GiB	100

볼륨 사이즈를 10GiB 에서 **30GiB** 로 변경한 후, **Modify** 버튼을 클릭합니다. 볼륨 크기를 늘리기 위해 파일 시스템의 볼륨을 새 크기로 확장해야 한다는 안내창이 나옵니다. Modify 버튼을 누릅니다.



다시 Cloud9 IDE에 접속하여 터미널 창에서 아래의 명령어를 순서대로 입력합니다.

```
sudo growpart /dev/nvme0n1 1
sudo
xfs_growfs -d /
```

마지막으로 아래의 명령어를 입력하면 아래와 같이 볼륨이 30GiB로 증가한 것을 확인할 수 있습니다.
df -h

```
TeamRole:~/environment $ sudo growpart /dev/nvme0n1 1
CHANGED: partition=1 start=4096 old: size=20967391 end=20971487 new: size=62910431 end=62914527
TeamRole:~/environment $ sudo xfs_growfs -d /
meta-data=/dev/nvme0n1p1      isize=512    agcount=6, agsize=524159 blks
                           =      sectsz=512   attr=2, projid32bit=1
                           =      crc=1     finobt=1 spinodes=0
data      =      bsize=4096   blocks=2620923, imaxpct=25
                           =      sunit=0   swidth=0 blks
naming    =version 2        bsize=4096   ascii-ci=0 ftype=1
log       =internal         bsize=4096   blocks=2560, version=2
                           =      sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none            extsz=4096   blocks=0, rtextents=0
data blocks changed from 2620923 to 7863803
TeamRole:~/environment $ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   0    3.8G  0% /dev
tmpfs          3.8G   0    3.8G  0% /dev/shm
tmpfs          3.8G  460K  3.8G  1% /run
tmpfs          3.8G   0    3.8G  0% /sys/fs/cgroup
/dev/nvme0n1p1  30G  8.0G  23G  27% /
tmpfs         777M   0   777M  0% /run/user/1000
```

만약 Cloud9 생성 시, 인스턴스 타입을 T2로 선택했을 경우, 아래의 명령어로 볼륨 크기를 늘려야

```
sudo growpart /dev/xvda
1
```

3. Amplify 설치하기

AWS Amplify Framework는 클라이언트용 프레임워크입니다. 그리고 해당 프레임워크를 구성하는 요소 중 하나인 Amplify CLI 및 콘솔을 이용하여 AWS 서비스 생성 및 전체 애플리케이션을 빌드, 테스트, 배포 및 호스팅할 수 있습니다. 본 챕터에서는 Amplify를 설치하고 구성하는 작업을 수행합니다.

3-1. 아래의 명령어를 이용하여 Amplify CLI를 설치합니다.

```
npm install -g @aws-amplify/cli
```

3-2. 설치가 완료되면 Amplify 환경을 설정합니다.

```
amplify configure
```

설정 값을 입력하기 위해, 엔터를 누릅니다. 그 다음 **키보드 아래 방향 버튼을 사용하여 서울 리전 코드(ap-northeast-2)를 선택합니다.**

```
dev:~/environment $ amplify configure
Follow these steps to set up access to your AWS account:

Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue
● spawn xdg-open ENOENT

Resolution: Please report this issue at https://github.com/aws-amplify/amplify-console/issues/new?title=CLI+Region+selection+issue+in+Linux+distros+like+Ubuntu+18.04+and+Debian+10.0+using+xdg-open+to+open+the+region+selection+menu
Learn more at: https://docs.amplify.aws/cli/project/troubleshooting/

Session Identifier: 38cbe746-052f-4477-91ca-96b154034192

Specify the AWS Region
? region:
  eu-west-3
  eu-central-1
  ap-northeast-1
  > ap-northeast-2
  ap-southeast-1
  ap-southeast-2
  ap-south-1
(Move up and down to reveal more choices)■
```

그 다음 임의의 유저 값과 함께 새로운 IAM User를 생성하기 위한 질문이 나옵니다.

엔터를 누른 후, **초록색 링크를 클릭**하면 유저 생성하는 콘솔창이 나옵니다.

```
Specify the username of the new IAM user:
? user name: amplify-u4rjz
Complete the user creation using the AWS console
https://console.aws.amazon.com/iam/home?region=ap-northeast-2#/users$new?step=final&accessKey&secretKey
Press Enter to continue
● spawn xdg-open ENOENT

Resolution: Please report this issue at https://github.com/aws-amplify/amplify-console/issues/new?title=CLI+User+creation+issue+in+Linux+distros+like+Ubuntu+18.04+and+Debian+10.0+using+xdg-open+to+open+the+user+creation+wizard
Learn more at: https://docs.amplify.aws/cli/project/troubleshooting/

Session Identifier: 38cbe746-052f-4477-91ca-96b154034192
■
```

아래의 화면에서 Next:Permissions 버튼을 클릭합니다.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type* **Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

AdministratorAccess-Amplify 권한이 선택된 것을 확인한 후, Next:Tags 버튼을 클릭합니다.

Add user

1 2 3 4 5

▼ Set permissions

Filter policies ▾ Search Showing 864 results

	Policy name ▾	Type	Used as
<input type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (7)
<input checked="" type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	Permissions policy (3)
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	Amazon_EventBridge_Invoke_SageMaker_Pipeline_1337222507	Customer managed	Permissions policy (1)
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	Permissions policy (1)
<input type="checkbox"/>	AmazonAppFlowFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppFlowReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppStreamFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppStreamPCAAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppStreamReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppStreamServiceAccess	AWS managed	Permissions policy (1)
<input type="checkbox"/>	AmazonAthenaFullAccess	AWS managed	Permissions policy (1)
<input type="checkbox"/>	AmazonAugmentedAIFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAugmentedAIHumanLoopFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAugmentedAIIntegratedAPIAccess	AWS managed	None
<input type="checkbox"/>	AmazonBraketFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonBraketJobsExecutionPolicy	AWS managed	None

Cancel Previous Next: Tags

본 실습에서는 태그를 별도로 지정하지 않기에 Next:Review 버튼을 클릭합니다. 권한을 확인한 다음, Create user 버튼을 클릭합니다. 아래와 같은 화면이 나오면 CSV 파일을 다운로드 받습니다.

Add user

1 2 3 4 5

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://617073285002.sigin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key
▶ <input checked="" type="checkbox"/>	amplify-SGpvP	AKIAY7LDPC6FI3NDVNHF 	***** Show

다시 Cloud9 IDE 환경으로 돌아와 'Enter'키를 누른 후

방금 다운 받은 CSV 파일에 있는 **Access key ID** 및 **Secret access key** 를
복사/붙여넣기합니다.

```
TeamRole:~/environment $ amplify configure
Follow these steps to set up access to your AWS account:

Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue

Specify the AWS Region
? region: ap-northeast-2
Specify the username of the new IAM user:
? user name: amplify-SGpvP
Complete the user creation using the AWS console
https://console.aws.amazon.com/iam/home?region=ap-northeast-2#/users$new?step=final&accessKey&userNames=amplify-SGpvP&permissionType=policies&policies=arn:aws:iam::aws:policy%2FAdministratorAccess-Amplify
Press Enter to continue

Enter the access key of the newly created user:
? accessKeyId: *****
? secretAccessKey: *****
This would update/create the AWS Profile in your local machine
? Profile Name: default

Successfully set up the new user.
```

Profile Name 은 default 로 두고 'Enter'를 클릭합니다.

여기까지 amplify 를 구성하기 위한 IAM 권한 생성과 AWS 에 접속하기 위한 credential 설정이 완료 되었습니다.

4. 워크샵 소스 코드 다운로드

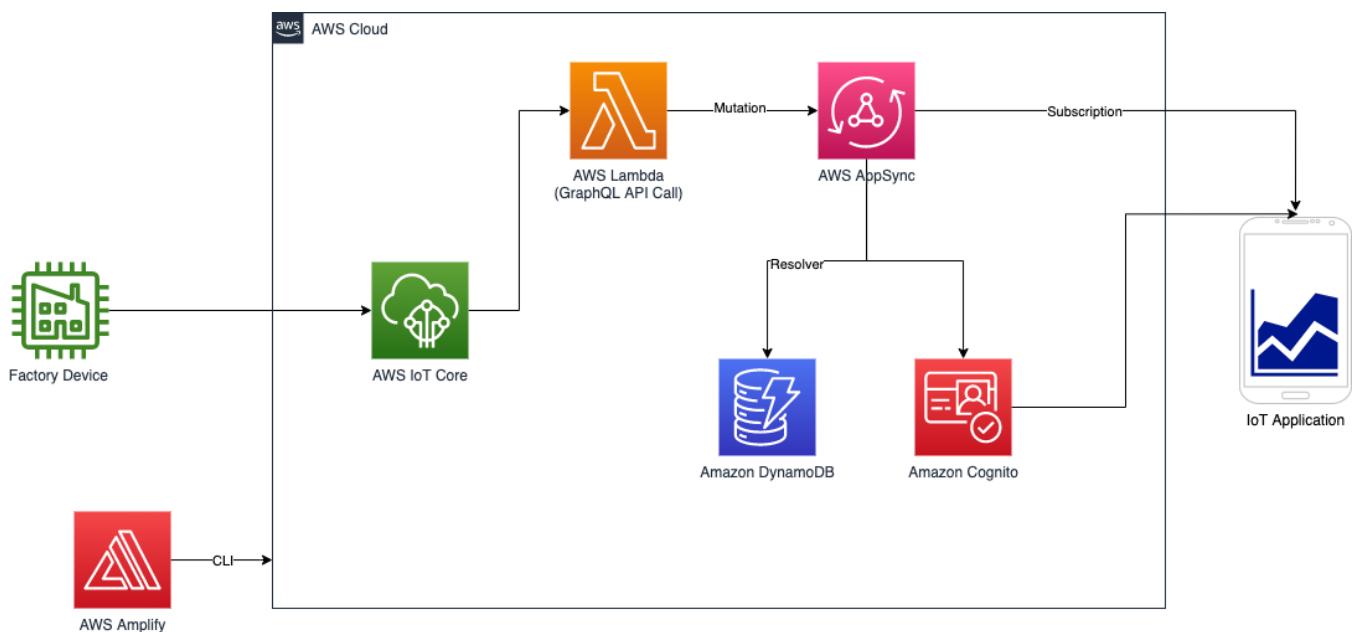
이제 애플리케이션에서 사용할 소스 코드를 다운로드 하겠습니다. /home/ec2-user/environment 위치에서 아래의 명령어를 입력합니다.

```
git clone https://github.com/JinseonyLee/iot-app-hol-online
```

소스 코드는 크게 두 가지 폴더로 구성되어 있습니다.

- **React-amplified** 폴더의 경우, react 로 만들어진 IoT Application 소스 코드와 미리 생성해둔 Amplify 백엔드 소스로 구성되어 있습니다.
- **Sensor-data-simulator** 폴더의 경우, IoT Application 으로 데이터를 1 초에 한번씩 전달하기 위한 스크립트가 작성되어 있습니다.

본격적으로 IoT 애플리케이션을 구축하기 이전 최종 아키텍처에 대해 살펴봅니다.



5. 애플리케이션 만들기 – Amplify 를 이용한 backend

Amplify 를 이용한 backend 를 구축하기 위해 Cloud9 에서 아래와 같이 IoT 애플리케이션 험 디렉토리로 이동합니다.

`cd iot-app-hol-online/react-amplified/`

5-1. Amplify 에서는 다양한 CLI 명령어를 지원합니다. 지원되는 명령어를 확인하기 위해서 Cloud9 터미널 창에 아래의 명령어를 입력합니다.

`amplify`

```
TeamRole:~/environment/iot-app-hol-online/react-amplified (main) $ amplify
amplify <command> <subcommand>

init           Initializes a new project, sets up deployment resources in the cloud, and makes your project ready for Amplify.
configure      Configures the attributes of your project for amplify-cli, such as switching front-end framework and adding/removing cloud-provider plugins.
push          Provisions cloud resources with the latest local developments.
pull          Fetch upstream backend environment definition changes from the cloud and updates the local environment to match that definition.
publish       Executes amplify push, and then builds and publishes client-side application for hosting.
serve         Executes amplify push, and then executes the project's start command to test run the client-side application locally.
status [category> ...]   Shows the state of local resources not yet pushed to the cloud (Create/Update/Delete).
status -v [category> ...] Shows the detailed verbose diff between local and deployed resources, including cloudformation-diff
delete        Deletes all of the resources tied to the project from the cloud.
<category> add    Adds a resource for an Amplify category in your local backend
<category> update Update resource for an Amplify category in your local backend.
<category> push   Provisions all cloud resources in a category with the latest local developments.
<category> remove Removes a resource for an Amplify category in your local backend.
<category> mock   Displays subcommands of the specified Amplify category.
mock          Run mock server for testing categories locally.
codegen       Generates GraphQL statements(queries, mutations and eventHandlers) and type annotations.
env           Displays and manages environment related information for your Amplify project.
console       Opens the web console for the selected cloud resource.
logout        If using temporary cloud provider credentials, this logs out of the account.
upgrade       Download and install the latest version of the Amplify CLI
uninstall     Remove all global Amplify configuration files and uninstall the Amplify CLI. This will not delete any Amplify projects.

where <category> is one of: notifications, api, auth, custom, storage, analytics, function, geo, hosting, interactions, predictions, xr
```

아래와 같이 status 명령어를 입력하면 CLI 로 생성한 Amplify Framework Category 리소스들의 상태를 확인할 수 있습니다.

`amplify status`

```
dev:~/environment/iot-app-hol-online/react-amplified (main) $ amplify status


| Category | Resource name          | Operation | Provider plugin   |
|----------|------------------------|-----------|-------------------|
| Auth     | reactamplified092bb062 | Create    | awscloudformation |
| Api      | reactamplified         | Create    | awscloudformation |
| Function | reactamplifiedaa038e73 | Create    | awscloudformation |


Tag Changes Detected
● No Amplify backend project files detected within this folder.

Resolution:
Either initialize a new Amplify project or pull an existing project.
- "amplify init" to initialize a new Amplify project
- "amplify pull <app-id>" to pull your existing Amplify project. Find the <app-id> in the AWS Console or Amplify Studio.

Learn more at: https://docs.amplify.aws/cli/project/troubleshooting/

Session Identifier: f01fc580-4ae8-4872-90c1-ae6005f97e6a
? An unexpected error has occurred, opt in to send an error report to AWS Amplify with non-sensitive project configuration files. Confirm (y/N) >
```

보시는 바와 같이 현재 **Auth, Api, Function** 이라는 세 가지 카테고리에 해당하는 리소스가 Create 상태로 남아 있습니다. 또한, 에러 메세지를 보면 새로운 프로젝트를 초기화하기 위해선 amplify init이 필요하다고 안내되어 있습니다.

우선 error report를 AWS에 보내겠냐고 물어보는 란에 'N'을 입력하고 빠져 나옵니다.

그리고 초기화 명령을 아래와 같이 수행합니다.

amplify init

입력 값은 아래 화면을 참고하여 넣습니다. 환경 이름은 디폴트로 제공되는 dev 대신 **test**를 입력하고 엔터를 누릅니다.

디폴트 프로바이더를 사용하여 CloudFormation Stack 을 프로비저닝합니다. AWS CloudFormation 콘솔창에서 어떤 리소스들이 배포되는지 확인할 수 있습니다.

Authentication method는 AWS profile로 그대로 두고 엔터를 클릭하고

Profile도 default로 설정하게 엔터를 클릭합니다.

```
dev:~/environment/iot-app-hol-online/react-amplified (main) $ amplify init
Note: It is recommended to run this command from the root of your app directory
? Do you want to use an existing environment? No
? Enter a name for the environment test
Using default provider awscloudformation
? Select the authentication method you want to use: AWS profile

For more information on AWS Profiles, see:
https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html

? Please choose the profile you want to use default
Adding backend environment test to AWS Amplify app: d3n7ugc3kew5h8

Deploying resources into test environment. This will take a few minutes. ...
Deploying root stack reactamplified [-----] 0/4
  amplify-reactamplified-test-1... AWS::CloudFormation::Stack      CREATE_IN_PROGRESS   Fri Jan 06 2023 01:57:19...
  UnauthRole                  AWS::IAM::Role                CREATE_IN_PROGRESS   Fri Jan 06 2023 01:57:23...
  DeploymentBucket            AWS::S3::Bucket              CREATE_IN_PROGRESS   Fri Jan 06 2023 01:57:24...
  AuthRole                   AWS::IAM::Role                CREATE_IN_PROGRESS   Fri Jan 06 2023 01:57:24...
```

초기화 작업이 완료된 다음 다시 status 명령어를 입력하면 이전 화면과 달리 에러 메세지가 사라진 것을 확인할 수 있습니다.

amplify status

```
dev:~/environment/iot-app-hol-online/react-amplified (main) $ amplify status
Current Environment: test



| Category | Resource name          | Operation | Provider plugin   |
|----------|------------------------|-----------|-------------------|
| Auth     | reactamplified092bb062 | Create    | awscloudformation |
| Api      | reactamplified         | Create    | awscloudformation |
| Function | reactamplifiedaa038e73 | Create    | awscloudformation |



GraphQL transformer version: 2

dev:~/environment/iot-app-hol-online/react-amplified (main) $
```

5-2. Auth, Api, Function 리소스를 **amplify push** 명령어를 통해 AWS 서비스로 생성하는 작업을 수행합니다.

amplify push

```
dev:~/environment/iot-app-hol-online/react-amplified (main) $ amplify push
Fetching updates to backend environment: test from the cloud. ✓ GraphQL schema compiled successfully.

Edit your schema at /home/ec2-user/environment/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema.graphql
/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema
✓ Successfully pulled backend environment test from the cloud.
└ Building resource api/reactamplified ✓ GraphQL schema compiled successfully.

Edit your schema at /home/ec2-user/environment/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema.graphql
/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema

Current Environment: test



| Category | Resource name          | Operation | Provider plugin   |
|----------|------------------------|-----------|-------------------|
| Auth     | reactamplified092bb062 | Create    | awscloudformation |
| Api      | reactamplified         | Create    | awscloudformation |
| Function | reactamplifiedaa038e73 | Create    | awscloudformation |



? Are you sure you want to continue? (Y/n) Y
```

"Are you sure you want to continue? (Y/n) [Yes(Y)]" 입력

```
? Are you sure you want to continue? Yes
✓ GraphQL schema compiled successfully.

Edit your schema at /home/ec2-user/environment/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema.graphql
/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema
└ Building resource api/reactamplified ✓ GraphQL schema compiled successfully.

Edit your schema at /home/ec2-user/environment/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema.graphql
/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema
? Do you want to update code for your updated GraphQL API? Yes
? Do you want to generate GraphQL statements (queries, mutations and subscription) based on your schema types?
This will overwrite your current graphql queries, mutations and subscriptions Yes

Deploying resources into test environment. This will take a few minutes. :
Deploying root stack reactamplified [-----] 0/4
    amplify-reactamplified-test-1.. AWS::CloudFormation::Stack      UPDATE_IN_PROGRESS
Deploying auth reactamplified092bb062 [-----] 0/10
Deploying api reactamplified [-----] 0/6
Deploying function reactamplifiedaa038e73 [-----] 0/4

```

작업을 수행하는 동안, 모든 설정은 디폴트 값으로 입력합니다.

"Do you want to update code for your updated GraphQL API:" [Yes(Y)] 로 입력

"Do you want to generate GraphQL statements(queries, mutations and subscription) :" [Yes(Y)] 로 입력

모든 자원을 프로비저닝하는데 **수 분이 소요됩니다**. CloudFormation 콘솔창(<https://ap-northeast-2.console.aws.amazon.com/cloudformation/home?region=ap-northeast-2#/stacks>)

에서 진행 경과를 살펴볼 수 있습니다. 프로비저닝이 완료되면 amplify status 명령어를 통해 상태를 확인합니다.

amplify status

```
dev:~/environment/iot-app-hol-online/react-amplified (main) $ amplify status
Current Environment: test



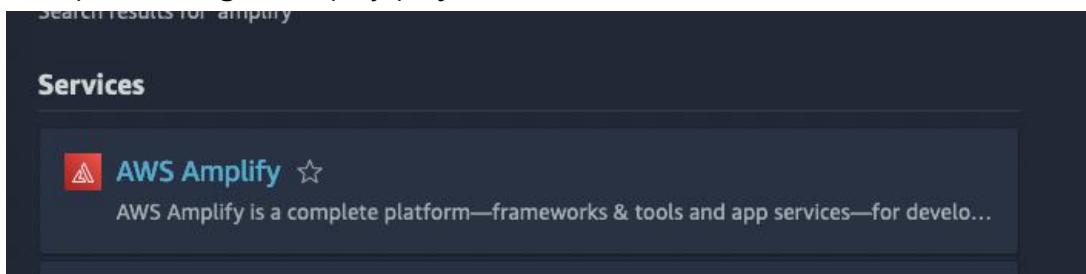
| Category | Resource name          | Operation | Provider plugin   |
|----------|------------------------|-----------|-------------------|
| Api      | reactamplified         | No Change | awscloudformation |
| Auth     | reactamplified092bb062 | No Change | awscloudformation |
| Function | reactamplifiedaa038e73 | No Change | awscloudformation |



GraphQL endpoint: https://h3f3inh15nbirok2yfpkpdbnx.eappsing-api.ap-northeast-2.amazonaws.com/graphql
GraphQL transformer version: 2
dev:~/environment/iot-app-hol-online/react-amplified (main) $
```

5-3. 각 카테고리에 해당하는 리소스가 어떻게 프로비저닝되었는지 AWS 콘솔창에서 확인합니다.

먼저 provisioning 된 amplify project를 확인합니다.



Deploy된 환경이 test environment와 api, authentication, functions가 생성됐음을 확인할 수 있습니다.

The screenshot shows the AWS Amplify app details page for "reactamplified". The left sidebar includes links for "All apps" (selected), "reactamplified", "App settings" (General, Amplify Studio settings), "Documentation", and "Support". The main content area shows the "reactamplified" app homepage, which lists all deployed frontend and backend environments. It highlights the "Backend environments" tab, which displays a "test" environment. The "test" environment status is shown as "Deployment completed 1/4/2023, 1:40:12 PM". Other sections include "Categories added" (API, Authentication, Functions) and a "Set up Amplify Studio" button.

먼저, Auth 리소스를 확인하기 위해, AWS 콘솔창 상단의 검색 창에서 Cognito 를 입력하고

클릭합니다.

Amazon Cognito 인트로 페이지에서 인증을 담당하는 사용자 풀을 확인하기 위해, **Manage User Pools** 버튼을 클릭합니다. 생성되어 있는 유저 풀을 클릭하면 구성 값을 확인할 수 있습니다.

Amazon Cognito > User pools > reactamplified092bb062_userpool_092bb062-test

User pool name	ARN	Created time
reactamplified092bb062_userpool_092bb062-test	arn:aws:cognito-idp:ap-northeast-2:986221661979:userpool/ap-northeast-2_FccuoM9FT	January 4, 2023 at 13:41 GMT+9
User pool ID	Estimated number of users	Last updated time
ap-northeast-2_FccuoM9FT	1	January 4, 2023 at 13:41 GMT+9

두 번째, API 리소스를 확인하기 위해, AWS 콘솔창 상단의 검색 창에서 **AppSync** 및 **DynamoDB** 를 검색합니다.

AWS AppSync 화면을 보면 **reactamplified-test**라는 이름의 API 가 생성된 것을 확인할 수 있습니다. 해당 박스를 클릭하여 Schema, Queries 와 같은 기능을 확인해봅니다.

```

1+ input CreateSensorInput {
2   id: ID!
3   sensorType: String!
4   value: Float!
5   isWarning: Boolean!
6   timestamp: AWSTimestamp!
7 }
8
9 input DeleteSensorInput {
10   sensorType: String!
11   timestamp: AWSTimestamp!
12 }
13
14 enum ModelAttributeTypes {
15   binary
16   binarySet
17   bool
18   list
19   map
20   number
21   numberSet
22   string
23   stringSet
24   _null
25 }
26
27 input ModelBooleanInput {
28   ne: Boolean
29   eq: Boolean
30   attributeExists: Boolean
31   attributeType: ModelAttributeTypes
32 }
33
34 input ModelFloatInput {
35   ne: Float
36   eq: Float
37   le: Float
38   lt: Float
39   ge: Float
40   gt: Float
41   between: [Float]
42   attributeExists: Boolean
43   attributeType: ModelAttributeTypes
44 }
45
46 input ModelIDInput {
47   ...
}

```

그리고 AppSync 의 데이터 소스로 사용된 DynamoDB 도 살펴봅니다. DynamoDB 콘솔창에서 사이드바에 **Tables** 를 클릭하면 정보를 확인할 수 있습니다. Partition key, Sort key 와 같은 속성을 살펴봅니다.

AWS IoT (Amplify, AppSync, Cognito) Application Hands-on

The screenshot shows two separate AWS service pages. On the left, the 'Data Sources' section of the AWS AppSync console is displayed, showing a table of existing data sources. One entry is visible: 'SensorTable' of type 'AMAZON_DYNAMODB' connected to the resource 'Sensor-52tzlil7cfenffiw6xmudg6ixa-test'. On the right, the 'Sensor-52tzlil7cfenffiw6xmudg6ixa-test' table details page in the AWS DynamoDB console is shown. The table has one item, 'Sensor-52tzlil7cfenffiw6xmudg6ixa-test', with a partition key 'sensorType' set to 'String' and a sort key 'timestamp' set to 'Number'. The table is in 'On-demand' capacity mode and is active with no alarms.

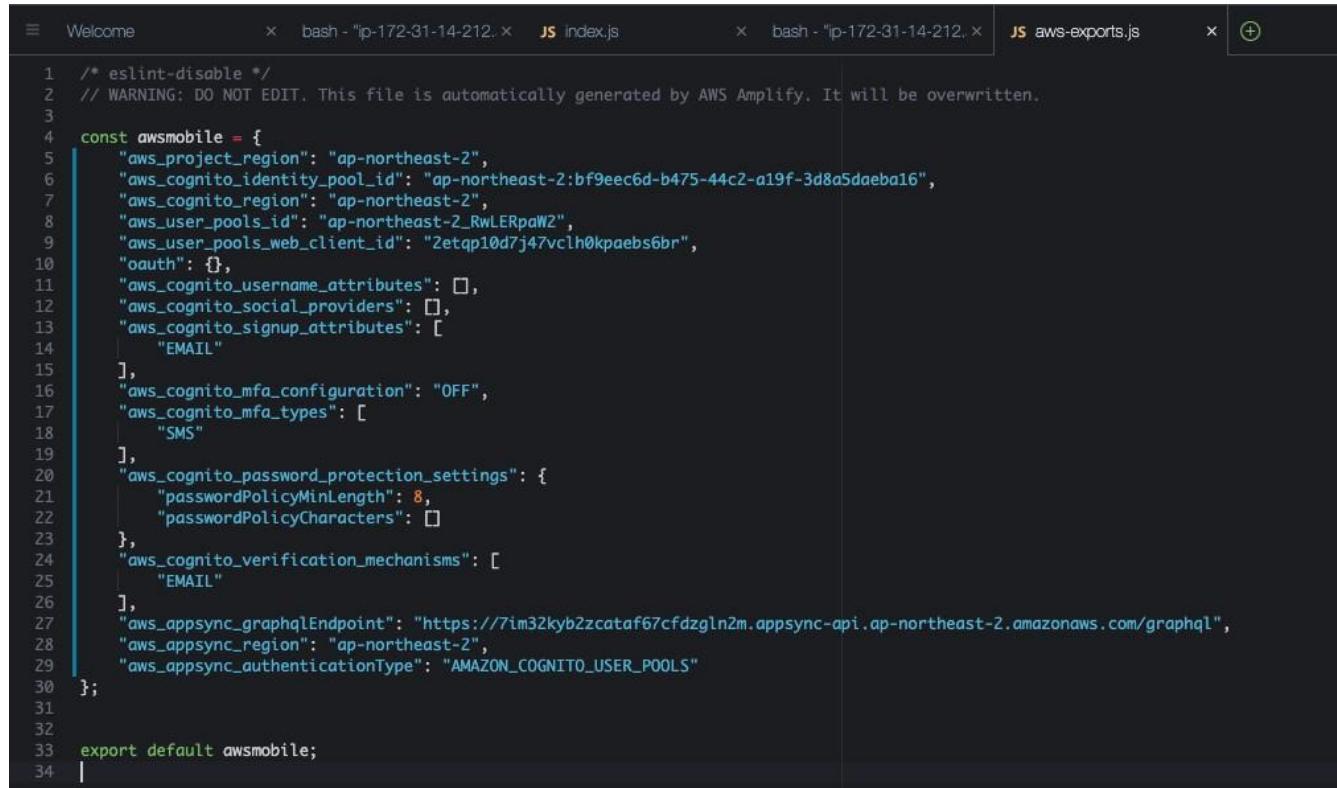
마지막으로 **Function 리소스**를 확인하기 위해, AWS 콘솔창 상단의 검색 창에서 **Lambda**를 검색합니다.

Lambda Functions 메뉴에서 이름의 마지막에 **-test**로 끝나는 함수를 클릭합니다. 해당 함수의 자세한 코드는 Cloud9 IDE 환경에서 /home/ec2-user/environment/iot-app-hol-online/react-amplified/amplify/backend 위치에 function 폴더 안에 기록되어 있습니다. 아래 화면과 같이 Lambda 콘솔창에서 환경 변수 및 관련 정보를 확인할 수 있습니다.

Environment Variable	Value
API_REACTAMPLIFIED_GRAPHQLAPIENDPOINTOUTPUT	https://h3f3inhl5nbirok2yfpkpdbnx.eappsing-api.ap-northeast-2.amazonaws.com/graphql
API_REACTAMPLIFIED_GRAPHQLAPIIDOUTPUT	S2tzll7cfenffw6xmudg6ixa
ENV	test
REGION	ap-northeast-2

Cloud9 IDE 환경에서 /home/ec2-user/environment/iot-app-hol-online/react-amplified/src 위치에 있는 파일

중, **aws-exports.js** 를 확인해보면 backend 에 대한 meta data 정보를 확인할 수 있습니다.



```
1 /* eslint-disable */
2 // WARNING: DO NOT EDIT. This file is automatically generated by AWS Amplify. It will be overwritten.
3
4 const awsmobile = {
5   "aws_project_region": "ap-northeast-2",
6   "aws_cognito_identity_pool_id": "ap-northeast-2:bf9eec6d-b475-44c2-a19f-3d8a5daeba16",
7   "aws_cognito_region": "ap-northeast-2",
8   "aws_user_pools_id": "ap-northeast-2_RwLERpaWZ",
9   "aws_user_pools_web_client_id": "2etqp10d7j47vclh0kpaebs6br",
10  "oauth": {},
11  "aws_cognito_username_attributes": [],
12  "aws_cognito_social_providers": [],
13  "aws_cognito_signup_attributes": [
14    "EMAIL"
15  ],
16  "aws_cognito_mfa_configuration": "OFF",
17  "aws_cognito_mfa_types": [
18    "SMS"
19  ],
20  "aws_cognito_password_protection_settings": {
21    "passwordPolicyMinLength": 8,
22    "passwordPolicyCharacters": []
23  },
24  "aws_cognito_verification_mechanisms": [
25    "EMAIL"
26  ],
27  "aws_appsync_graphqlEndpoint": "https://7im32kybzczataf67cfdzgln2m.appsync-api.ap-northeast-2.amazonaws.com/graphql",
28  "aws_appsync_region": "ap-northeast-2",
29  "aws_appsync_authenticationType": "AMAZON_COGNITO_USER_POOLS"
30 };
31
32
33 export default awsmobile;
34 |
```

6. 애플리케이션 만들기 – React 를 이용한 frontend

React 를 이용한 frontend 를 구축하기 위해 Cloud9 에서 아래와 같이 IoT 애플리케이션 훙 디렉토리로 이동합니다.

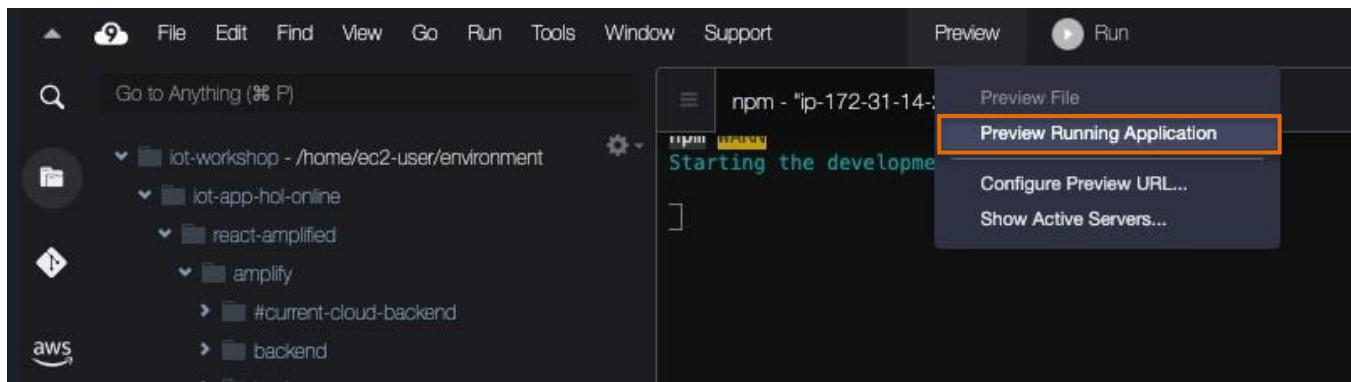
`cd iot-app-hol-online/react-amplified/`

React 애플리케이션 코드는 이미 다운 받은 폴더에 저장되어 있습니다. 애플리케이션을 실행하기 위해 아래의 명령어를 입력합니다.

`npm install npm`

`npm start`

잠시 후, 로컬 환경에서 서버가 로딩되면 상단 Preview 메뉴에서 **Preview Running Application** 을 선택합니다.



아래와 같이 로그인 화면이 나오는지 확인합니다. **Create Account** 탭을 클릭 후, 회원 가입을 진행합니다.

The screenshot displays two windows from the AWS Cloud9 IDE. On the left, a terminal window titled 'npm - "ip-172-31-14-212.eu...' shows the output of an npm build command. It includes local and network URLs, asset details, and a summary of the build process. On the right, a browser window titled '[B] https://01ed39b819b6...' shows a simple sign-in form with fields for 'Username' and 'Password', and a 'Sign in' button. There are also 'Create Account' and 'Forgot your password?' links.

```

npm - "ip-172-31-14-212.eu...
You can now view react-amplified in the browser.

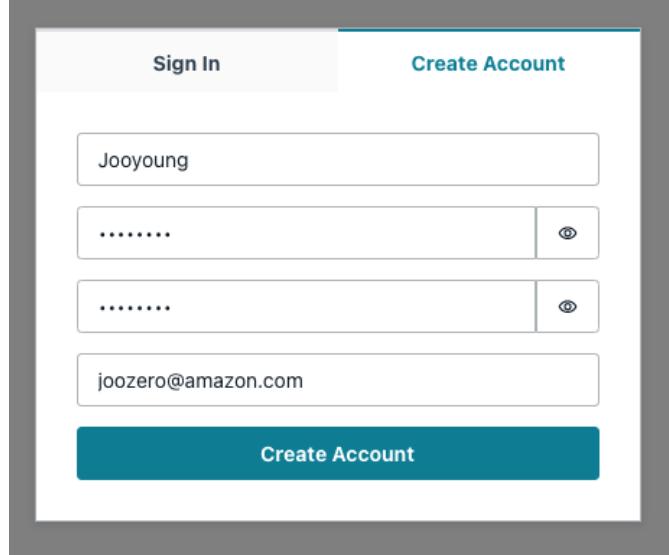
Local:          http://localhost:8080
On Your Network:  http://172.31.14.212:8080

Note that the development build is not optimized.
To create a production build, use npm run build.

asset static/js/bundle.js 13.4 MiB [emitted] (name: main) 1 related asset
asset index.html 1.8 KiB [emitted]
asset asset-manifest.json 190 bytes [emitted]
orphan modules 7.12 MiB [orphan] 2649 modules
runtime modules 29.7 KiB 15 modules
cacheable modules 10.4 MiB
  javascript modules 10.4 MiB
    modules by path ./node_modules/ 10.4 MiB 2166 modules
    modules by path ./src/ 20.9 KiB 8 modules
      ./util.inspect (ignored) 15 bytes [built] [code generated]
      crypto (ignored) 15 bytes [optional] [built] [code generated]
  json modules 26.2 KiB
    ./node_modules/@aws-sdk/client-cognito-identity/dist/es/package.json 2.
  78 KiB [built] [code generated]
    ./node_modules/@aws-sdk/client-pinpoint/dist/es/package.json 2.56 KiB [
  built] [code generated]
      ./node_modules/@aws-sdk/client-kinesis/dist/es/package.json 2.72 KiB [b
  uilt] [code generated]
      ./node_modules/@aws-sdk/client-firehose/dist/es/package.json 2.56 KiB [
  built] [code generated]
      ./node_modules/@aws-sdk/client-personalize-events/dist/es/package.json
  2.6 KiB [built] [code generated]
      + 5 modules
webpack 5.71.0 compiled successfully in 49971 ms

```

메일을 통해, 검증하는 과정이 있기에 반드시 유효한 메일 주소를 입력합니다.



입력한 메일 주소로 온 6 자리 인증 코드를 입력한 후, **Confirm** 버튼을 클릭하면 아래와 같은 화면을 확인할 수 있습니다. 아직 모니터링할 데이터가 수신되지 않았기에 아무런 정보를 확인할 수 없습니다.

```
You can now view react-amplified in the browser.

Local:          http://localhost:8080
On Your Network:  http://172.31.14.212:8080

Note that the development build is not optimized.
To create a production build, use npm run build.

asset static/js/bundle.js 13.4 MiB [emitted] (name: main) 1 related asset
asset index.html 1.8 KiB [emitted]
asset asset-manifest.json 190 bytes [emitted]
orphan modules 7.12 MiB [orphan] 2649 modules
runtime modules 29.7 KiB 15 modules
cacheable modules 10.4 MiB
javascript modules 10.4 MiB
  modules by path ./node_modules/ 10.4 MiB 2166 modules
  modules by path ./src/ 20.9 KiB 8 modules
    ./util.inspect (ignored) 15 bytes [built] [code generated]
    crypto (ignored) 15 bytes [optional] [built] [code generated]
  json modules 26.2 KiB
    ./node_modules/@aws-sdk/client-cognito-identity/dist/es/package.json 2.78
    KiB [built] [code generated]
    ./node_modules/@aws-sdk/client-pinpoint/dist/es/package.json 2.56 KiB [bu
    ilt] [code generated]
    ./node_modules/@aws-sdk/client-kinesis/dist/es/package.json 2.72 KiB [bu
    ilt] [code generated]
    ./node_modules/@aws-sdk/client-firehose/dist/es/package.json 2.56 KiB [bu
    ilt] [code generated]
    ./node_modules/@aws-sdk/client-personalize-events/dist/es/package.json 2.
    6 KiB [built] [code generated]
      + 5 modules
webpack 5.71.0 compiled successfully in 49971 ms
```

IoT Dashboard

공장 1 라인

차트를 클릭하면 시계열 그래프를 보실 수 있습니다.

wRMSCurrent

Range	Value
0 - 30	~30
30 - 60	~30
60 - 90	~30
90 - 120	~30
120 - 150	~60

wCurrentLoad

Range	Value
0 - 30	~30
30 - 60	~30
60 - 90	~30
90 - 120	~30
120 - 150	~60

7. IoT 환경 구성하기 – AWS IoT Core

AWS 콘솔 상단 검색 창에 IoT Core 를 입력하고 **AWS IoT Core (Click)** 서비스로 이동합니다.

인트로 페이지에서 **Connect device** 버튼을 클릭합니다.

다음 화면에서 디바이스를 AWS IoT 로 연결하는 순서에 대해 안내합니다. 큰 흐름은 세 가지입니다. 디바이스를 등록하고, 연결 키트를 다운로드한 다음, 디바이스에 구성하고 AWS IoT Core 로 정상적으로 송신이 되는지 테스트합니다. 해당 절차를 확인한 하고, 디바이스 측면에서 준비해야할 내용들을 살펴 봅니다(아래의 내용 참조).

1. AWS IoT Core 로 데이터를 송신하기 위해 디바이스가 인터넷에 연결되어 있는지 확인합니다.
2. Connection Kit를 디바이스에게 로드하는 방법을 확인합니다.
3. CLI 에 접근 가능한지 확인합니다.
4. 터미널 창에서 아래의 명령어가 실행되는지 확인합니다.

Step 1 Prepare your device

Step 2 Register and secure your device

Step 3 Choose platform and SDK

Step 4 Download connection kit

Step 5 Run connection kit

How it works

In this wizard, we'll be creating a thing resource in AWS IoT. A thing resource is a digital representation of a physical device or logical entity.

A thing resource uses certificates to secure communication between your device and AWS IoT. AWS IoT policies control access to the AWS IoT resources. This wizard creates the certificate and policy for your device.

When a device connects to AWS IoT, policies enable it to subscribe and publish MQTT messages with AWS IoT message broker.

Prepare your device

- Turn on your device and make sure it's connected to the internet.
- Choose how you want to load files onto your device.
 - If your device supports a browser, open the AWS IoT console on your device and run this wizard. You can download the files directly to your device from the browser.
 - If your device doesn't support a browser, choose the best way to transfer files from the computer with the browser to your device. Some options to transfer files include using the file transfer protocol (FTP) and using a USB memory stick.
- Make sure that you can access a command-line interface on your device.
 - If you're running this wizard on your IoT device, open a terminal window on your device to access a command-line interface.
 - If you're not running this on your IoT device, open an SSH terminal window on this device and connect it to your IoT device.
- From the terminal window, enter this command:

```
ping a13vpn9lg96blm-ats.iot.ap-northeast-2.amazonaws.com
```

After you complete these steps and get a successful ping response, you're ready to continue and connect your device to AWS IoT.

Cancel **Next** Next

사물(thing)의 이름을 지정하고 등록합니다. 본 실습에서는 **MyThing** 으로 이름을 입력합니다.

Thing properties

Create a new thing Choose an existing thing

Thing name

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations
You can use these configurations to add detail that can help you to organize, manage, and search your things.

- ▶ **Thing type** - *optional*
- ▶ **Searchable thing attributes** - *optional*
- ▶ **Thing groups** - *optional*
- ▶ **Billing group** - *optional*

사물에 적용할 OS 플랫폼 및 AWS IoT Device SDK 를 선택합니다. 본 실습에서는 **Linux/macOS** 와 **Python** 을 선택합니다. 그 다음 **Next** 버튼을 클릭합니다. 참고로 사물은 Python 과 Git 이 설치되어 있어야 합니다.

Platform and SDK

Choose the platform OS and AWS IoT Device SDK that you want to use for your device.

Device platform operating system

This is the operating system installed on the device that will connect to AWS.

Linux / macOS
Linux version: any
macOS version: 10.13+

Windows
Version 10

AWS IoT Device SDK

Choose a Device SDK that's in a language your device supports.

Node.js
Version 10+
Requires Node.js and npm to be installed

Python
Version 3.6+
Requires Python and Git to be installed

Java
Version 8
Requires Java JDK, Maven, and Git to be installed

Cancel **Previous** **Next**

연결 키트의 설정 값을 확인한 다음 **연결 키트**를 다운로드합니다.

Connection kit

Certificate

MyThing.cert.pem

Private key

MyThing.private.key

AWS IoT Device SDK

Python

Script to send and receive

messages

start.sh

Policy

MyThing-Policy

[View policy](#)

Download

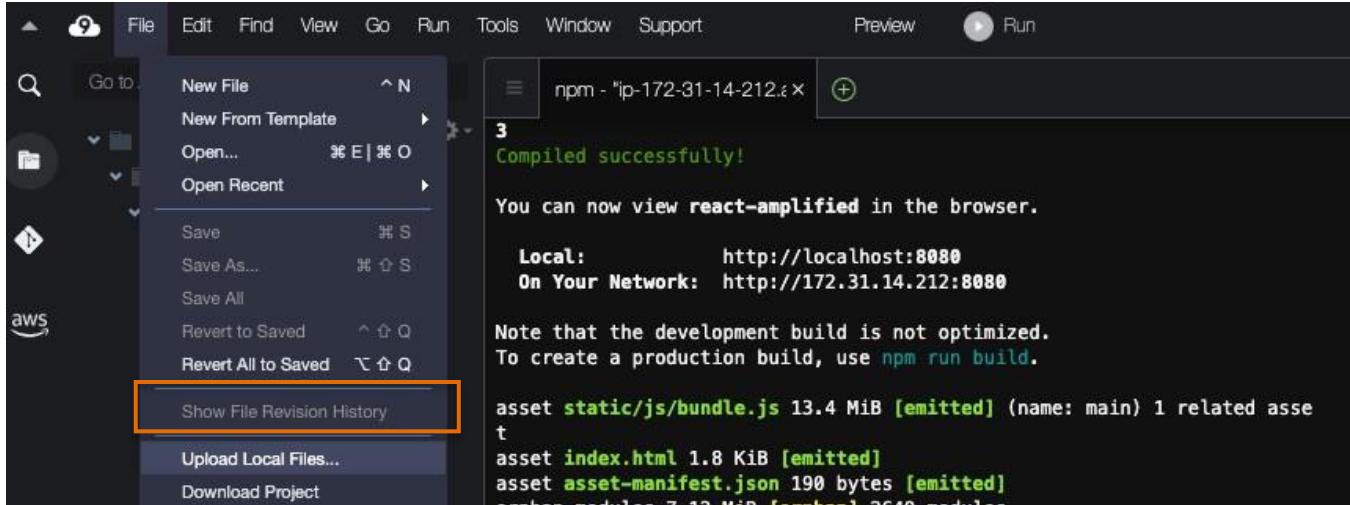
If you are running this from a browser on the device, after you download the connection kit, it will be in the browser's download folder.

If you are not running this from a browser on your device, you'll need to transfer the connection kit from your browser's download folder to your device using the method you tested when you prepared your device in step 1.

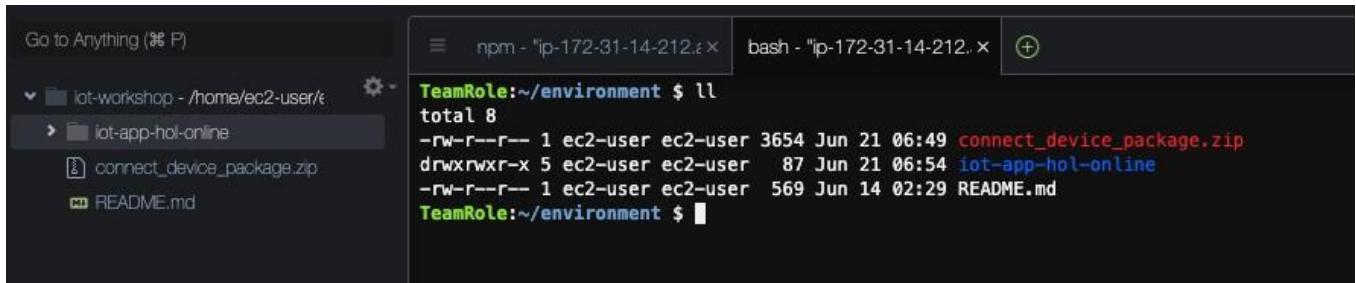
 [Download connection kit](#)

다운받은 파일의 경우, 실제 thing 디바이스에 옮겨야 합니다. 해당 파일에는 인증서 및 private key 등 디바이스에서 사용할 주요 정보들을 포함하고 있습니다. 본 실습에서는 Cloud9 을 thing 디바이스로 사용할 예정입니다.

Cloud9 IDE 환경으로 다시 이동합니다. **다운 받은 connection kit** 를 Cloud9 환경에 업로드합니다. 좌측 상단에 **File > Upload Local Files** 를 클릭합니다. Drag & drop 알람창이 나오면 connection kit 를 업로드합니다.



이때, Cloud9 왼쪽 사이드 바에서 특정 파일을 조회했었다면 해당 위치에 zip 파일이 업로드될 것입니다. 만약 그럴 경우, 홈 디렉토리로 해당 파일을 끌어 올려줍니다. 그리고 새로운 터미널 창을 엽니다. 파일 리스트를 조회하면 아래와 같은 결과를 확인할 수 있습니다.



Connection kit 압축을 해제하고 쉘스크립트를 구동하는 일련의 작업을 아래의 명령어들로 수행합니다.

```

unzip connect_device_package.zip
chmod +x start.sh
sudo ./start.sh

```

start.sh 스크립트에는 **AWS IoT SDK** 를 설치하는 과정이 포함되어 있습니다. IoT Thing Device에서 AWS IoT SDK를 사용하면 매우 편리하게 AWS IoT Core와 통신할 수 있는 라이브러리를 지원받을 수 있습니다. 이때, connection kit에 포함된 인증서 및 private key를 이용하여 토픽을 발행합니다. 웰 스크립트를 실행시키면 화면과 같이 **sdk/test/python** 이라는 토픽으로 MQTT 메세지를 퍼블리시하는 것을 확인할 수 있습니다.

```
dev:~/environment $ sudo ./start.sh

Downloading AWS IoT Root CA certificate from AWS...
% Total    % Received   % Xferd  Average Speed   Time     Time   Current
          Dload  Upload   Total Spent  Left Speed
100  1188  100  1188    0     0  46856      0 --:--:-- --:--:-- --:--:-- 47520

Cloning the AWS SDK...
Cloning into 'aws-iot-device-sdk-python-v2'...
remote: Enumerating objects: 1706, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (88/88), done.
remote: Total 1706 (delta 39), reused 65 (delta 22), pack-reused 1595
Receiving objects: 100% (1706/1706), 1.91 MiB | 17.93 MiB/s, done.
Resolving deltas: 100% (1004/1004), done.

Installing AWS SDK...
WARNING: Running pip install with root privileges is generally not a good idea. Try `python3 -m pip install --user` instead.
Processing ./aws-iot-device-sdk-python-v2
Collecting awscrt==0.16.0
  Downloading awscrt-0.16.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.9 MB)
    [██████████] 7.9 MB 24.3 MB/s
Using legacy 'setup.py install' for awsiotsdk, since package 'wheel' is not installed.
Installing collected packages: awscrt, awsiotsdk
  Running setup.py install for awsiotsdk ... done
Successfully installed awscrt-0.16.0 awsiotsdk-1.0.0.dev0

Running pub/sub sample application...
Connecting to a13vpn9lg96blm-ats.iot.ap-northeast-2.amazonaws.com with client ID 'basicPubSub'...
Connected!
Subscribing to topic 'sdk/test/python'...
Subscribed with QoS.AT LEAST_ONCE
Sending messages until program killed
Publishing message to topic 'sdk/test/python': Hello World! [1]
Received message from topic 'sdk/test/python': b'"Hello World! [1]"'
Publishing message to topic 'sdk/test/python': Hello World! [2]
Received message from topic 'sdk/test/python': b'"Hello World! [2]"'
Publishing message to topic 'sdk/test/python': Hello World! [3]
Received message from topic 'sdk/test/python': b'"Hello World! [3]"'
Publishing message to topic 'sdk/test/python': Hello World! [4]
Received message from topic 'sdk/test/python': b'"Hello World! [4]"'
Publishing message to topic 'sdk/test/python': Hello World! [5]
Received message from topic 'sdk/test/python': b'"Hello World! [5]"'
Publishing message to topic 'sdk/test/python': Hello World! [6]
Received message from topic 'sdk/test/python': b'"Hello World! [6]"'
Publishing message to topic 'sdk/test/python': Hello World! [7]
Received message from topic 'sdk/test/python': b'"Hello World! [7]"'
Publishing message to topic 'sdk/test/python': Hello World! [8]
Received message from topic 'sdk/test/python': b'"Hello World! [8]"'
Publishing message to topic 'sdk/test/python': Hello World! [9]
Received message from topic 'sdk/test/python': b'"Hello World! [9]"'
Publishing message to topic 'sdk/test/python': Hello World! [10]
Received message from topic 'sdk/test/python': b'"Hello World! [10]"'
```

위와 같이 퍼블리시되는 내용을 AWS IoT 콘솔창에서 실시간으로 확인할 수 있습니다.

Step 3: Return to this screen to view your device's messages

After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.

The screenshot shows the AWS IoT Subscriptions interface. On the left, there is a sidebar with a single item: 'sdk/test/Python'. The main area is titled 'sdk/test/Python' and contains two messages. Each message is timestamped 'August 06, 2022, 21:06:11 (UTC+0900)' and has a collapse icon (▼). The first message is a JSON object:

```
{  
  "message": "Hello World!",  
  "sequence": 15  
}
```

The second message is also a JSON object:

```
{  
  "message": "Hello World!",  
  "sequence": 14  
}
```

At the top right of the main area, there are two buttons: 'Pause' and 'Clear'.

다음 버튼을 클릭 후, 설정 값을 확인합니다.

또한 Cloud9 환경에서도 메세지가 정상적으로 수신되는 것을 확인했다면 Cloud9 IDE에서 쉘 스크립트를 실행시킨 터미널 창에서 **Ctrl + C**를 입력하여 **start.sh**를 종료시킵니다.

그 다음 AWS IoT Core 페이지 왼쪽 사이드바에서 **Manage > Security > Policies** 를 클릭합니다.

MyThing-Policy 하이퍼 링크를 클릭합니다.

The screenshot shows the AWS IoT Policies page. At the top, there are three buttons: a grey 'Create' button, a white 'Delete' button, and an orange 'Create policy' button. Below these are search and filter fields. The main list contains one item: 'MyThing-Policy'. To the right of the list is a pagination control showing page 1 of 1, and a settings gear icon.

Policy name
MyThing-Policy

아래의 페이지에서 오른쪽 상단에 **Edit active version** 버튼을 클릭합니다.

The screenshot shows the MyThing-Policy details page. At the top right, there are two buttons: an orange 'Edit active version' button and a white 'Delete' button. Below this is a 'Details' section with four columns: Policy ARN (redacted), Active version (1), Created (June 21, 2022, 14:59:11 UTC+0900), and Last updated (June 21, 2022, 14:59:11 UTC+0900). Below the details are tabs for Versions, Targets, Noncompliance, and Tags. The 'Versions' tab is selected. Under 'Active version: 1' (Info), there are three rows of policy statements:

Policy effect	Policy action	Policy resource
Allow	iot:Publish	arn:aws:iot:ap-northeast-2:617073285002:topic/sdk/test/java
Allow	iot:Publish	arn:aws:iot:ap-northeast-2:617073285002:topic/sdk/test/Python

At the top right of the 'Active version: 1' section are 'Builder' and 'JSON' buttons.

Policy statements 탭에서 Policy document 아래 **JSON 버튼**을 클릭합니다. AWS IoT Policy에는 하나 이상의 정책이 포함되어 있습니다. 각 statement에는 작업을 허용하거나 거부하는 내용, 어떤 리소스에 대해 해당 action을 맵핑할 것인지 등에 대한 정보가 기재되어 있습니다.

The screenshot shows the AWS IoT Policy Editor interface. At the top, there's a breadcrumb navigation: AWS IoT > Security > Policies > MyThing-Policy > Edit. Below that, the title is "Edit policy: MyThing-Policy (Version 1)". There are two tabs at the top: "Policy statements" (which is orange, indicating it's selected) and "Policy examples". Under the "Policy document" section, there are two buttons: "Builder" and "JSON", with "JSON" being highlighted by a red box. A small note below says: "An AWS IoT policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources."

기존 정책을 모두 지우고 아래와 같이 입력한 후, 새버전으로 저장합니다. 아래에 기재되어 있는 policy의 경우, **IoT 관련 action을 모두 허용**한다는 의미입니다.

실습의 편리성을 위해, 아래와 같이 막강한 권한을 부여하였으나 실제 production level에서 서비스를 사용할 경우, 필요한 권한만 부여하는 것이 맞습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow", "Action": [
        "iot:*"
      ],
      "Resource": "*"
    }
  ]
}
```

아래의 화면처럼 policy document 를 작성한 다음, **policy version status**에 체크 표시를 합니다.

체크 표시를 해야만 변경사항이 반영됩니다.

마지막으로 **Save as new version** 버튼을 누릅니다.

The screenshot shows the AWS IoT Policy Editor interface. At the top, the navigation bar includes 'AWS IoT > Security > Policies > MyThing-Policy > Edit'. Below the navigation is the title 'Edit policy: MyThing-Policy (Version 1)'. There are two tabs: 'Policy statements' (selected) and 'Policy examples'. Under 'Policy document', there are two buttons: 'Builder' (selected) and 'JSON'. The JSON code is displayed in a code editor with line numbers 1 through 13. Lines 1 through 12 are highlighted with an orange box. Line 13 is the closing brace '}' of the policy object. The code is as follows:

```

1▼ {
2  "Version": "2012-10-17",
3▼   "Statement": [
4▼     {
5       "Effect": "Allow",
6▼       "Action": [
7         "iot:/*"
8       ],
9       "Resource": "*"
10      }
11    ]
12  }
13

```

Below the code editor, it says 'JSON Line 13, Column 1 Errors: 0 Warnings: 0'. In the 'Policy version status' section, there is a box labeled 'Active policy' containing the checked checkbox 'Set the edited version as the active version for this policy'. A note below says 'You can change this setting later in the policy's detail page.' At the bottom right are 'Cancel' and 'Save as new version' buttons, with 'Save as new version' being orange.

8. 토픽 시뮬레이션

이제 우리는 Hello World!라는 메세지 대신 실제 IoT Application에서 관찰할 데이터를 생성하여 전달하는 작업을 진행합니다. start.sh 를 수정하여 미리 받아 놓은 **sensor-data-simulator 폴더의 스크립트를 실행시키는** 작업을 수행하겠습니다.

* Device Connection Kit이 2022년9월 기준으로 IOTSDK V2 버전만 다운로드 받게 변경되어 시뮬레이터는 IOTSDK1, IOTSDK2 환경으로 각각 호환되는 파이썬 소스를 지원하고 있습니다.

Ubuntu chaged generate_data source code - refactoring		
		435612a 20 hours ago History
..		
<input type="checkbox"/> AmazonRootCA1.pem	first	2 years ago
<input type="checkbox"/> README.md	Committed after changing source code due to IoT default SDK version c...	yesterday
<input type="checkbox"/> command_line_utils.py	Committed after changing source code due to IoT default SDK version c...	yesterday
<input type="checkbox"/> config.json	first	2 years ago
<input type="checkbox"/> config_test.json	first	2 years ago
<input type="checkbox"/> data_simulator.py	chaged generate_data source code	20 hours ago
<input type="checkbox"/> pubsub_simulator_inference.py	change load_config value	6 months ago
<input type="checkbox"/> pubsub_simulator_inference_v2.py	chaged generate_data source code - refactoring	20 hours ago
<input type="checkbox"/> pubsub_simulator_training.py	first	2 years ago
<input type="checkbox"/> pubsub_simulator_training_v2.py	Committed after changing source code due to IoT default SDK version c...	yesterday

Cloud9 IDE 환경에서 왼쪽 사이드바에서 start.sh파일을 더블 클릭합니다. **/home/ec2-user/environment/start.sh 파일에서 맨 아래(36 번째 줄)로 이동합니다**

기존 start.sh의 --endpoint 앞의 python 실행파일과 --topic 다음의 토픽 이름을 변경합니다.

iot-app-hol-online/sensor-data-simulator/pubsub_simulator_inference_v2.py

iot/sensors

```
python3 aws-iot-device-sdk-python-v2/samples/pubsub.py --endpoint a13vpn9lg96blm-ats.iot.ap-northeast-2.amazonaws.com --ca_file root-CA.crt --cert MyThing.cert.pem --key MyThing.private.key --client_id basicPubSub --topic sdk/test/python --count 0
```

```
python3 iot-app-hol-online/sensor-data-simulator/pubsub_simulator_inference_v2.py --endpoint a13vpn9lg96blm-ats.iot.ap-northeast-2.amazonaws.com --ca_file root-CA.crt --cert MyThing.cert.pem --key MyThing.private.key --client_id basicPubSub --topic iot/sensors --count 0
```

```

34 # run pub/sub sample app using certificates downloaded in package
35 printf "\nRunning pub/sub sample application...\n"
36 python3 iot-app-hol-online/sensor-data-simulator/pubsub_simulator_inference_v2.py --topic iot/sensors --endpoint a13vpn9lg96blm-ats.iot.ap-northeast-2.amazonaws.com --ca_file root-CA.crt

```

그 다음 Ctrl + S 를 눌러 파일을 저장합니다. start.sh 파일 탭 옆에 녹색 동그라미가 사라져야 변경 사항이 정상적으로 저장된 것입니다. 그 다음 새로운 터미널 창을 열고

`sudo ./start.sh`

명령어를 다시 실행시켰을 때, MQTT 메세지가 정상적으로 수신되는 것을 확인합니다.

```

dev:~/environment $ sudo ./start.sh
Running pub/sub sample application...
Connecting to a13vpn9lg96blm-ats.iot.ap-northeast-2.amazonaws.com with client ID 'basicPubSub'...
Connected!
Published topic iot/sensors: {"rVibration_Temp": 47.24, "rVibration_Z_RMS_Velocity": 41.57, "rVibration_X_RMS_Velocity": 100.73, "wRMSCurrent": 102.13, "wCurrentLoad": 65.59, "wEncoderVelocity": 16.99, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:27.955", "timeInSeconds": 1672973847}

Published topic iot/sensors: {"rVibration_Temp": 46.5, "rVibration_Z_RMS_Velocity": 41.87, "rVibration_X_RMS_Velocity": 101.81, "wRMSCurrent": 103.22, "wCurrentLoad": 64.62, "wEncoderVelocity": 17.82, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:28.958", "timeInSeconds": 1672973848}

Published topic iot/sensors: {"rVibration_Temp": 44.94, "rVibration_Z_RMS_Velocity": 39.11, "rVibration_X_RMS_Velocity": 101.32, "wRMSCurrent": 103.74, "wCurrentLoad": 63.94, "wEncoderVelocity": 15.84, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:29.961", "timeInSeconds": 1672973849}

Published topic iot/sensors: {"rVibration_Temp": 45.35, "rVibration_Z_RMS_Velocity": 40.55, "rVibration_X_RMS_Velocity": 102.31, "wRMSCurrent": 102.61, "wCurrentLoad": 63.62, "wEncoderVelocity": 16.02, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:30.964", "timeInSeconds": 1672973850}

Published topic iot/sensors: {"rVibration_Temp": 45.17, "rVibration_Z_RMS_Velocity": 41.05, "rVibration_X_RMS_Velocity": 103.97, "wRMSCurrent": 103.51, "wCurrentLoad": 63.64, "wEncoderVelocity": 16.57, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:31.968", "timeInSeconds": 1672973851}

Published topic iot/sensors: {"rVibration_Temp": 46.91, "rVibration_Z_RMS_Velocity": 40.93, "rVibration_X_RMS_Velocity": 101.91, "wRMSCurrent": 102.84, "wCurrentLoad": 65.02, "wEncoderVelocity": 18.54, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:32.971", "timeInSeconds": 1672973852}

Published topic iot/sensors: {"rVibration_Temp": 47.23, "rVibration_Z_RMS_Velocity": 39.31, "rVibration_X_RMS_Velocity": 102.85, "wRMSCurrent": 104.44, "wCurrentLoad": 63.83, "wEncoderVelocity": 18.56, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:33.974", "timeInSeconds": 1672973853}

Published topic iot/sensors: {"rVibration_Temp": 46.23, "rVibration_Z_RMS_Velocity": 42.37, "rVibration_X_RMS_Velocity": 101.09, "wRMSCurrent": 103.31, "wCurrentLoad": 65.39, "wEncoderVelocity": 18.17, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:34.977", "timeInSeconds": 1672973854}

Published topic iot/sensors: {"rVibration_Temp": 45.7, "rVibration_Z_RMS_Velocity": 40.68, "rVibration_X_RMS_Velocity": 101.47, "wRMSCurrent": 103.14, "wCurrentLoad": 64.41, "wEncoderVelocity": 18.35, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:35.981", "timeInSeconds": 1672973855}

Published topic iot/sensors: {"rVibration_Temp": 45.53, "rVibration_Z_RMS_Velocity": 40.97, "rVibration_X_RMS_Velocity": 102.42, "wRMSCurrent": 104.72, "wCurrentLoad": 63.3, "wEncoderVelocity": 18.82, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:36.984", "timeInSeconds": 1672973856}

Published topic iot/sensors: {"rVibration_Temp": 45.2, "rVibration_Z_RMS_Velocity": 39.43, "rVibration_X_RMS_Velocity": 102.77, "wRMSCurrent": 103.95, "wCurrentLoad": 63.35, "wEncoderVelocity": 16.18, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:37.987", "timeInSeconds": 1672973857}

Published topic iot/sensors: {"rVibration_Temp": 45.61, "rVibration_Z_RMS_Velocity": 40.25, "rVibration_X_RMS_Velocity": 101.96, "wRMSCurrent": 103.86, "wCurrentLoad": 64.65, "wEncoderVelocity": 16.41, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:38.991", "timeInSeconds": 1672973858}

Published topic iot/sensors: {"rVibration_Temp": 45.27, "rVibration_Z_RMS_Velocity": 40.66, "rVibration_X_RMS_Velocity": 101.77, "wRMSCurrent": 103.05, "wCurrentLoad": 64.27, "wEncoderVelocity": 16.02, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:39.994", "timeInSeconds": 1672973859}

Published topic iot/sensors: {"rVibration_Temp": 46.87, "rVibration_Z_RMS_Velocity": 41.27, "rVibration_X_RMS_Velocity": 103.76, "wRMSCurrent": 102.81, "wCurrentLoad": 65.3, "wEncoderVelocity": 18.39, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:40.997", "timeInSeconds": 1672973860}

```

마지막으로 테스트 종료시에는 Ctrl + C 를 입력하여 스크립트 실행을 종료 시킵니다.

9. AWS IoT Rule 엔진 만들기

AWS IoT 규칙 작업은 규칙이 트리거될 때 어떤 작업을 수행할지 정의하는 것을 의미합니다. 예를 들어 데이터를 DynamoDB에 보내거나 AWS Lambda 함수를 트리거시키는 것이 있습니다. 본 실습에서는 AWS IoT Core로 수신된 데이터를 AWS Lambda 함수로 보내는 작업을 수행합니다.

AWS IoT 콘솔창에 접속 후, 왼쪽 사이드바에서 Manage 메뉴에서 **Message Routing > Rules**를 선택합니다. 아래의 화면이 보이면 **Create rule** 버튼을 클릭합니다.

The screenshot shows the AWS IoT Rules list interface. At the top, there are buttons for Activate, Deactivate, Edit, Delete, and Create rule. The 'Create rule' button is highlighted with a red box. Below the buttons is a search bar labeled 'Find rules'. A table below lists rules, showing columns for Name, Status, Rule topic, and Created date. A message indicates 'No rules' and 'You don't have any rules in ap-northeast-2.' At the bottom of the table is another 'Create rule' button.

생성할 규칙 이름을 설정합니다. 본 실습에서는 **GraphQL_Call_API**로 입력한 후, Next 버튼을 누릅니다.

The screenshot shows the 'Specify rule properties' step of the rule creation wizard. It includes fields for Rule name (containing 'GraphQL_Call_API'), Rule description (containing 'A description of your new rule'), and Tags. The 'Rule name' field is highlighted with a red box. At the bottom right are 'Cancel' and 'Next' buttons.

SQL 구문을 추가하여 MQTT 토픽에서 수신한 메세지를 필터링하고 다른 곳에 데이터를 푸시하는 작업을 수행합니다. 아래와 같이 값을 입력한 후, Next 버튼을 클릭합니다.

```
SELECT * FROM 'iot/sensors'
```

Configure SQL statement Info

Add a simplified SQL syntax to filter messages received on an MQTT topic and push the data elsewhere.

SQL statement

SQL version
The version of the SQL rules engine to use when evaluating the rule.
2016-03-23

SQL statement
Enter an SQL statement using the following: SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see AWS IoT SQL Reference.

```
1 | SELECT * FROM 'iot/sensors'  
2 |  
SQL Line 1, Column 1
```

Cancel Previous Next

다음 화면에서 Rule actions 항목에서 **Action 1**에 **Lambda**를 선택합니다. 그리고 Lambda 함수의 경우, 끝에 **~test**로 끝나는 함수를 선택합니다. 다음 단계로 넘어간 후, 구성 내용을 모두 살펴보고 규칙을 생성합니다.

Rule actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1

▼ Lambda
Send a message to a Lambda function Remove

Lambda function Info

reactamplifiedaa038e73-test



View

Create a Lambda function **Lambda function version**

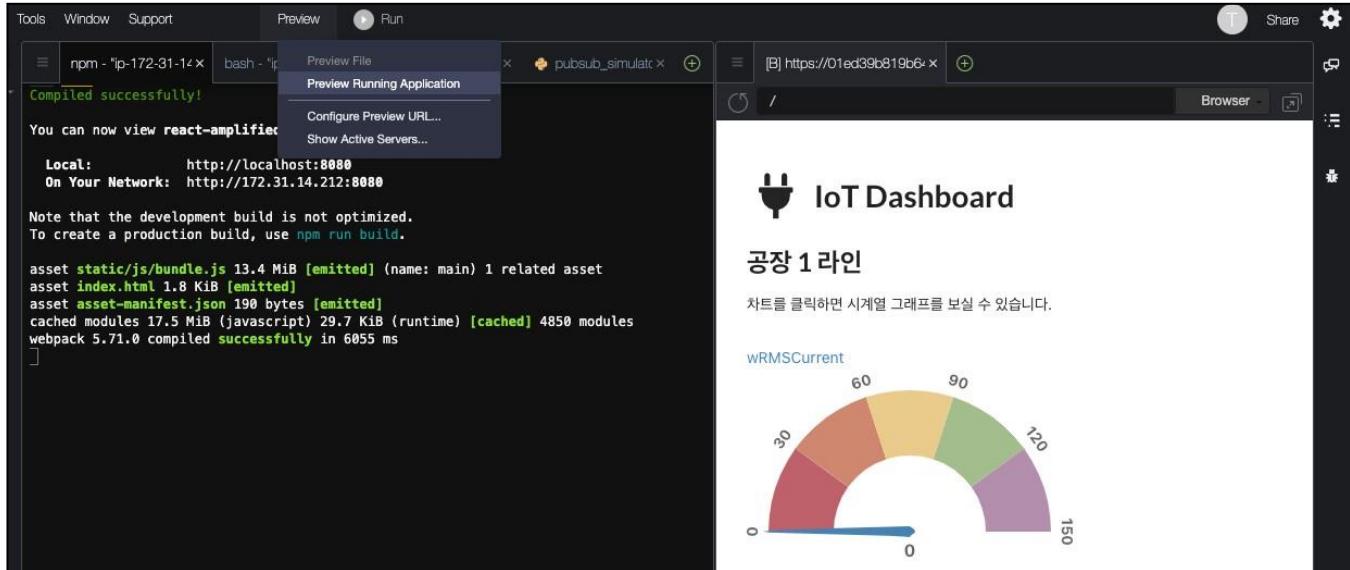
Choose Lambda function version

**Add rule action**

다시 Cloud9 IDE 환경으로 돌아가 추가 작업을 진행합니다. 기본적으로 **두 개의 터미널 창**이 필요합니다. 첫 번째 터미널 창에서는 애플리케이션을 실행합니다. 이전 실습에서 구동 시킨 후, 종료하지 않았다면 추가 작업이 필요 없으나 만약 종료했다면 아래의 명령어를 수행합니다.

```
cd iot-app-hol-online/react-amplified
npm start
```

그 다음 Cloud9 상단에 **Preview > Preview Running Application** 을 클릭하여 화면을 띠웁니다.



그 다음 두 번째 터미널 창에서 아래의 명령어를 수행합니다. 챕터 9에서 작업한 start.sh 를 시작하는 작업입니다.

```
sudo ./start.sh
```

```
dev:~/environment $ pwd
/home/ec2-user/environment
dev:~/environment $ sudo ./start.sh

Running pub/sub sample application...
Connecting to a13vpn9lg96bm-ats.iot.ap-northeast-2.amazonaws.com with client ID 'basicPubSub'...
Connected!
Published topic iot/sensors: {"rVibration_Temp": 47.24, "rVibration_Z_RMS_Velocity": 41.57, "rVibration_X_RMS_Velocity": 100.73, "wRMSCurrent": 102.13, "wCurrentLoad": 65.59, "wEncoderVelocity": 16.99, "wCylinderStatus": 0.0, "sDateTime": "2023-01-06 02:57:27.955", "timeInSeconds": 1672973847}
```

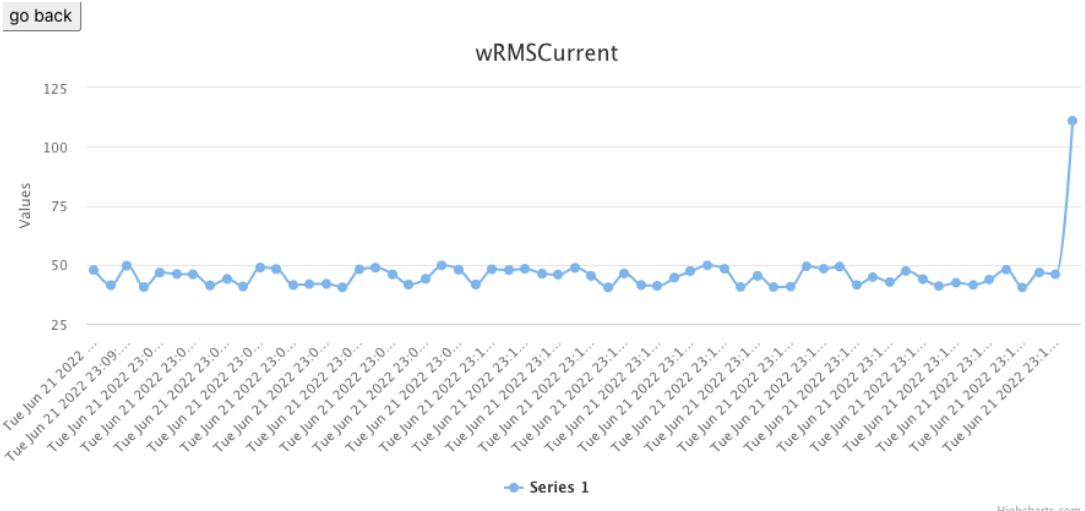
쉘 스크립트가 실행되면 thing 에서 AWS IoT Core 로 MQTT 메세지가 수신되고, 해당 메세지가 Lambda 및 AppSync API 를 통해, 화면에 나타납니다. 오른쪽 애플리케이션 화면에서 바늘이 움직이는 것을 확인할 수 있습니다.

```

npm - ip-172-31-14-212:~ sudo - ip-172-31-14-212:~ 
"rRMSCurrent": 45.31, "timeInSeconds": 1655828570, "sDateTime": "2022-06-21 14:09:30.491", "wEncoderVelocity": 22.9, "rVibration_Temp": 46.61, "rVibration_Z_RMS_Velocity": 90.92}
from topic:
iot/sensors
-----
2022-06-21 14:09:30,537 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2022-06-21 14:09:31,524 - AWSIoTPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...
2022-06-21 14:09:31,524 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS=0) event callback...
2022-06-21 14:09:31,546 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event
2022-06-21 14:09:31,547 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event
2022-06-21 14:09:31,547 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2022-06-21 14:09:31,548 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - This custom event callback is for pub/sub/unsub, removing it after invocation...
2022-06-21 14:09:31,566 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event
2022-06-21 14:09:31,568 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [message] event
Received a new message:
{"wCurrentLoad": 49.98, "wCylinderStatus": 0.0, "rVibration_X_RMS_Velocity": 86.51,
"rRMSCurrent": 44.97, "timeInSeconds": 1655828571, "sDateTime": "2022-06-21 14:09:31.524",
"wEncoderVelocity": 21.0, "rVibration_Temp": 44.7, "rVibration_Z_RMS_Velocity": 83.32}
from topic:
iot/sensors
-----
2022-06-21 14:09:31,568 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2022-06-21 14:09:32,557 - AWSIoTPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...
2022-06-21 14:09:32,557 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS=0) event callback...
2022-06-21 14:09:32,582 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event
2022-06-21 14:09:32,582 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event
2022-06-21 14:09:32,582 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2022-06-21 14:09:32,582 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - This custom event callback is for pub/sub/unsub, removing it after invocation...
2022-06-21 14:09:32,603 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event
2022-06-21 14:09:32,604 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [message] event
Received a new message:
{"wCurrentLoad": 42.45, "wCylinderStatus": 0.0, "rVibration_X_RMS_Velocity": 87.81,
"rRMSCurrent": 40.11, "timeInSeconds": 1655828572, "sDateTime": "2022-06-21 14:09:32.557",
"wEncoderVelocity": 22.23, "rVibration_Temp": 48.51, "rVibration_Z_RMS_Velocity": 95.45}
from topic:
iot/sensors
-----
2022-06-21 14:09:32,604 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...

```

차트를 클릭하면 시계열 그래프를 확인할 수 있습니다.



10. 대시보드에 frontend 코드 추가하기 (optional)

App.js에서 시뮬레이터에서 보내는 값 중 하나를 대시보드에 더 추가해보겠습니다.

아래와 같이 기존 대시보드에 rVibration_Temp값을 추가하기 위해

코드를 추가합니다.

```
function App() {
  const [wCurrentLoad, setWCurrentLoad] = useState();
  const [wRMSCurrent, setWRMSCurrent] = useState();
  const [rVibration_Temp, setRVibration_Temp] = useState();

  useEffect(() => {
    const RMSsubscription = API.graphql(
      graphqlOperation(subscriptions.onCreateSensor, {
        sensorType: "wRMSCurrent",
      })
    ).subscribe({
      next: (eventData) => {
        setWRMSCurrent(eventData.value.data.onCreateSensor.value);
      },
    });
  });

  const Loadsubscription = API.graphql(
    graphqlOperation(subscriptions.onCreateSensor, {
      sensorType: "wCurrentLoad",
    })
  ).subscribe({
    next: (eventData) => {
      setWCurrentLoad(eventData.value.data.onCreateSensor.value);
    },
  });
}
```

```

const Vibration_Tempsubscription = API.graphql(
  graphqlOperation(subscriptions.onCreateSensor, {
    sensorType: "rVibration_Temp",
  })
).subscribe({
  next: (eventData) => {
    setRVibration_Temp(eventData.value.data.onCreateSensor.value);
  },
});

return () => {
  console.log("clean up in main");

  RMSsubscription.unsubscribe();
  Loadsubscription.unsubscribe();
  Vibration_Tempsubscription.unsubscribe();
};

}, []);

```

ReactSpeedometer를 아래와 같이 추가해줍니다.

```

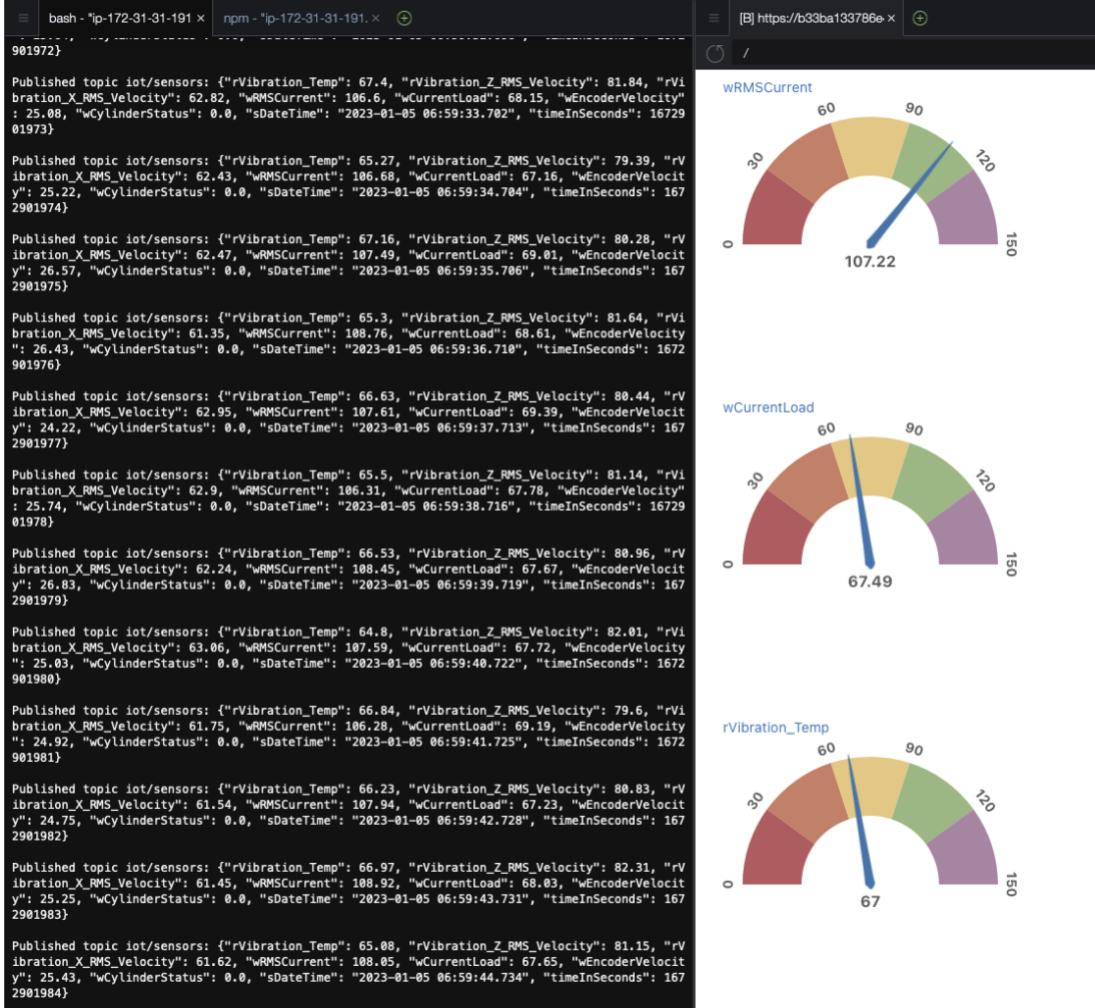
<Link to={"/detailPage?sensorType=rVibration_Temp"}>
  rVibration_Temp
  <ReactSpeedometer
    width={300}
    height={300}
    maxValue={150}
    segments={5}
    segmentColors={[
      "#bf616a",
      "#d08770",
      "#ebcb8b",
      "#a3be8c",
      "#b48ead",
    ]}
  >

```

```
value={rVibration_Temp}
/>
</Link>
```

App.js파일을 저장하고 터미널창에서 npm start가 실행 되어 있지 않을 경우 npm start를 통해 다시

변경사항을 빌드하고 preview 화면을 통해 rVibration_Temp값이 시뮬레이터에 의해 변경되는 것을 확인할 수 있습니다.



11. 호스팅 추가하기 (optional)

위 실습에서는 frontend 서비스를 local 환경에서 preview를 통해 테스트 해봤다면 실제 호스팅은 어떤 방식으로 하는지 간단하게 실습을 해보겠습니다.

호스팅을 하는 방법은 "Hosting environments"에서 github, codecommit, gitlab, bitbucket등 다양한 리파지토리와 연결하여 연결된 리파지토리에 변동사항이 생길 경우 자동으로 배포되는 CI/CD파이프라인을 구축할 수도 있습니다.

The screenshot shows the AWS Amplify console interface for the 'reactamplified' app. On the left sidebar, under 'App settings', 'General' is selected. The main content area is titled 'reactamplified' and displays the message: 'The app homepage lists all deployed frontend and backend environments.' Below this, there are two tabs: 'Hosting environments' (selected) and 'Backend environments'. Under 'Hosting environments', there is a section titled 'Host a web app' with the sub-instruction 'Connect your source code from a Git repository or upload files to host a web app in minutes.' Five options are listed: GitHub (selected), Bitbucket, GitLab, AWS CodeCommit, and Deploy without Git provider. At the bottom right of this section is a red 'Connect branch' button.

본 실습에서는 cloud9에서 manual deployment를 통해 클라우드 상에서 frontend 서비스를 확인해 보겠습니다. 지금까지 실습했던 Amplify 콘솔에서 All apps > reactamplified 에서는 백엔드 API, 인증, Lambda함수 등이 생성되어 있지만 frontend 서비스를 볼 수는 없었습니다.

Deployment가 되면 호스팅된 도메인 URL이 표시가 됩니다.

The screenshot shows the AWS Amplify console interface for the 'reactamplified' app. On the left sidebar, under 'App settings', 'General' is selected. The main content area is titled 'reactamplified' and displays the message: 'The app homepage lists all deployed frontend and backend environments.' Below this, there are two tabs: 'Hosting environments' (selected) and 'Backend environments'. Under 'Backend environments', there is a section titled 'test' with the sub-instruction 'Continuous deploys not set up.' To the right of this, the 'Deployment status' is shown as 'Deployment completed 1/9/2023, 8:31:58 PM'. Below the deployment status, there are sections for 'Categories added' (API, Authentication, Functions) and a note: 'A visual interface for managing your backend outside the AWS console.' At the bottom left is a blue 'Set up Amplify Studio' button, and at the bottom center is a blue 'Edit backend' button.

Cloud9 터미널에서 아래와 같이 호스팅을 추가하는 명령을 실행합니다.

amplify hosting add

화살표를 따라 Hosting with Amplify console 선택 후

Hosting type을 선택하라고 하면 “**Manual deployment**”를 선택합니다.

```
TeamRole:~/environment/iot-app-hol-online/react-amplified (main) $ amplify hosting add
✓ Select the plugin module to execute · Hosting with Amplify Console (Managed hosting with custom domains, Continuous deployment)
? Choose a type Manual deployment

You can now publish your app using the following command:

Command: amplify publish
TeamRole:~/environment/iot-app-hol-online/react-amplified (main) $ amplify publish
```

호스팅을 추가한 후 실제 deploy를 수행하는 publish 명령을 수행합니다.

amplify publish

Continue에 “Y”를 입력합니다.

```
TeamRole:~/environment/iot-app-hol-online/react-amplified (main) $ amplify publish
: Fetching updates to backend environment: test from the cloud. ✓ GraphQL schema compiled successfully.

Edit your schema at /home/ec2-user/environment/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema.graphql
nt/iot-app-hol-online/react-amplified/amplify/backend/api/reactamplified/schema
✓ Successfully pulled backend environment test from the cloud.

Current Environment: test



| Category | Resource name          | Operation | Provider plugin   |
|----------|------------------------|-----------|-------------------|
| Hosting  | amplifyhosting         | Create    | awscloudformation |
| Auth     | reactamplified092bb062 | No Change | awscloudformation |
| Api      | reactamplified         | No Change | awscloudformation |
| Function | reactamplifiedaa038e73 | No Change | awscloudformation |



? Are you sure you want to continue? (Y/n) ■
```

AWS IoT (Amplify, AppSync, Cognito) Application Hands-on

```
Deployment completed.
Deployed root stack reactamplified [ ===== ] 5/5
  amplify-reactamplified-test AWS::CloudFormation::Stack    UPDATE_COMPLETE
  hostingamplifyhosting     AWS::CloudFormation::Stack    CREATE_COMPLETE
  authreactamplified092bb062 AWS::CloudFormation::Stack    UPDATE_COMPLETE
  apireactamplified        AWS::CloudFormation::Stack    UPDATE_COMPLETE
  functionreactamplifiedaa038e73 AWS::CloudFormation::Stack    UPDATE_COMPLETE
Deployed hosting amplifyhosting [ ===== ] 1/1
  AmplifyBranch           AWS::Amplify::Branch      CREATE_COMPLETE
Mon Jan 09 2023 11:32:49...
Mon Jan 09 2023 11:32:15...
Mon Jan 09 2023 11:32:06...
Mon Jan 09 2023 11:32:30...
Mon Jan 09 2023 11:32:33...
Mon Jan 09 2023 11:32:11...

GraphQL transformer version: 2

Publish started for amplifyhosting

> react-amplified@0.1.0 build
> react-scripts build

Creating an optimized production build...
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
Compiled successfully.

File sizes after gzip:

  530.06 kB  build/static/js/main.c5b96e86.js
  16.47 kB   build/static/css/main.566dd300.css

The bundle size is significantly larger than recommended.
Consider reducing it with code splitting: https://goo.gl/9VhYWB
You can also analyze the project dependencies: https://goo.gl/LeUzfb

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

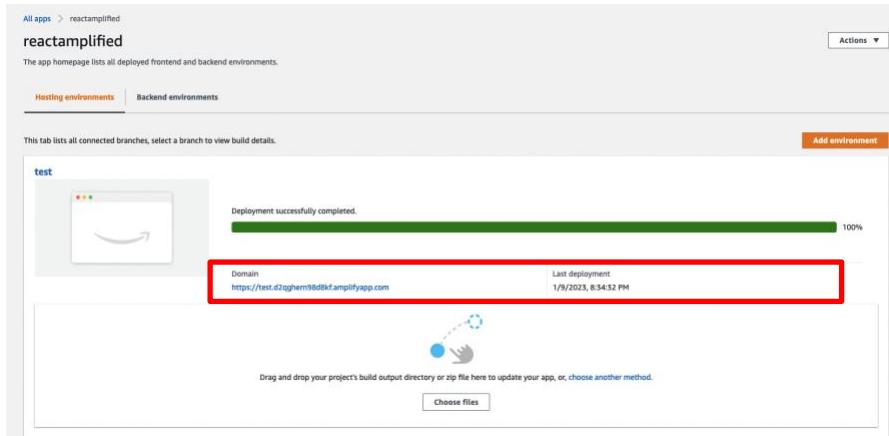
  npm install -g serve
  serve -s build

Find out more about deployment here:

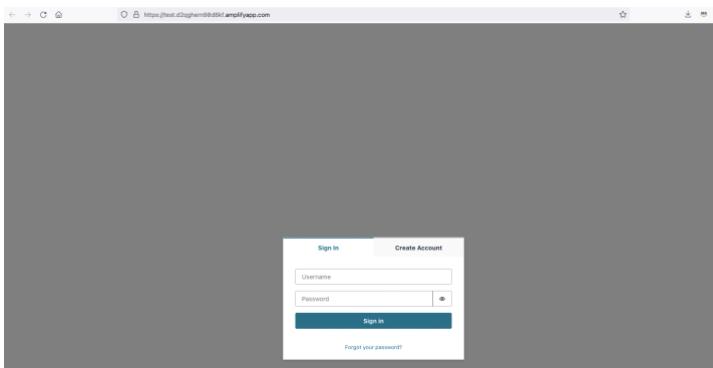
  https://cra.link/deployment

✓ Zipping artifacts completed.
✓ Deployment complete!
https://test.d2qghern98d8kf.amplifyapp.com
TeamRole:~/environment/iot-app-hol-online/react-amplified (main) $
```

Deployment complete 메시지가 보이고 종료되면 frontend에 대한 호스팅 배포가 완료 되었습니다. Amplify Console에서 아래와 같이 실제 cloudfront와 연계된 frontend url이 생성된 것을 확인할 수 있습니다.



배포된 url을 통해 로그인 화면에 접속할 수 있습니다.



https://test.d2qghern98d8kf.amplifyapp.com

🔌 IoT Dashboard

공장 1 라인

차트를 클릭하면 시계열 그래프를 보실 수 있습니다.

