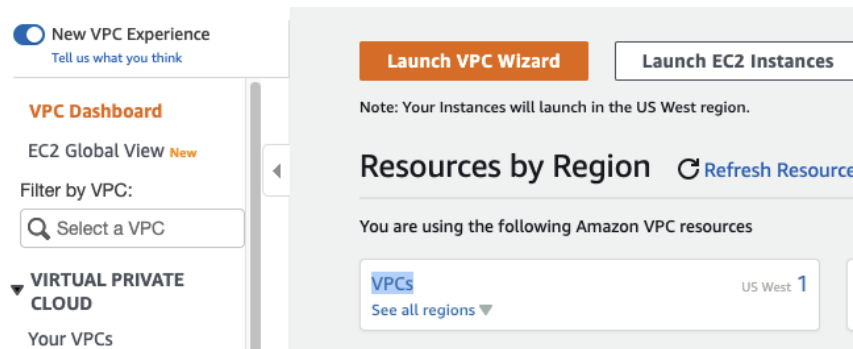




01) PostgreSQL, Dyanmodb 환경구성

[1] 실습 주의 사항

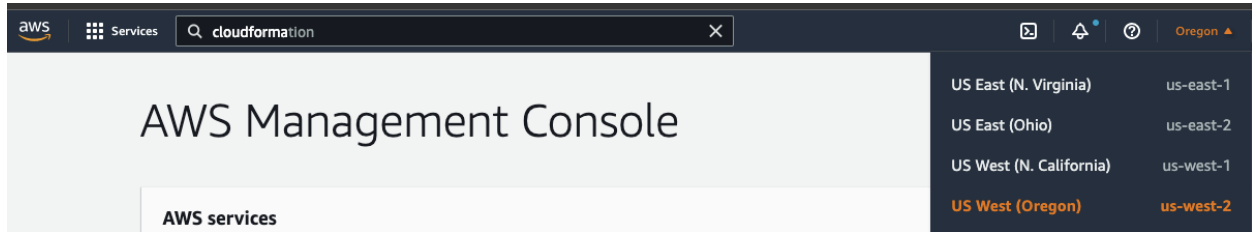
- IAM 사용자 : AWS 모범 사례에 따라 루트 계정 사용을 지양하고, 실습에 맞는 권한을 가진 IAM 사용자를 생성하시기를 바랍니다.
- 실습 region : (us-west-2 인 Oregon) 또는 (us-east-1 인 N.virginia) 을 기준으로 진행합니다
- VPC : 해당 리전의 실습용 VPC 2개 (해당 리전에서 사용하시는 VPC가 3개 이하이면 됩니다.)
 - 확인방법 : AWS management Console 돋보기에서 VPC 입력 —> VPC새화면 에서 보이는 VPCs 값이 3보다 작으면 됩니다.



- VPC 값이 4 또는 5 인 경우 : 삭제 가능한 VPC 삭제 후 3개 이하로 낮춤

[2] cloudformation 이용 PostgreSQL, Dyanmodb setup

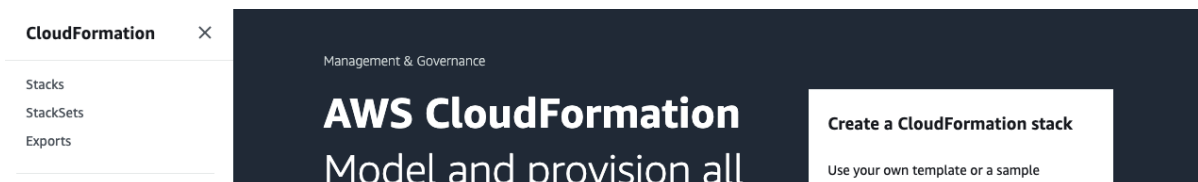
1. AWS management Console에서 우측 상단 region 을 (us-west-2 인 Oregon) 또는 (us-east-1 인 N.virginia) 선택하시고 해당 리전은 다시 지정 필요가 있어 메모해 두시기 바랍니다 (%주의 필요)



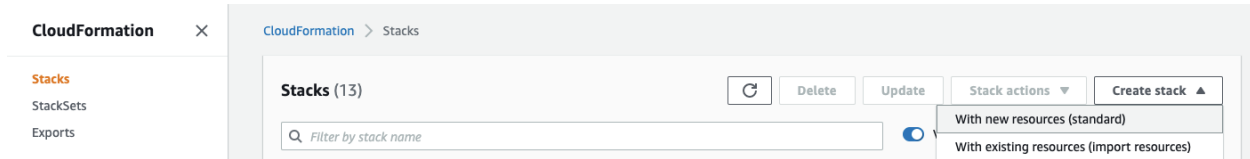
2. 첨부파일 builder202204.zip 파일의 cloudformation 폴더 안에 PurposeBuiltDB.json 파일 확인

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/fbf4bcf1-4850-4c6f-b0ed-b711043e963f/builder202204.zip>

3. AWS management Console 돋보기에서 cloudformation 입력 → cloudformation 새 화면 → 메뉴 좌측 상단 stack 선택

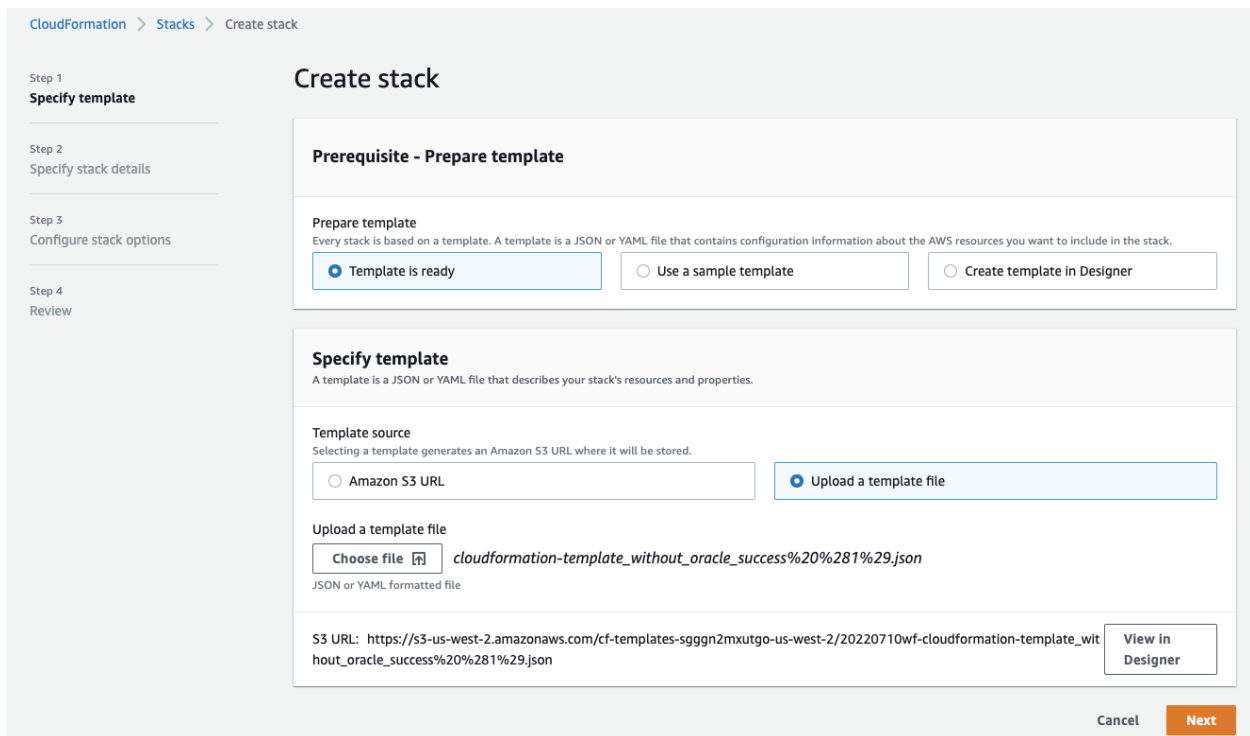


4. cloudformation 메뉴 → 우측 상단 create stack → with new resource (standard) 선택



5. 아래 화면기준 cloudformation step1 단계 --> 아래와 같이 설정후 Upload a template file 부분에서 **PurposeBuiltDB.json**을 선택 → next 클릭

- (다운로드한 Builder202204.zip 파일 압축을 풀고 cloudformation 폴더 안의 PurposeBuiltDB.json 임)



6. 아래 화면기준 cloudformation step2. step3 → stack name: **PurposeBuiltDB** 입력 → next 클릭 → next 클릭

- 참고 AuroraDBUsername : auradmin 이며 AuroraDBPassword 는 auradmin123 입니다.)

CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Specify stack details

Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Target Aurora PostgreSQL Database Configuration

AuroraDBName
Enter Aurora Database name

AuroraDBUsername
Enter Master admin username for Aurora RDS

AuroraDBPassword
Enter password for Aurora RDS Admin user

AuroraInstanceType
Aurora PostgreSQL DB Instance type

AuroraEngineType
Aurora DB Engine type

DMS Configuration

DMSInstanceType
DMS Replication instance type

ExistsDMSRole
check if the dms-vpc-role exists in your account, please enter Y, else enter N

Enter IP address for the DB Security group Configuration

ClientIP
The IP address range that can be used to connect to the RDS instances from your local machine. It must be a valid IP CIDR range of the form x.x.x.x/x. Pls get your address using checkip.amazonaws.com or whatismyip.org

Cancel Previous Next

7. 아래 화면기준 step 4 Review → 아래에 체크 → create stack 클릭

Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.



The template has changed

CloudFormation has detected changes between the template uploaded and the one being used for this operation. Verify that the changes are intentional.



Cancel

Previous

Create change set

Create stack

8. cloudformation 설치확인 → PurposeBuiltDB 와 aws-cloud9-Project-PurposeBuiltDB... 모두 CREATE_COMPLETE 임을 확인 (약 10분 소요)

CloudFormation > Stacks

Stacks (13)



Delete

Filter by stack name



View nested

	aws-cloud9-Project-PurposeBuiltDB-357c766b95f94a9b8a331022ef57d622	CREATE_COMPLETE	2022-03-02 20:51:08 UTC+0900
	PurposeBuiltDB	CREATE_COMPLETE	2022-03-02 20:50:29 UTC+0900

9. AWS Management console 돋보기에서 RDS 입력 → 새화면 Amazon RDS 에서 좌측 Databases 클릭 → 우측 상단에 생성된 DB identifier중 상단 purposebuiltdb-auroracluster... 을 클릭 → 하단의

2개 endpoint중 writer instance endpoint 복사 및 저장 (이후 데이터베이스 접속시 사용)

Amazon RDS

- Dashboard
- Databases**
- Query Editor
- Performance Insights
- Snapshots
- Automated backups
- Reserved Instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Events
- Event subscriptions
- Recommendations
- Certificate update

RDS > Databases > purposebulddb-auroracluster-1l28k7to3fccv

purposebulddb-auroracluster-1l28k7to3fccv

Modify Actions

Related

Filter by databases

DB identifier	Role	Engine	Region & AZ	Size	Status
purposebulddb-auroracluster-1l28k7to3fccv	Regional cluster	Aurora PostgreSQL	us-west-2	1 instance	Available
pa1erlee2wthz3v	Writer instance	Aurora PostgreSQL	us-west-2a	db.r5.large	Available

Connectivity & security Monitoring Logs & events Configuration Maintenance & backups Tags

Endpoints (2)

Filter by endpoint

Actions Create custom endpoint

Endpoint name	Status	Type	Port
purposebulddb-auroracluster-1l28k7to3fccv.cluster-cymvdb9y03l.us-west-2.rds.amazonaws.com	Available	Writer instance	5432
purposebulddb-auroracluster-1l28k7to3fccv.cluster-ro-cymvdb9y03l.us-west-2.rds.amazonaws.com	Available	Reader instance	5432

10. AWS Management console → 돋보기에서 cloud9 입력 및 이동 → 새화면 AWS cloud9 에서 Project-PurposeBuiltDB 선택 → open IDE클릭

AWS Cloud9

- Your environments**
- Shared with you
- Account environments
- How-to guide

AWS Cloud9 > Your environments

Your environments (2)

Open IDE View details Edit Delete Create environment

labs-Cloud9-IDE

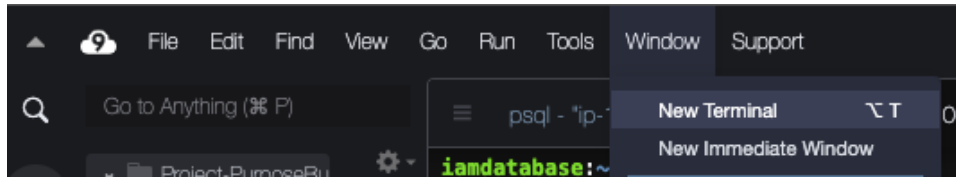
Project-PurposeBuiltDB

Type: EC2
Permissions: Owner
Description: labs-Cloud9-IDE
Owner Arn: arn:aws:iam::692464257295:user/iamdatabase

Type: EC2
Permissions: Owner
Description: No description available
Owner Arn: arn:aws:iam::692464257295:user/iamdatabase

Open IDE Open IDE

11. 새화면 cloud9 에서 상단 window → new Terminal 을 클릭하여 새로운 terminal 을 오픈함



12. cloud9 터미널에서 다음 내용 입력하여 posgreSQL 관련 설치 작업 수행

```
# shell 에서 수행
cd ~/environment
wget https://yum.postgresql.org/11/redhat/rhel-6.9-x86_64/postgresql11-libs-11.9-1PGDG.rhel6.x86_64.rpm
wget https://yum.postgresql.org/11/redhat/rhel-6.9-x86_64/postgresql11-11.9-1PGDG.rhel6.x86_64.rpm

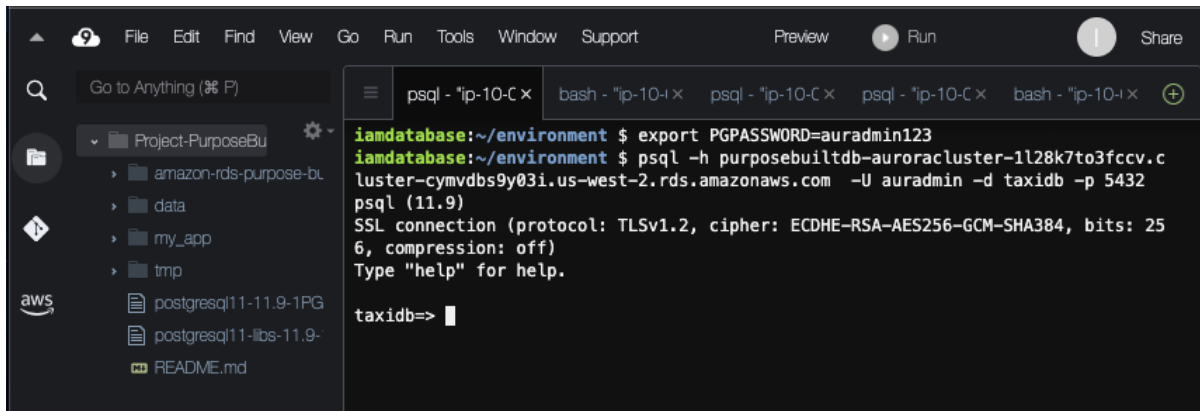
sudo yum clean all
sudo rpm -ivh postgresql11-libs-11.9-1PGDG.rhel6.x86_64.rpm
sudo rpm -ivh postgresql11-11.9-1PGDG.rhel6.x86_64.rpm
```

13. cloud9 터미널에서 psql 을 수행하여 PostgreSQL database 접속 (관련 형식 `psql -h <위에서 복사한 endpoint> -U <Username> -d <Database Name> -p <Port>`)

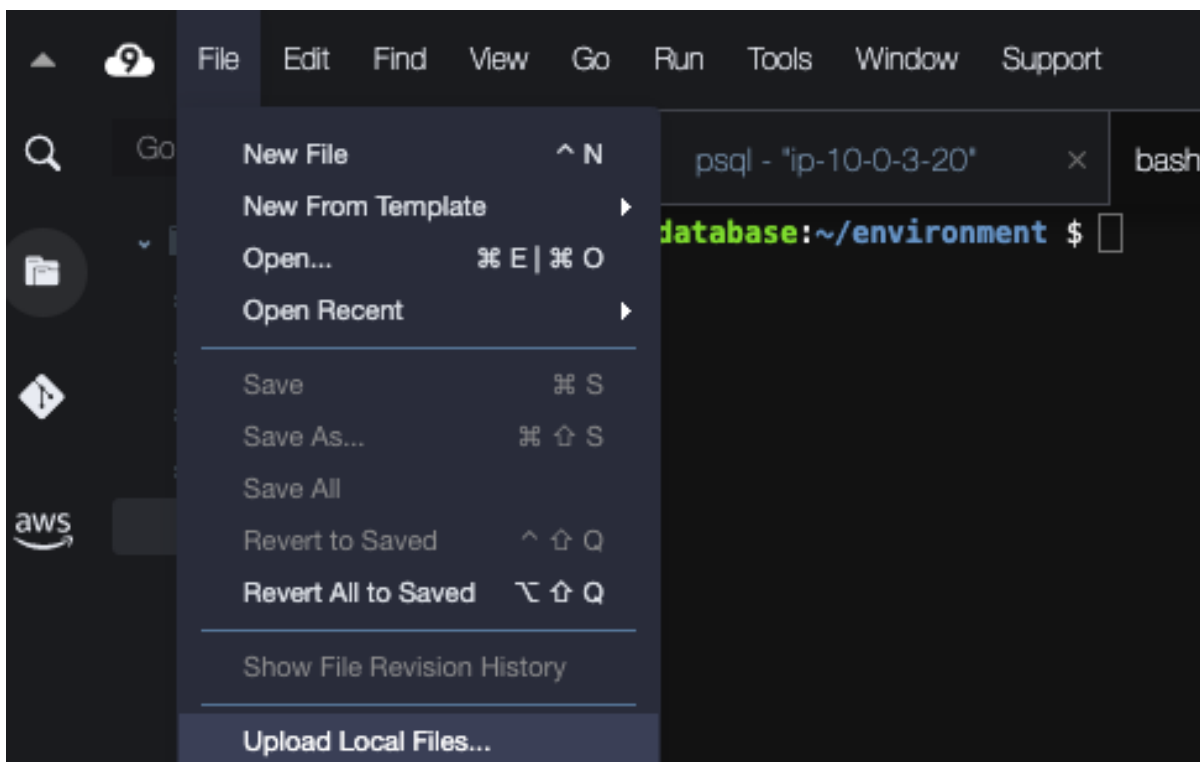
- 아래와 같이 taxidb=> 결과가 보이면 성공입니다.

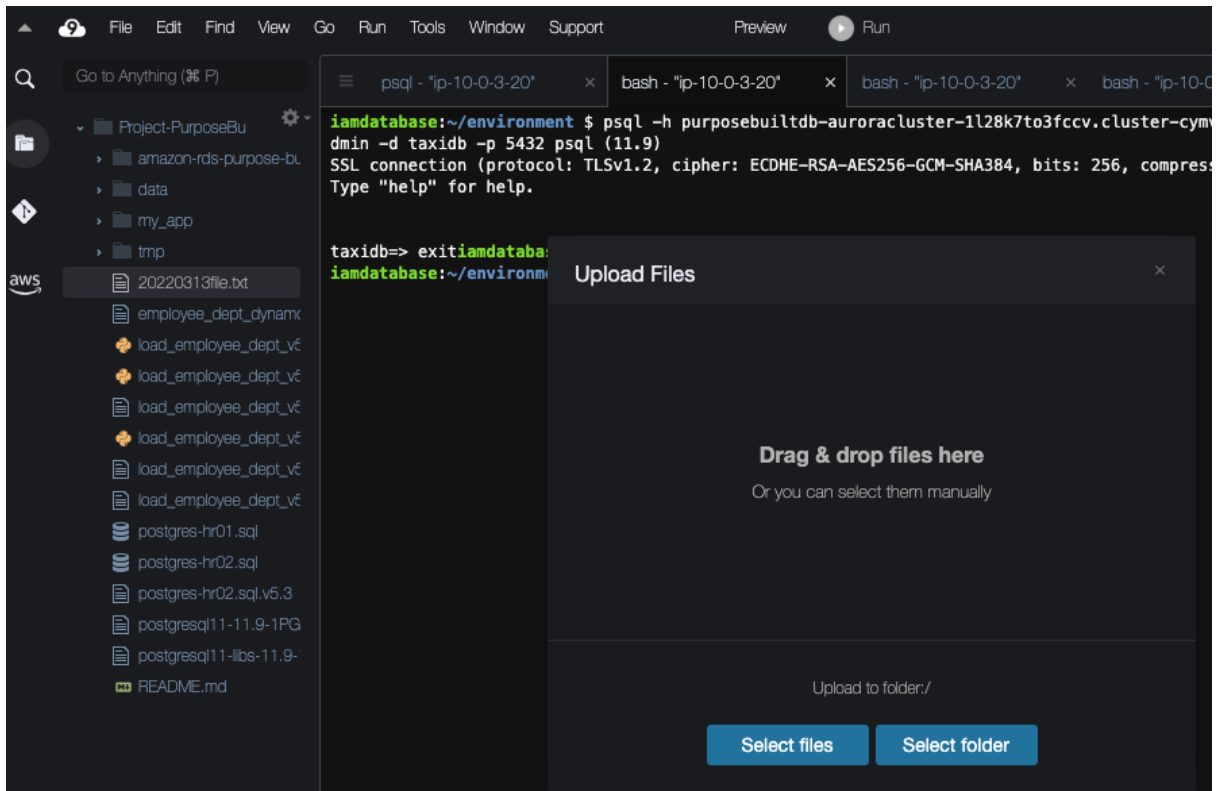
```
# shell 에서 수행
cd ~/environment
export PGPASSWORD=auradmin123

# PostgreSQL 에서 수행
psql -h <앞에서복사한 PostgreSQL endpoint> -U auradmin -d taxidb -p 5432
exit
```

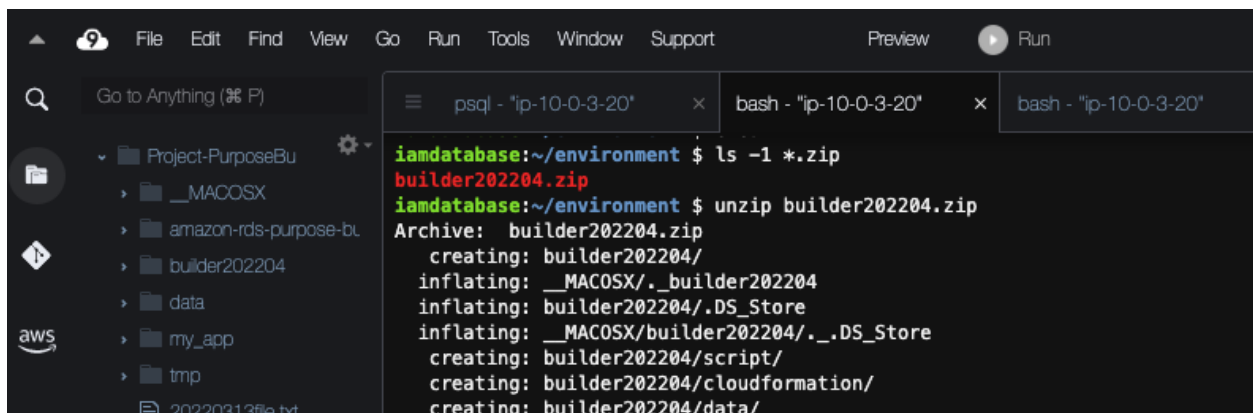


14. 터미널에서 `cd ~/environment` 수행후 → cloud9 메뉴 → file → upload local files 선택 → 팝업된 화면위에 다운로드한 builder202204.zip 파일을 drag & drop → upload files 화면 종료





15. 터미널에서 `ls -l *.zip`을 통해 해당 `builder202204.zip` 파일이 아래와 같이 업로드 됨을 확인 —> `unzip builder202204.zip` 수행



```
# shell 에서 수행
# 파일이 존재함을 확인
ls -l *.zip
```

```
# 해당 파일 압축 풀기
unzip builder202204.zip
```

-
- 수고하셨습니다. 다음 챕터로 이동하세요 —> 🍌 02) PostgreSQL 에서 초기데이터적재, 입력, 수정, 삭제, 조회 수행