INZENT

aws Qualified Software

MyData Shield V1.0

Contents

1. Product Overview 4

1.1 Introduction 4

1.1.1 Prerequisites and Requirements 4

1.1.2 Region support 5

1.1.3 Architecture Diagrams 5

1.1.4 use case 6

2. Planning Guidance 6

2.1 Security 6

2.2 Costs and Licenses 7

2.3 Sizing 7

3. Deployment steps 8

3.1 Mydata Shield Installation 8

3.1.1 Create VPC AND Subnet 8

3.1.2 Create Network ACLs 9

3.1.3 Create Security Group 11

3.1.4 Create Instance 14

4. Operational Guidance 17

4.1 Support for MyData Shield Backup and Restore in AWS 18

4.1.1 MyData Shield Backup and Restore 18

4.2 MyData Shield Health Check with CloudWatch 19

4.3 Database Credentials 21

4.4 Routine Maintenance 21

4.5 Emergency Maintenance 22

4.5.1 Startup process 22

4.5.2 Health Check 22
4.5.3 Type of MyData Shield failures 23
4.5.4 Recovery procedure for MyData Shield failure 23
4.5.5 Recovery procedure when MyData Shield recovery fails 25
4.5.6 RTO 26
5. Solution operation 26
5.1 How to set 26
5.2 How to run 31
5.3 Key management 31
5.4 Patches and updates management 32
6. Support 33
6.1 Technical support 33
6.2 Support Costs 33
6.3 SLA 34

1. Product Overview

This article assumes you have used AWS before and are familiar with AWS services. If you are new to AWS, please refer to the AWS documentation (https://docs.aws.amazon.com/). You should also be familiar with the following AWS technologies:

:

- Amazon Virtual Private Cloud(Amazon VPC) - The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS Cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

- Amazon EC2 – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.

1.1 Introduction

MyData Shield plays the role of pseudonymizing/anonymizing personal information contained in Json format MyData in order for MyData operator to analyze the MyData of the information provider. If you set an 'item name' that requires pseudonymization set according to the MyData API standard, the data is searched for and stored in eXperDB after pseudonymization is flexibly processed.

- What is MyData?

  As Korea's 3 data law was passed, the MyData concept of "I am the owner of my data" is attracting attention. Once the MyData concept is established, consumers can peddle the sovereignty of the data they create, and financial companies can find new business models such as customized asset management by receiving data with individual consent.

1.1.1 Prerequisites and Requirements

This topic describes the prerequisites and resource requirements for using MyData Shield with Amazon Web Services (AWS).

- **Prerequisites**

The MyData Shield AMI is a standalone solution that requires no additional software to be installed. It is possible to deploy MyData Shield through basic AWS technology, and it is deployed including AWS EC2 only.

For MyData Shield, the AMI is available on Amazon Linux, and you can use the OS you are familiar with on Linux.

The database of MyData Shield is installed in the image where Postgresql-based eXperDB is deployed.

- **Requirements**

Virtual Machines(VMs) are required to install MyData Shield

VM Name (Tag)	VM type	Default VM Count
MyData Shield	t2.medium or t3.medium	**1**

The number may change according to customer environment.

1.1.2 Region support

The regions where the product is supported are as follows

Region code	Region Name	Remarks
us-east-1	US East (N. Virginia)	-
ap-northeast-2	Asia Pacific(Seoul)	-

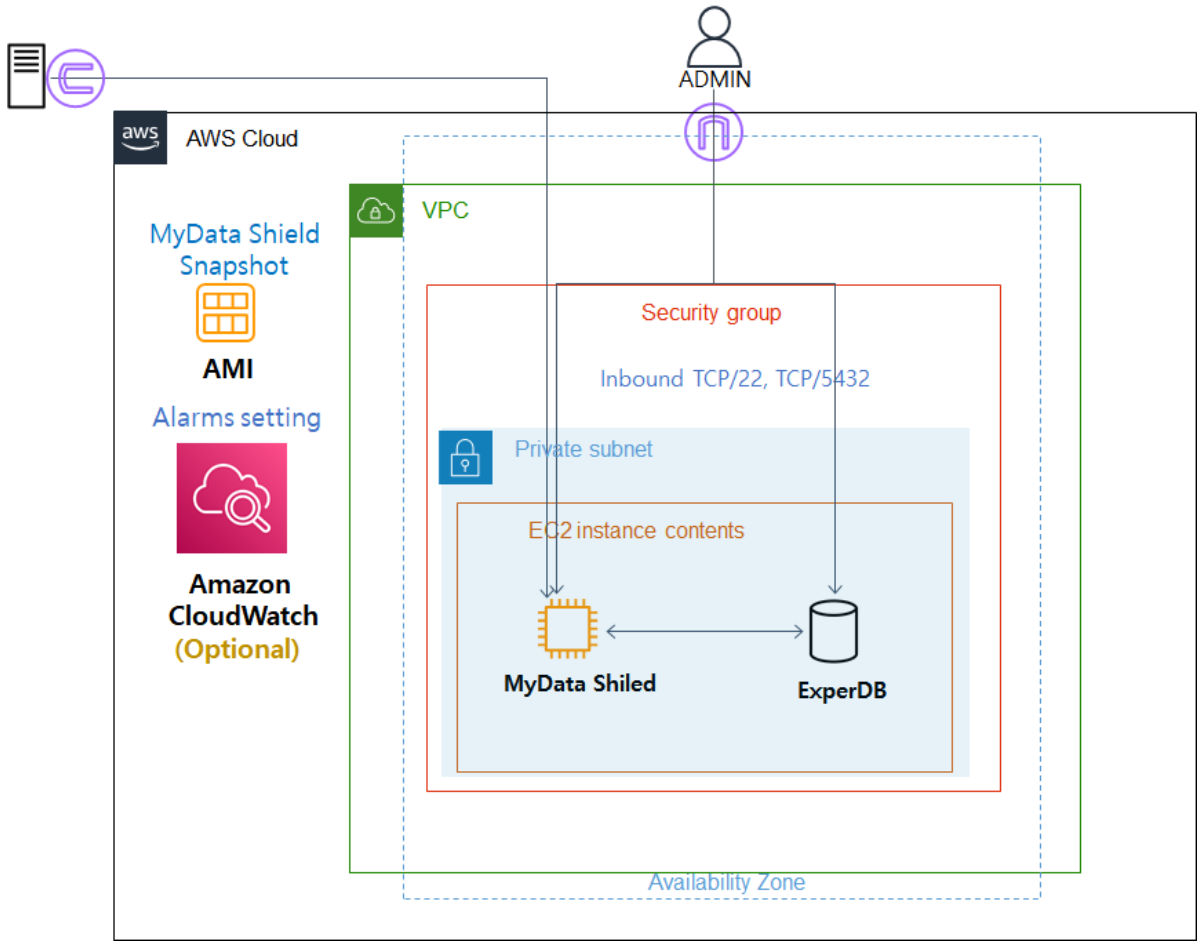
1.1.3 Architecture Diagrams

- Private subnet, MyData Shield EC2 instance

- Backup and Multiplex Using Amazon Machine Image(AMI)

- **(optional)** MyData Shield health monitoring and notification using Amazon Cloud Watch service

- **MyData Shield Architecture Diagrams**



1.1.4 use case

The link below is an example of a project that performed pre-processing using MyData Shield's pseudonymous processing function
https://m.etnews.com/20211223000103?obj=Tzo4OiJzdGRDbGFzcyI6Mjp7czo3OiJyZWZlcmVyIjtOO3M6NzoiZm9yd2FyZCI7czoxMzoid2VilHRvIG1vYmlsZSI7fQ%3D%3D

2. Planning Guidance

2.1 Security

The only thing you need to be able to install/control you MyData Shield deployment is SSH access (key-based authentication/sudo or similar mechanisms are preferred)

- Not using AWS root credentials for access.

2.2 Costs and Licenses

The MyData Shield product is provided free of charge. In addition, please contact us through the link below for user configuration information, database configuration information, and AMI provision that exist in thse EC2 product.
Link : http://www.inzent.com/board/board.php?bo_table=faq&pageName=main

- Full list of billable AWS services

  You are responsible for the cost of AWS services. The cost of the resources generated by the menu depends on the instance you use. For more information, see the pricing pages for the AWS services you use in this guide (https://aws.amazon.com/pricing/).

  A. EC2 Instance(essential)

  B. EBS(essential)

  C. Cloudwatch(optional)

  D. Secret Manager(optional)

2.3 Sizing

MyData Shield's AMI support the instance specifications shown in the table below on AWS. For up-to-date information on each type of instance, please refer to the link next to it.
(https://aws.amazon.com/ko/ec2/instance-types/)

Count of data	Instance type	Vcpu	Memory(GiB)	EBS Volume	EBS Volume Type
~5000000	t2.medium or t3.medium	2	4	50GB	General Purpose SSD (gp2)
~100000000	t2.large or t3.large	2	8	1T	General Purpose SSD (gp2)
~1000000000	t2.large or t3.large	2	8	10T	General Purpose SSD (gp2)

3. Deployment steps

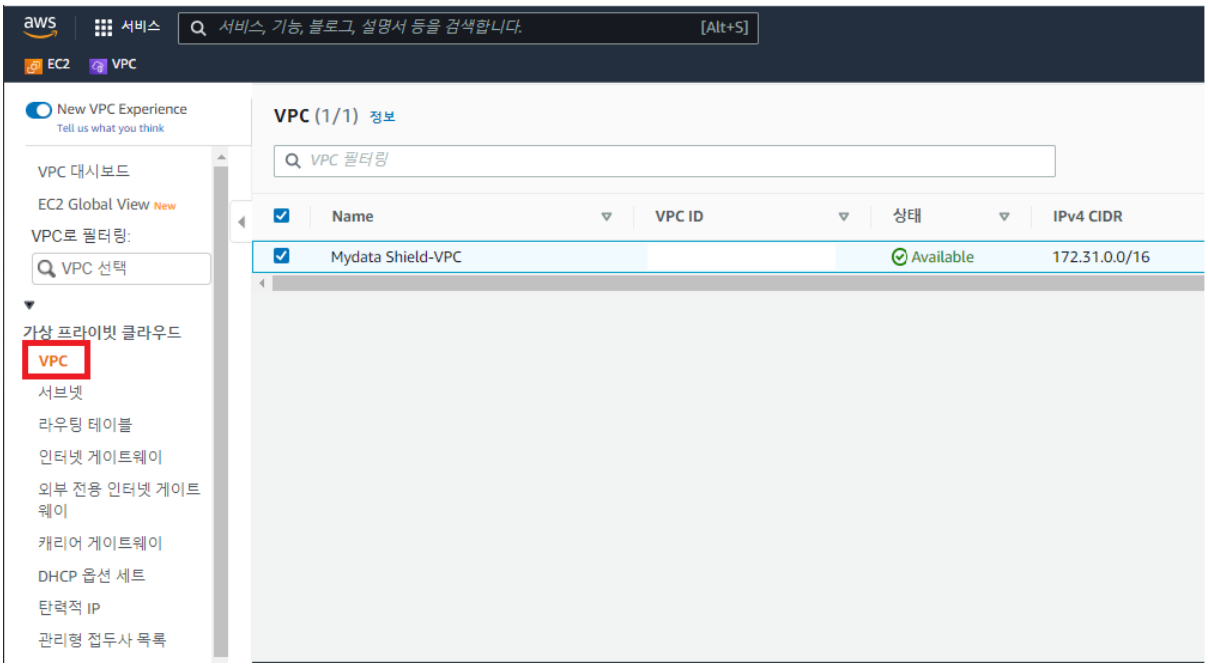
3.1 Mydata Shield Installation

3.1.1 Create VPC AND Subnet

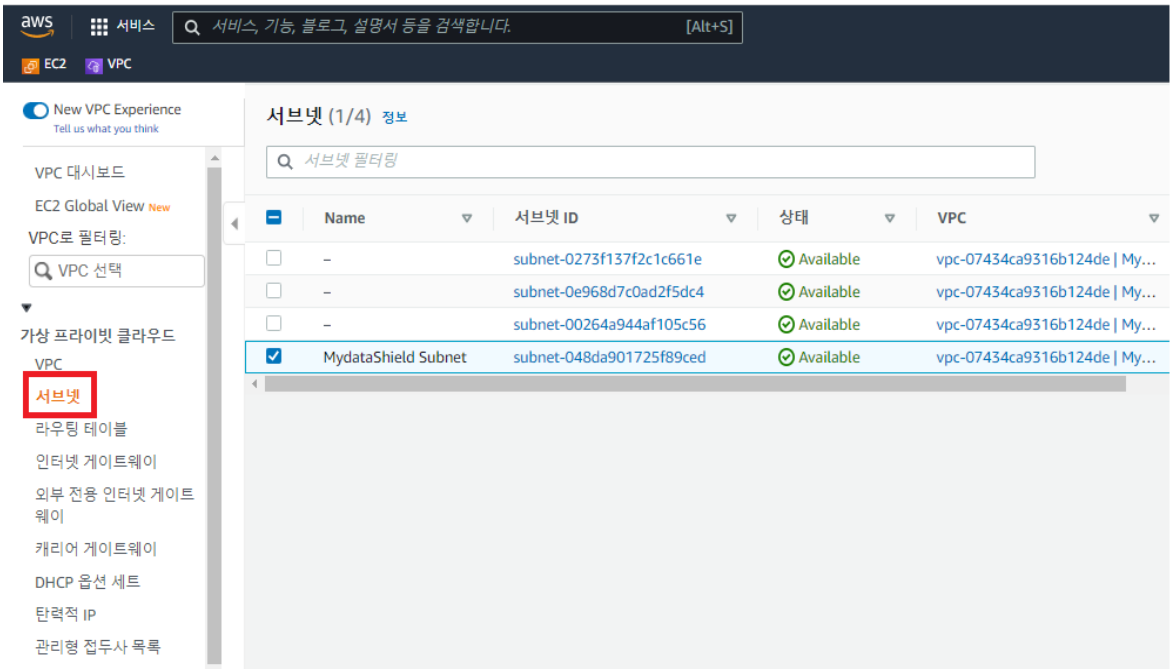
Subnet Setting that manage MyData Shield.

- Check existing VPCs and subnets

1. [console home> service > Networking and content delivery> VPC] You can check the VPC setting



2. [console home> service > Networking and content delivery> VPC > subnet] You can check the subnet setting



- Create MyData Shield subnet

1. Go to the Amazon VPC console(https://console.aws.amazon.com/vpc).
2. In the navigation pane, select Subnet and then select Create Subnet.
3. Optionally specify subnet details and choose Create.

Menu	Input Value
Subnet name	MyData Shield Subnet
VPC	Choose an existing VPC that is the same as your web tier
VPC CIDRS	-
Availability Zone	Refer to [1.1.3] Architecture Diagrams
IPv4 CIDR block	For information about Subnet group, see the following link. : https://docs.aws.amazon.com/ko_kr/vpc/latest/userguide/VPC_Subnets.html

3.1.2 Create Network ACLs

Optional : If you nee an extra layer of security, you can create network ACLs and add rules

- Create MyData Shield Network ACL

1. Open the Amazon VPC console at (https://console.aws.amazon.com/vpc/).

2. In the navigation pane, choose Network ACLs.

3. Choose Create Network ACL.

4. In the Create Network ACL

Menu	Input Value
Name tage	MyData Shield NACL
VPC	Choose the same existing VPC as web tier of the user and MyData storage device

5. In the navigation pane, choose Network ACLs.

6. Depending on the type of rule you need to add to the details select the Inbound Rules, Outbound Rules tab to edit.

- Inbound rule

Rule#	Source IP	Protocol	Port	Allow/Deny	Comments
100	IP address range of the solution operator	TCP	22	Allow	Allow SSH traffic from solution users
110	IP address range of MyData storage device	TCP	5432	Allow	Allow incoming RDB traffic from MyData storage device
*	0.0.0.0/0	all	all	Deny	-

- Outbound rule

Rule#	Source IP	Protocol	Port	Allow/Deny	Comments

100	IPv4 address range of devices from which users want to utilize data	TCP	5432	허용	Allow outbound responses to the MyData Shield RDB network
*	0.0.0.0/0	모두	모두	거부	-

7. When you are done, choose Save

8. Associating a Subnet with a Network ACL, In the navigation pane, choose Network ACLs, and then select [MyData Shield NACL].

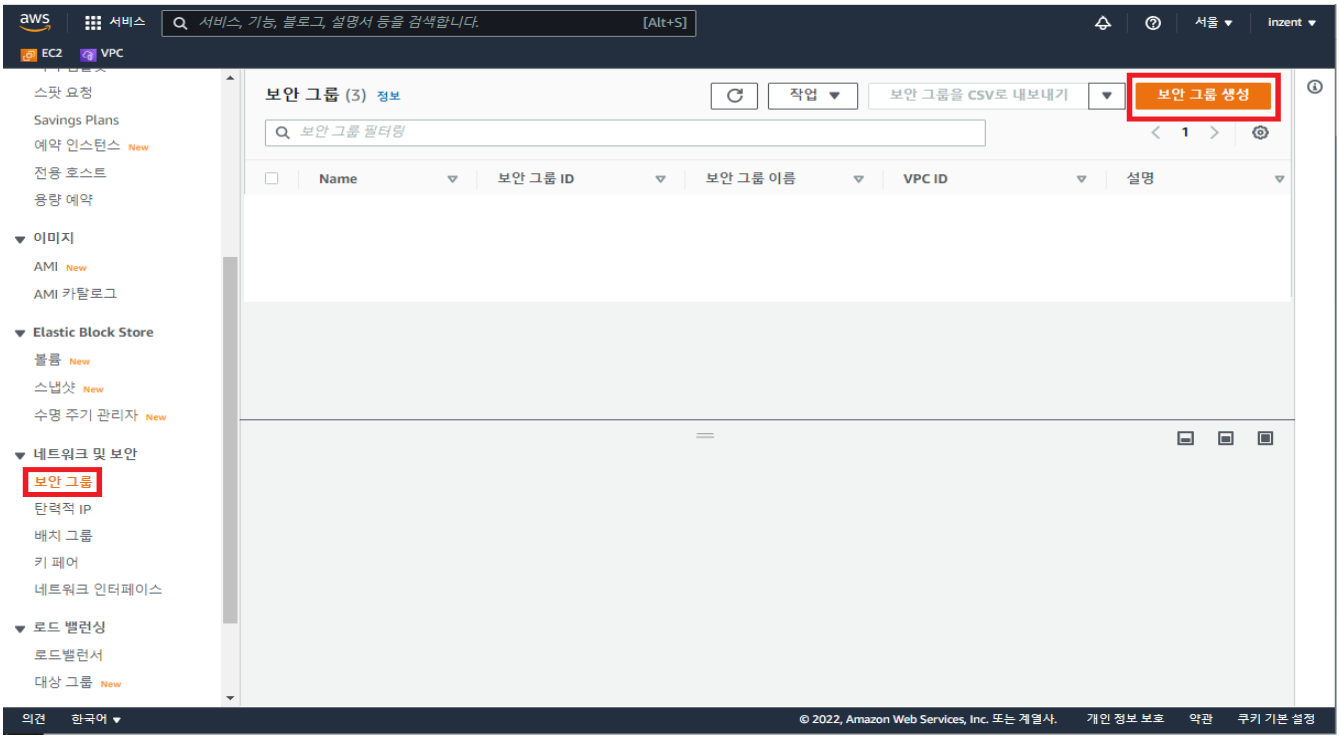
9. On the details screen Connections tab, choose Edit. Select the connection checkbox for [MyData Shield Subnet] to associate with the network ACL, then choose Save

3.1.3 Create Security Group

You need to set up security groups for the MyData Shield and the user, MyData Storage devices to communicate with each other

- Create MyData Shield Securiy Groups

1. Access the AWS EC2 Mangement Console.
2. Select [NETWORK & SECURITY > Securiy Groups] and click on the Create Security Group button



3. As shown below, add a new rule to the Inbound rule..

Menu	Input Value
Security Group name	sMyData Shield-SSH SG
Description	MyData Shield-SSH SG
VPC	Choose the same VPC as the device used by the operator
Type	SSH
Protocol	TCP
Port Range	22
Source	User IP or Device IP to be used by user

Menu	Input Value
Security Group name	MyData Shield-DB SG
Description	MyData Shield-DB SG
VPC	Choose the same VPC as the device used by the operator
Type	PostgreSQL
Protocol	TCP
Port Range	5432
Source	MyData Storage device IP

4. Set outbound rules for devices that will operate data

Menu	Input Value
Security Group name	MyData Shield-DB SG
Description	MyData Shield-DB SG
VPC	Choose the same VPC as the device used by the operator
Type	PostgreSQL
Protocol	TCP
Port Range	5432
Source	Device IP from which you want to view data

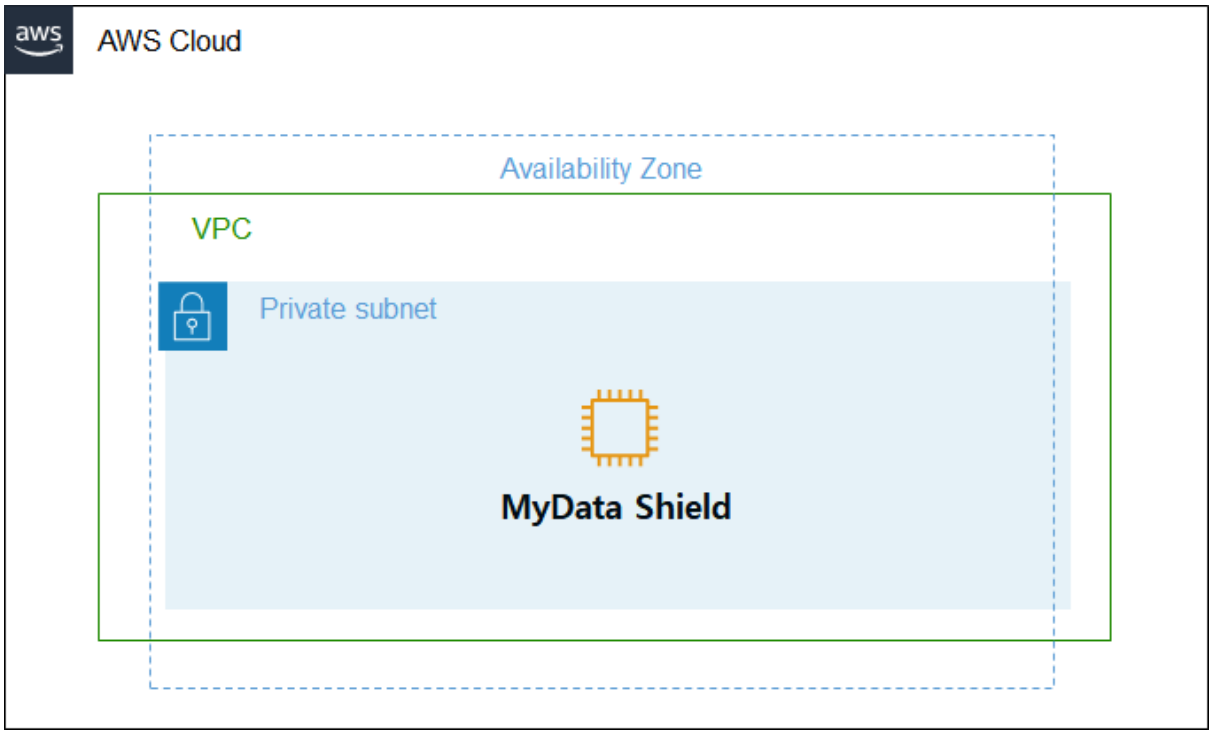
5. Add a Name Tage to [Tags] as follows

Menu	Input Value	Menu	Input Value
Tag	name	Description	Tagging to identify groups
Value	MyData Shield-SSH SG		

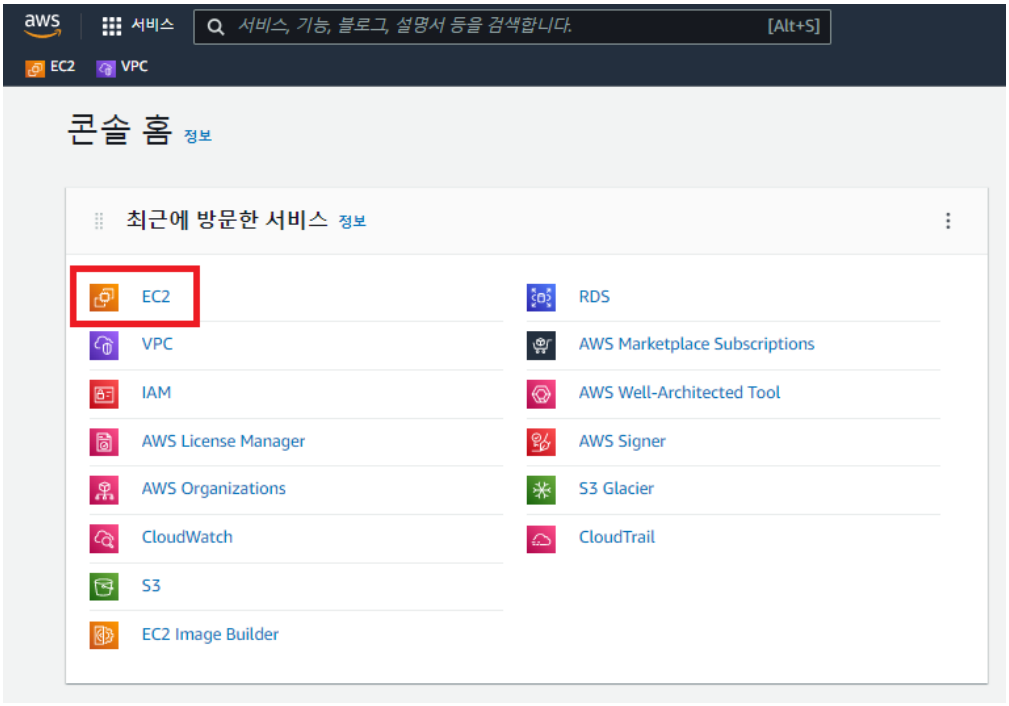
Menu	Input Value	Menu	Input Value
Tag	name	Description	Tagging to identify groups
Value	MyData Shield-DB SG		

3.1.4 Create Instance

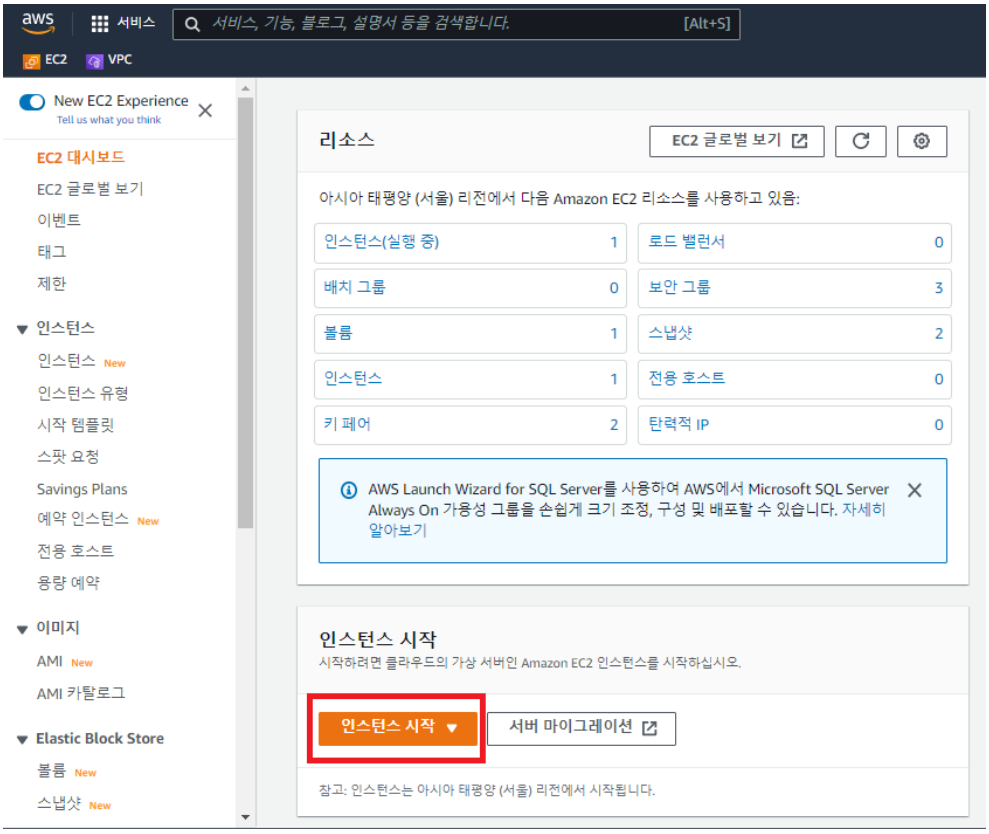
- MyData Shield AMI creates an instance by sharing AMI from vendor



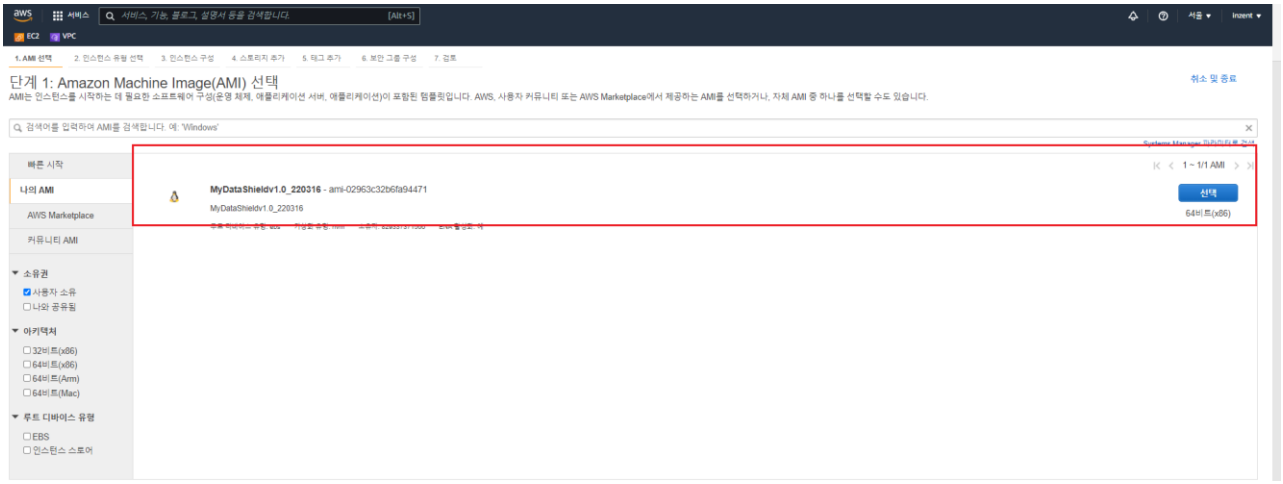
1. Log in to the AWS EC2 console



2. Click Launch Instance



3. Create a shared AMI instance



4. Choose an instance type

   - Select by referring to [2.3 Sizing].

5. Configure Instance Details

Menu	Input Value
Configure Instance Details	● Number of Instance : 1 ● Network :For information about VPC, see the following link: https://docs.aws.amazon.com/ko_kr/vpc/latest/userguide/working-with-vpcs.html ● Subnet : For information about Subnet, see the following link: https://docs.aws.amazon.com/ko_kr/vpc/latest/userguide/working-with-vpcs.html ● Auto-Assign Public IP : User subnet setting(Enable) ● IAM Role : None
ETC	If not informed, Select Default Option

6. Add storage

   - Select by referring to [2.3 Sizing].

7. Add tags

   - Tagging MyData Shield EC2 Instance

Menu	Input Value	Menu	Input Value
Tage key	Name	Description	Tagging to identify assets
Value	MyData Shield		

8. Configure Security Group

   - 기존 보안그룹 선택 : [MyData Shield-SSH SG], [MyData Shield-DB SG]

9. Select an existing key pair or create a new key pair

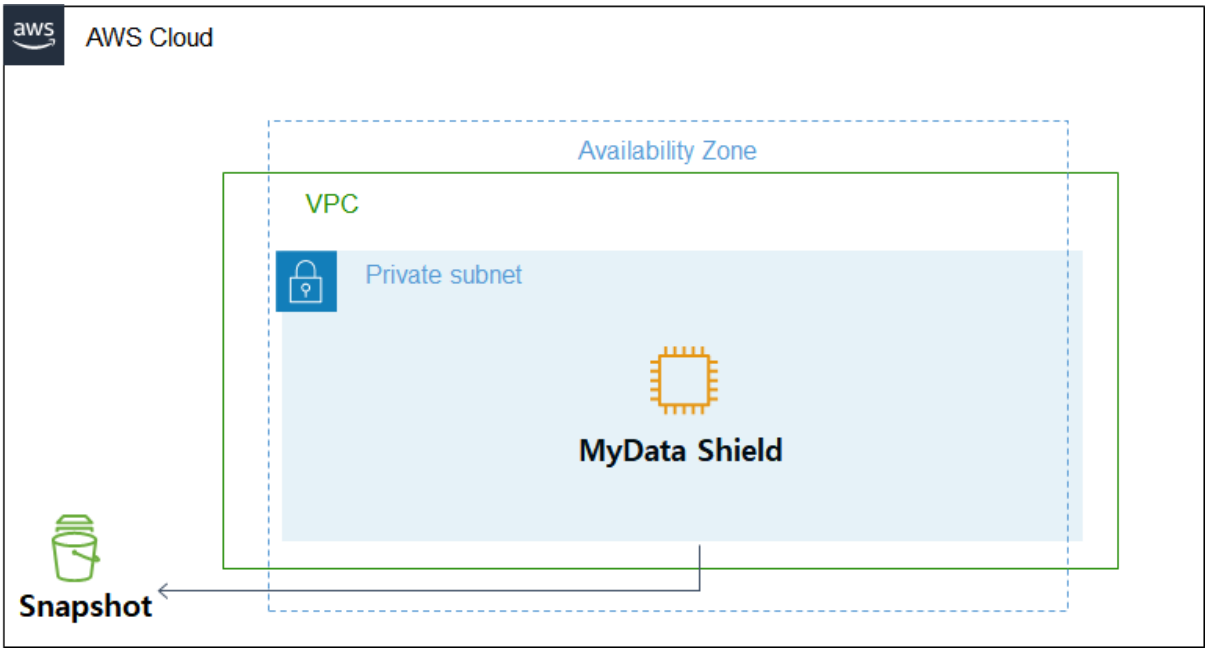
- If there is an existing key pair to use, start the instance by selecting the existing key pair

- If an existing key pair does not exist, create a key pair that can be connected via SSH by creating a new key pair

4. Operational Guidance

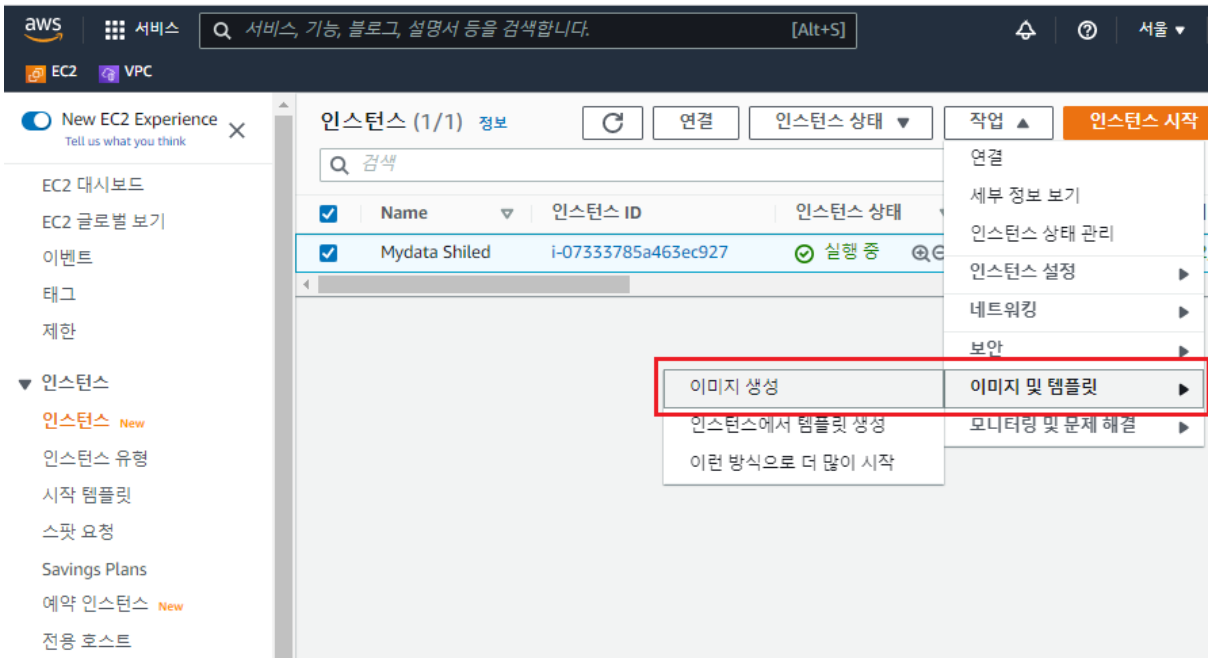
4.1 Support for MyData Shield Backup and Restore in AWS

4.1.1 MyData Shield Backup and Restore



A. Backup(Snapshot)

1. Create an AMI image of MyData Shield

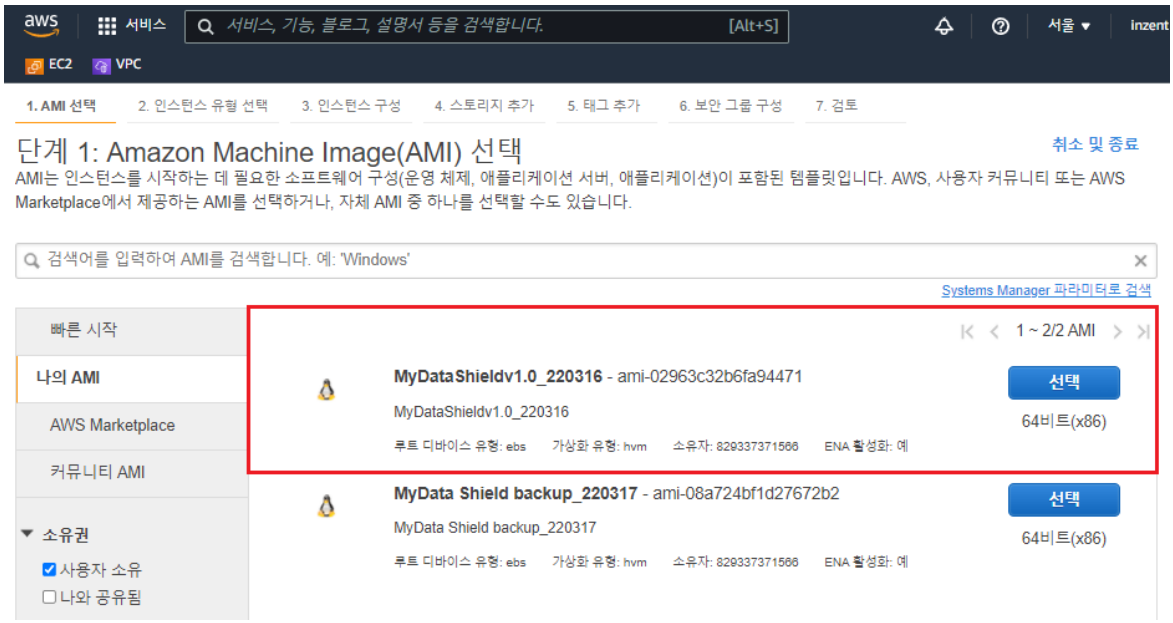


2. Create Image

Menu	Input Value
Image name	MyData Shield backup
Image description	MyData Shield backup
No reboot	Uncheck
Instance Volumes	Default configure

B. Restore

1. Choose an Amazon Machine Image (AMI)

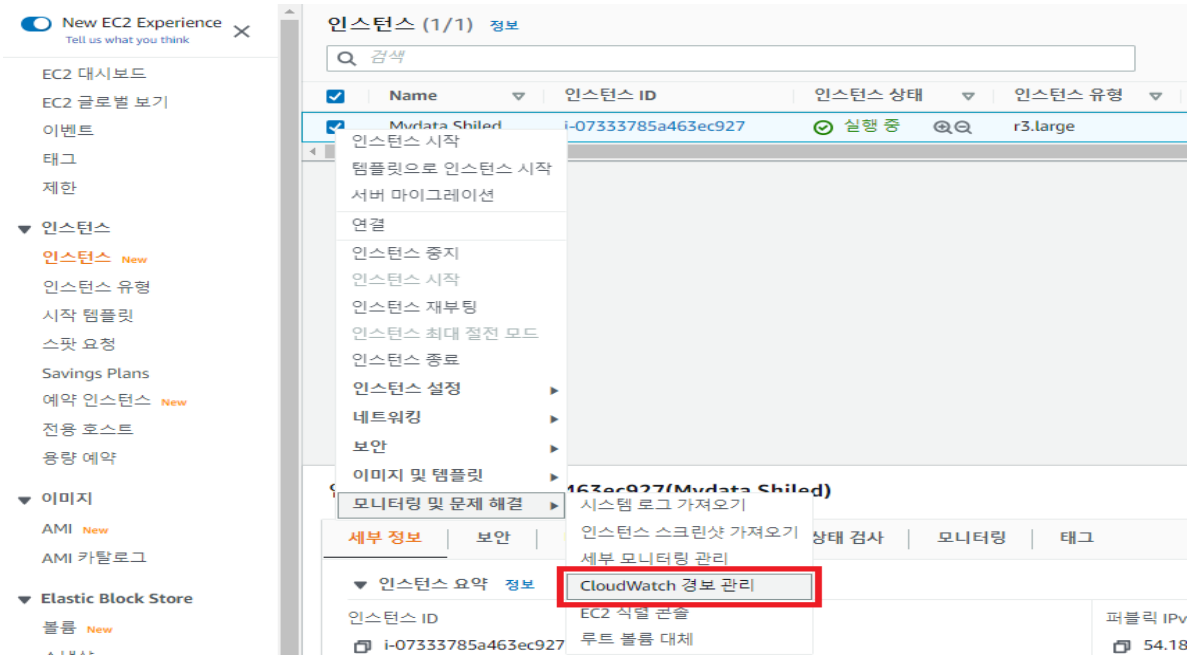


2. Select to create AMI(snapshot), see **[3.1.4 create Instance]**

4.2 MyData Shield Health Check with CloudWatch

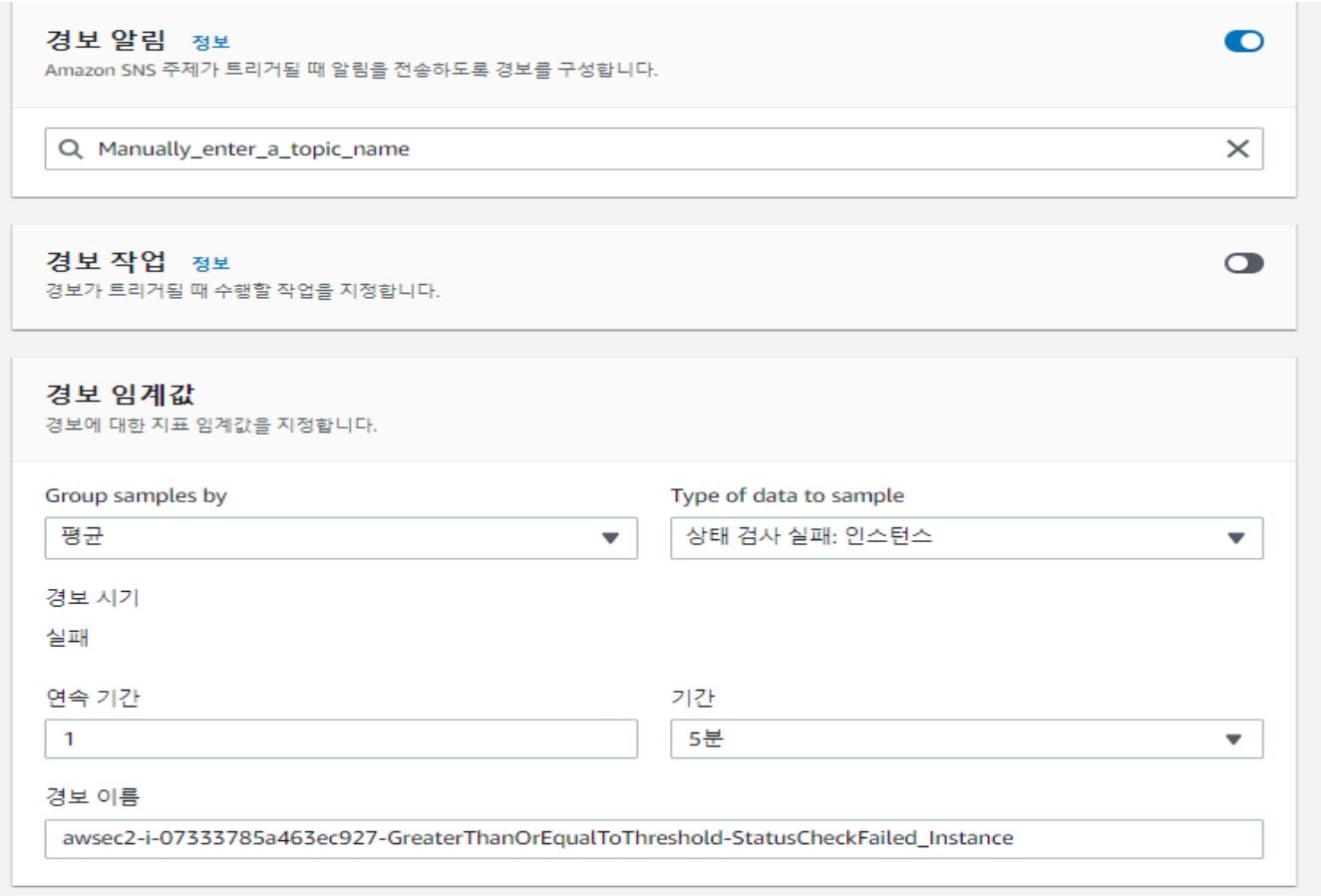
Optional : Intergrate with CloudWatch to support MyData Shield health check

1. Create an alarm on the deployed MyData Shield instance.



2. Create an alarm after setting the policy in the Create Alarm tab as shown below.

Item	Input Value	Remarks
Alarm notification	Alarm to SNS	
Alarm threshold	Type of data to sample : Status Scan failed : instance Continuos : 1	



4.3 Database Credentials

DB information of MyData Shield can be provided through the link **[2.2]**.

- **Optional:** You Can manage MyData Shield's database credentials using Secret Manager.

  Please get started via the link below.

  https://docs.aws.amazon.com/ko_kr/secretsmanager/latest/userguide/managing-secrets.html

4.4 Routine Maintenance

Maintenance fees are determined by contract policy and include latest release development and upgrade services

Details of maintenance and technical support may vary under additional agreements Maintenance is broadly divided into :

- Regular inspection : Conduct emergency inspections according to the maintenance contract
- Emergency inspection : Conduct emergency inspection according to the maintenance contract.

Maintenance scope :

- Check the solution problem
- Patches and upgrades

4.5 Emergency Maintenance

4.5.1 Startup process

- User-Startup process

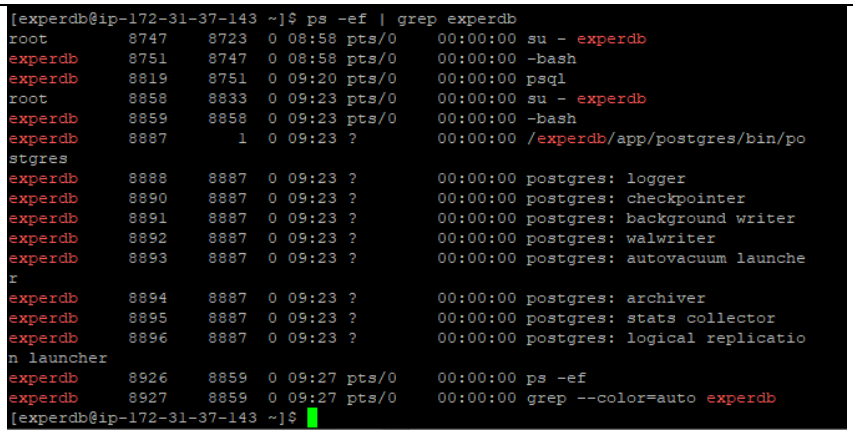
A. Start

Order	Description	Command
1	[mydata] SSH login with your account	-
2	Check Config.py settings	cat /home/mydata/MyData-Shield-Batch/MyData-Shield/Config.py
3	Modify Config.py settings using Nano script editor	source /home/mydata Config.sh
4	Log in with your [experdb] account	su - experdb
5	Start eXperdb	pg_ctl start

4.5.2 Health Check

- Check ExperDB process

Order	Description	Command
1	SSH login with [experdb] account	-
2	eXperdb process check	ps -ef | grep experdb
Normal after entering the command :		



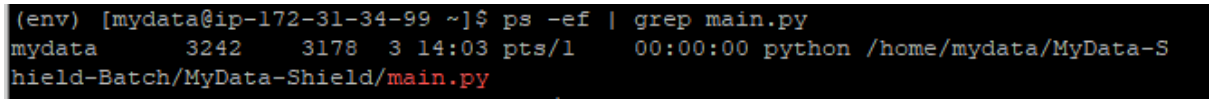
Abnormal after entering the command :

- Process not detected

- Check MyData Shield process

Order	Description	Command
1	SHH login with [mydata] account	-
2	MyData Shield process check	ps -ef | grep main.py

Normal after entering the command :



Abnormal after entering the command :

- Process not detected

4.5.3 Type of MyData Shield failures

- Insufficient service resources - EBS capacity

4.5.4 Recovery procedure for MyData Shield failure

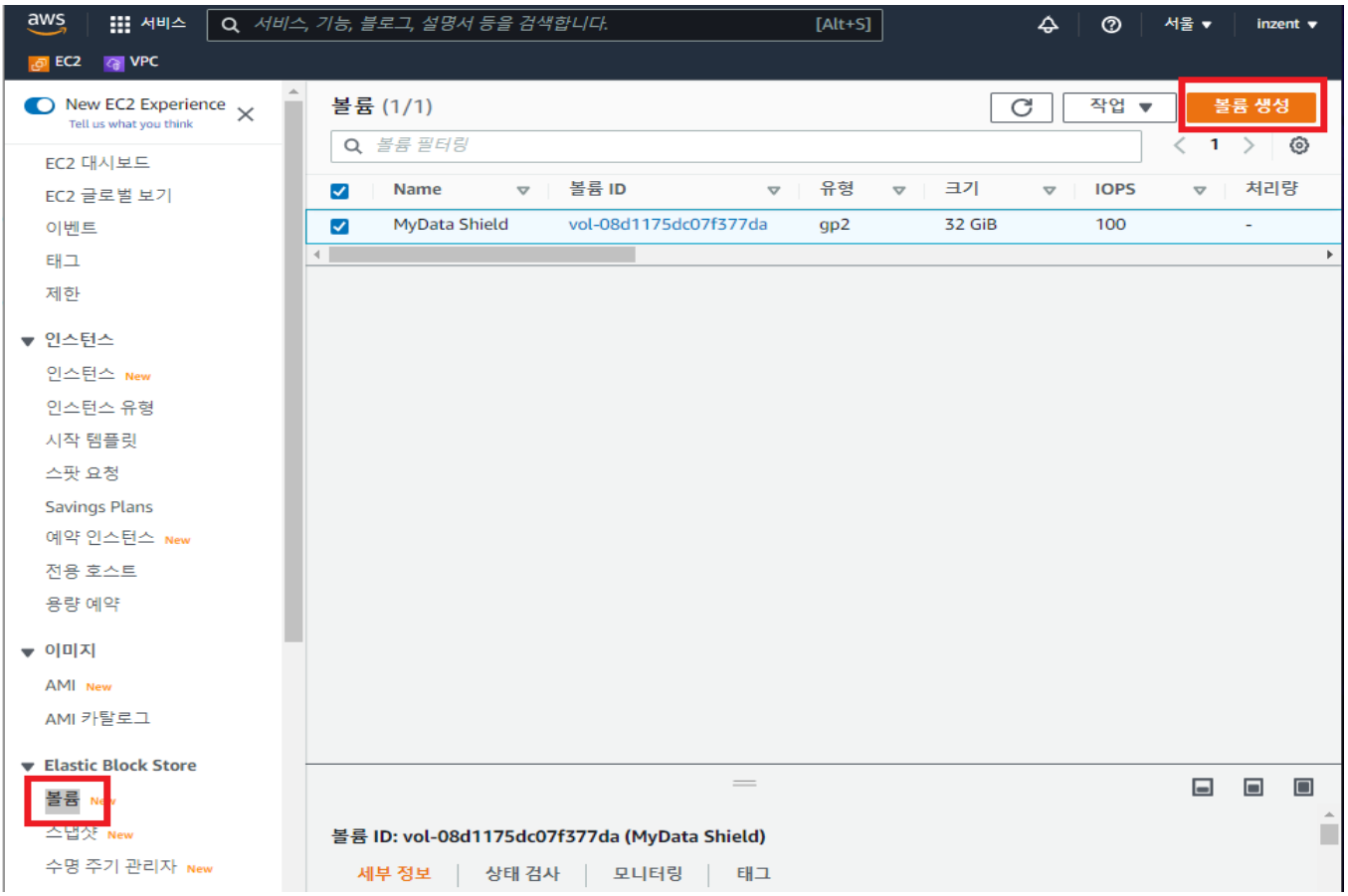
1) Insufficient service resources - EBS capacity

1. when capacity is 100%

A. Delete unnecessary data after checking MyData that has completed pseudonym/anonymity processing in eXperDB

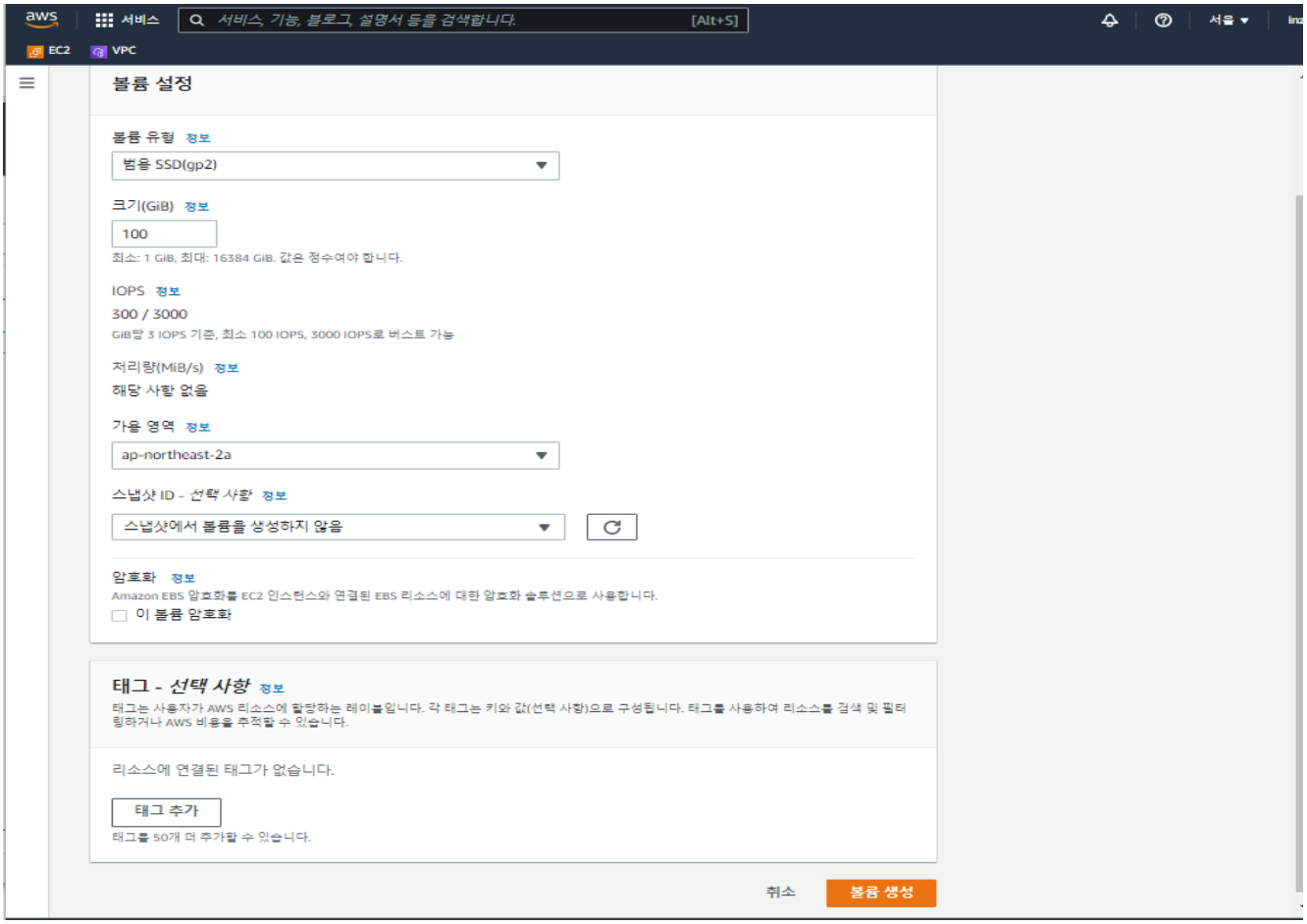
B. When additional EBS capacity expansion is required

- AWS EC2 Console > Elastic Block Store > Volumes > [MyData Shield] Instance > Click Create Volume



- The volume setting is set according to the size of MyData to be processed in the future.

Menu	Value
Volume type	General Purpose SSD(gp2)
Size(GIB)	Size of MyData to be processed



C. MyData Shield Instance reboot

4.5.5 Recovery procedure when MyData Shield recovery fails

- MyData Shield

In case of MyData Shield recovery failure, you can choose how to reinstall it depending on whether you have a snapshot or not.

A. Recreate AMI with snapshot of existing MyData Shield

B. Reinstall Manger according to [3.1.4 Create Instance] procedure

4.5.6 RTO

When MyData Shield fails, it takes at least 10 to 30 minutes to recreate an AMI with a snapshot of the previously installed instance and install it.

5. Solution operation

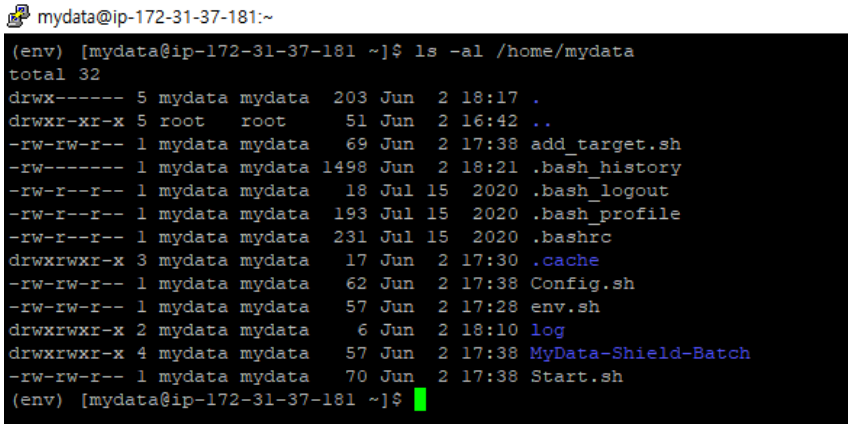
Users are recommended to have knowledge of the Linux CLI environment and experience with Python scripts

5.1 How to set

1. After setting all security groups and running instances, ssh access as **ec2-user**
2. **[su - experdb]** login with SSH account -> **[pg_ctl start]** Run eXperDB via command
3. The location of the initial run script is in **[/home/mydata]**

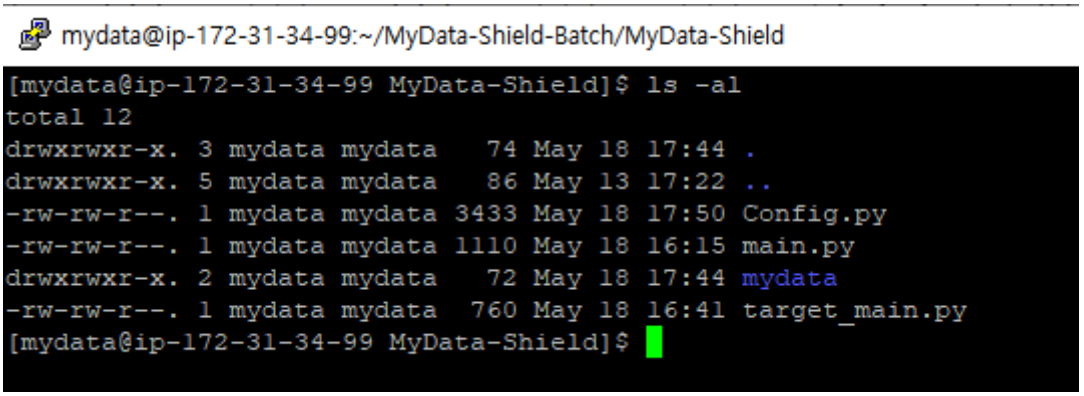
- user **mydata** SSH login via **[su – mydata]** command

- **[ls -al /home/mydata]** check run script and log folder



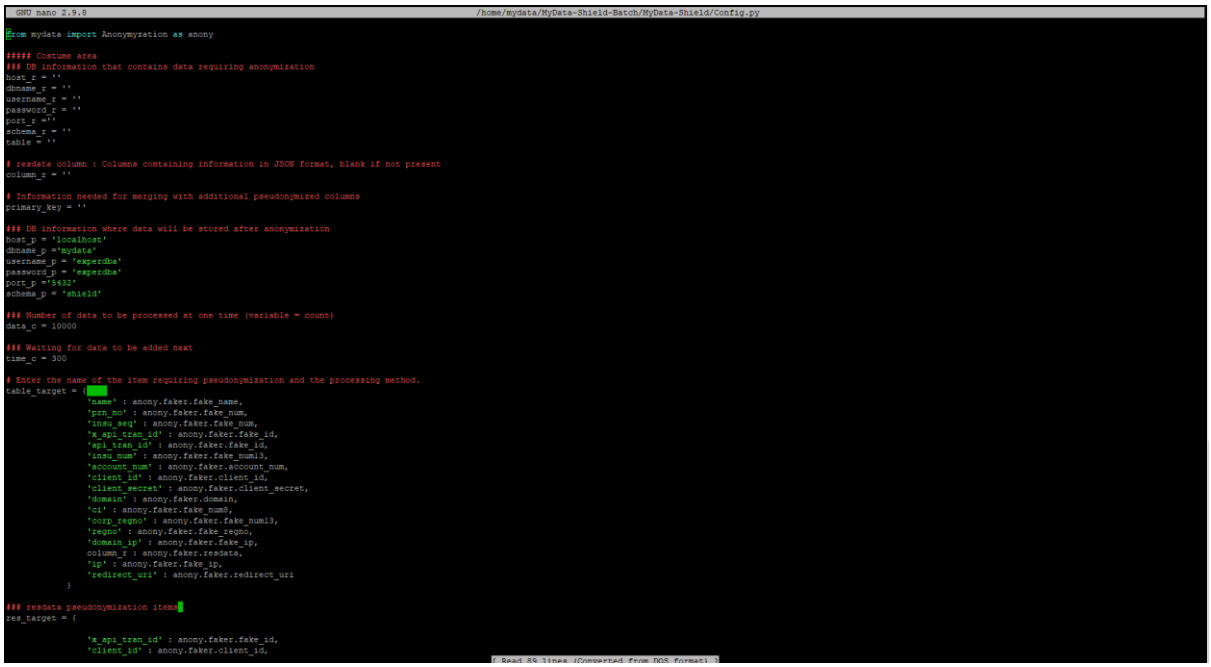
- **[ls -al /home/mydata/MyData-Shield-Batch/MyData-Shield]** check python module

(main.py, **target_main.py**, **Config.py**)



4. Go back to **mydata** home and user nano script editor to enter comment information and DB information according to the tables below and 'item name' to be pseudonymized **(MyData Shield only supports PostgreSQL TO PostgreSQL)**

- **[cd]** -> **[source /home/mydata/Config.sh]**



- Vi script editor is also available

- sudo vi /home/mydata/MyData-Shield-Batch/MyData-Shield /Config.py

5. How to set up a config file

A. Database information for which pseudonymization is desired

Item name	Value
host_r	Hostname of the DB where the data to be processed exists
dbname_r	DBname where data to be processed exists
user_r	MyData DB username
password_r	MyData DB Password
port_r,	MyData DB port
table	Table names requiring aliasing
column_r	Column name where data exists in Json type in the table where the data to be processed exists
primary_key	Primary key of table to be aliased

- DB info ex)

```
from mydata import Anonymyzation as anony

##### Costume area
### DB information that contains data requiring anonymization
host_r = '182.252.xxx.xx'
dbname_r = 'mydata_dbname'
username_r = 'mydata_username'
password_r = 'mydata_password'
port_r = '5433'
schema_r = 'mydata'
table = 'mydata'

# resdata column : Columns containing information in JSON format, blank if not
column_r = 'res_data'

# Information needed for merging with additional pseudonymized columns
primary_key = 'seq_no'
```

B. About MyData Shield instance eXperDB

Item name	Value
host_p	MyData Shield Hostname
dbname_p	DB name to sav e MyData
user_p	eXperDB username
password_p	eXperDB Password
port_p	MyDataShield Postgresql port

schema_p	Schema name to classify pseudonymized MyData

C. Set the number of data processing

Item name	Value
data_c	The number of data to ne processed by dividing the read data
time_c	Period to check if new data has been added after completing all pseudonymization processes

D. Settings for items and methods of pseudonymization

- Setting ex)

```
table_target = {
                  'item_name' = pseudonymization techniques,
                  column_r : anony.faker.resdata
                 }
 res_target = {
               'item_name' = pseudonymization techniques,
               'item_name' = pseudonymization techniques
              }
```

- Setting Description

Item name	Value
table_target	Settings for columns that require alias processing in MyData table
res_target	Setting of 'item name' that requires pseudonymization by referring to MyData API standard (setting of 'item name' existing in Json in MyData column)

- Setup pseudonymization

Pseudonymization techniques	How to use pseudonymization technique
Faker	• anony.faker.fake_name (홍길동 -> 최지현) • anony.faker.fake_character (private_info - > OMeJZiramzaABneExVh) • anony.faker.fake_num (num_faker shows 10 random digits) • anony.faker.fake_id (secret_info -> kNTHhhYZkoKHahL5u7I2D) • anony.faker.account_num (35600812438931 -> 3563368404692281) • anony.faker.client_id (20-50 random character generation) • anony.faker.client_secret (random md5(), ex) 3asdk1j20asdkl1 -> 8451f734df34b7f66dd5fd820383f122 • anony.faker.domain ('naber.com' -> 'hangim.jusighoesa.baggimno.com')

	• anony.faker.fake_num8 (num_faker shows 8 random digits) • anony.faker.fake_num13 (num_faker shows 13 random digits) • anony.faker.fake_regno (000-00-0000 -> 432-71-3211) • anony.faker.fake_ip (192.0.0.1 -> 104.67.80.197) • anony.faker.redirect_uri (http://naber.com -> http://yuhanhoesa.kr/) • anony.faker.sha256 (private_code -> 2d78ba29e9118929421ab2ab67db91208a770738aa0fad1b33e96cbbd092d042)
Masking	• anoy.Masking.p_data (abcde1234 -> abcd*****) • anony.Masking.p_name (홍길동 -> 홍**) • anony.Masking.p_car_num (차량 09 1234 -> 차량 09 ****) • anony.Masking.p_phone (01000000000 -> 0100000****) • anony.Masking.p_num (3500812443222278 -> 3500****4322****) • anony.Masking.address (인천광역시 중구 봉은사 14 거리 -> 인천광역시 중구) • anony.Masking.hashText (salt+ sha25(), ex) 홍길동 -> 831eec5830cb2b627d5829e9b61a8789d66e23c7b5a86f07e787a5ad77d4d0d1)

- Do not delete or modify the information of column_r in the yellow box

```
# Enter the name of the item requiring pseudonymization and the processing method.
table_target = {
                'name' : anony.faker.fake_name,
                'prn_no' : anony.faker.fake_num,
                'insu_seq' : anony.faker.fake_num,
                'x_api_tran_id' : anony.faker.fake_id,
                'api_tran_id' : anony.faker.fake_id,
                'insu_num' : anony.faker.fake_num13,
                'account_num' : anony.faker.account_num,
                'client_id' : anony.faker.client_id,
                'client_secret' : anony.faker.client_secret,
                'domain' : anony.faker.domain,
                'ci' : anony.faker.fake_num8,
                'corp_regno' : anony.faker.fake_num13,
                'regno' : anony.faker.fake_regno,
                'domain_ip' : anony.faker.fake_ip,
                column_r : anony.faker.resdata,
                'ip' : anony.faker.fake_ip,
                'redirect_uri' : anony.faker.redirect_uri
            }

### resdata pseudonymization items
res_target = {

                'x_api_tran_id' : anony.faker.fake_id,
                'client_id' : anony.faker.client_id,
```

- More details (https://github.com/jw0245/MyData-Shield)

5.2 How to run

1. After performing the **[5.1 How to set]** process, check the user login **[mydata]**.
2. Use the **[cd /home/mydata]** command to move to the location where the script is located
3. For normal module execution, run the virtual environment through the **[source env.sh]** command in the table below
4. You can perform alias processing through the **[source Start.sh]** command in the table below, and optionally additional pseudonymization through the **[source add_target.sh]** command

Command	Description
source env.sh	**(Essential)** Running a virtual environment to run modules
source Start.sh	After pseudonymizing the data in MyData Store, the process that is stored in eXperDB of the instance is executed
source add_target.sh	**(Optional)** There is no newly added data, but when an 'item name' for which you want to process a pseudonym is additionally set among 'item name' of MyData(Json type), The process storing data corresponding to the 'item name' in eXperDB after additional pseudonymization

5. Check the result

You can use the following command to connect to the DB by connecting to the DB by connecting to the experdb user, and it can be used in the same way as the postgresql syntax

- **[su – experdb] -> [psql -U [username] -d [DBname]]**

- Sample Data

id	resdata
ANZW4BZHK5R	{'name': '권상훈', 'age': 41, 'email': 'caeweon45@live.com', 'phone': '061-786-0116', 'address': '부산광역시 영등포구 도산대4거리'}
4COHGZPNSOH	{'name': '손성지', 'age': 49, 'email': 'dgim@jigang.com', 'phone': '02-1562-9960', 'address': '부산광역시 성동구 오금095길 (유진최읍)'}
WQ3ZSG5PQC5	{'name': '손은정', 'age': 84, 'email': 'gyu@naver.com', 'phone': '010-0974-3608', 'address': '경상남도 구리시 영통대2가'}
CZOCC8N4IRZ	{'name': '최준호', 'age': 99, 'email': 'zgim@hotmail.com', 'phone': '010-9106-4924', 'address': '충청남도 성남시 수정구 양재천862로 (지영이읍)'}
UZ398HLPQIT	{'name': '박재현', 'age': 41, 'email': 'to@nate.com', 'phone': '033-183-6997', 'address': '세종특별자치시 영등포구 가락4길'}
TWF7DZU9SOH	{'name': '강성미', 'age': 39, 'email': 'yeeuni@ani.net', 'phone': '043-334-2380', 'address': '서울특별시 중랑구 잠실가'}
6EKDR8C9SC4	{'name': '이승현', 'age': 96, 'email': 'iyeonghwan@nate.com', 'phone': '051-623-5262', 'address': '충청남도 여주시 서초중앙길 (영순강이읍)'}
ZRWZOZ684DS	{'name': '김미정', 'age': 34, 'email': 'oan@live.com', 'phone': '055-787-6810', 'address': '제주특별자치도 진천군 잠실822길 (경자최이면)'}
83ZDI1A86A7	{'name': '이상현', 'age': 36, 'email': 'seoyeweon@naver.com', 'phone': '070-7899-5171', 'address': '충청남도 영동군 학동가 (정숙이이읍)'}
W5OEY91ZZ3T	{'name': '장민서', 'age': 24, 'email': 'bagyeonghwan@dreamwiz.com', 'phone': '031-972-8403', 'address': '충청남도 춘천시 반포대길'}

(10개 행)

- Processed Data

id	resdata
ANZW4******	{'name': '박민서', 'age': '4*', 'email': 'caeweon45*********', 'phone': '061-78******', 'address': '부산광역시 영등포구'}
4COHG******	{'name': '서채원', 'age': '4*', 'email': 'dgim@ji********', 'phone': '02-156******', 'address': '부산광역시 성동구'}
WQ3ZS******	{'name': '이명숙', 'age': '8*', 'email': 'gyu@na*******', 'phone': '010-09*******', 'address': '경상남도 구리시'}
CZOCC******	{'name': '이지영', 'age': '9*', 'email': 'zgim@hot********', 'phone': '010-91*******', 'address': '충청남도 성남시'}
UZ398******	{'name': '송정순', 'age': '4*', 'email': 'to@na******', 'phone': '033-18******', 'address': '세종특별자치시 영등포구'}
TWF7D******	{'name': '김미정', 'age': '3*', 'email': 'yeeuni@*******', 'phone': '043-33******', 'address': '서울특별시 중랑구'}
6EKDR******	{'name': '문준영', 'age': '9*', 'email': 'iyeonghwa**********', 'phone': '051-62******', 'address': '충청남도 여주시'}
ZRWZO******	{'name': '홍미영', 'age': '3*', 'email': 'oan@li******', 'phone': '055-78******', 'address': '제주특별자치도 진천군'}
83ZDI******	{'name': '박현준', 'age': '3*', 'email': 'seoyeweon**********', 'phone': '070-78*******', 'address': '충청남도 영동군'}
W5OEY******	{'name': '이승현', 'age': '2*', 'email': 'bagyeonghwan*************', 'phone': '031-97******', 'address': '충청남도 춘천시'}

* For the syntax of postgresql, see https://www.postgresqltutorial.com/

5.3 Key management

A. EBS Key Management

You can use AWS KMS keys to support EBS encryption if needed

Check out EBS Volume Encryption Management via the link below :

https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/EBSEncryption.html

B. Secret Manager Key Management

If necessary, AWS KMS keys can be used to support encryption of DB information through Secret Manager.

Check it out via the link below :

https://docs.aws.amazon.com/secretsmanager/latest/userguide/security-encryption.html

5.4 Patches and updates management

Inquiries about patches/updates can be made through the link below and will be handled separately according to the contract terms

URL : http://www.inzent.com/board/board.php?bo_table=faq&pageName=main

6. Support

6.1 Technical support

The technical support service is provided only for the features specified in the document.

Technical support covers the following areas:

A. Installation support: Deployment guide provided, source provided through Github

➔ Github : https://github.com/jw0245/MyData-Shield

B. Customization support

C. Provides additional rea-time RestAPI function

D. Demo page for pseudonymization function

➔ https://mydapi.inzent.com/shield

Technical support contacts :

E. URL : http://www.inzent.com/board/board.php?bo_table=faq&pageName=main

F. TEL : 02-787-3600

- E-mail : info@inzent.com

6.2 Support Costs

- Free Tier

✓ Free

- Technical Service Pack

- 15 Hour : 2,250,000 won
- 30 Hour : 4,000,000 won
- 60 Hour : 7,500,000 won
- 120 Hour : 14,000,000 won

Provided upon contract :

- Real-time Rest API function added
- Customization support (man-hours calculated separately)

6.3 SLA

- Free Tier : Source provided, deployment guide
- Technical Service Pack : At the request of customer, technical support corresponding to the standard time is provided