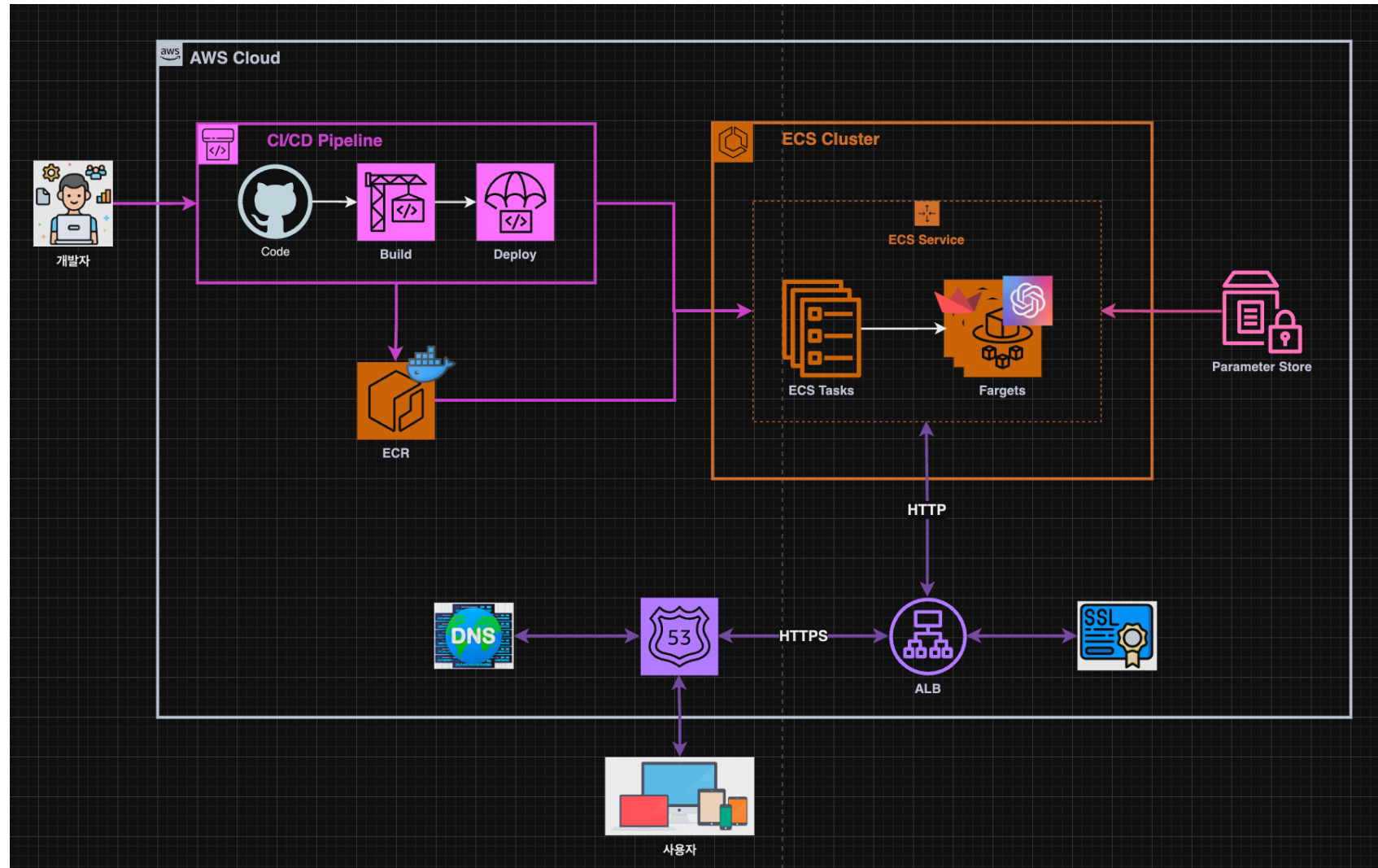


Architecture



Parameter Store

- AWS Systems Manager의 기능인 Parameter Store는 구성 데이터 관리 및 암호 관리를 위한 안전한 계층적 스토리지를 제공합니다.
- 암호, 데이터베이스 문자열, Amazon Machine Image(AMI) ID, 라이선스 코드와 같은 데이터를 파라미터 값으로 저장할 수 있습니다.

파라미터 스토어의 특징

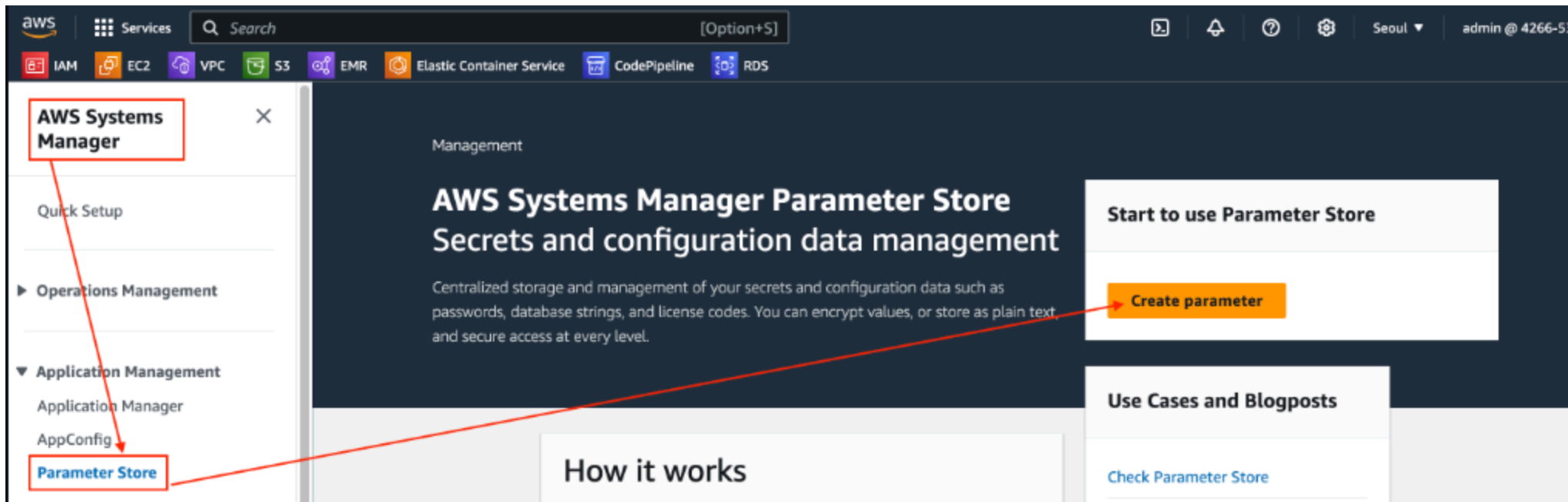
- 무료다.
- 키-값 쌍으로 값을 저장한다.
- AWS Key Management Service(KMS)를 이용해 암호화된 값을 저장할 수 있다.
- SecureString 파라미터 생성 시 Parameter Store에서 비용이 부과되지 않지만 AWS Key Management Service(KMS) 암호화 사용에 대한 **요금**이 적용됩니다.
- IAM을 이용해 일부 사용자만 접근할 수 있도록 설정할 수 있다.
- 값에 대한 변경 이력까지 저장하고 있다.

파라미터 생성시 제약조건

- **대/소문자 구분:** 파라미터 이름은 대/소문자를 구분합니다.
- **공백:** 파라미터 이름에는 공백이 포함될 수 없습니다.
- **유효한 값:** 파라미터 이름은 a-zA-Z0-9_-과 같은 기호와 문자로 구성될 수 있습니다.
 - 또한 슬래시 문자(/)는 파라미터 이름의 계층 구조를 나타내는 데 사용됩니다. 예 :
`/Dev/Production/East/Project-ABC/MyParameter`
- **길이:** 생성할 수 있는 파라미터 이름의 최대 길이는 1011자입니다.
 - 여기에는 `arn:aws:ssm:us-east-2:111122223333:parameter/` 와 같이 지정한 이름 앞에 오는 ARN의 문자가 포함됩니다.

파라미터 생성하기

단계1: 파라미터 스토어 생성




단계2: 파라미터 이름

Parameter details

Name

Q /TEST/CICD/STREAMLIT/GROQ_API_KEY



When naming a parameter, you can use forward slashes (/) to organize it into a hierarchy. [Learn more about hierarchies](#) 

Description — *Optional*

Groq Api Key

단계3: Tier > Standard 선택

Tier



Parameter Store offers standard and advanced parameters.

☒ Standard

Store up to 10,000 standard parameters. Store parameter values up to 4 KB. Parameter policies and sharing with other AWS accounts are not available. No additional charge.

☐ Advanced

Store up to 100,000 advanced parameters. Store parameter values up to 8 KB. Add parameter policies. Share with other AWS accounts. Charges apply.

 Standard parameters cannot be shared with other AWS accounts. [Learn more](#) 

단계4: 파라미터 타입 선택

Type

- ☒ **String**
Any string value.
- ☐ **StringList**
Separate strings using commas.
- ☐ **SecureString**
Encrypt sensitive data using KMS keys from your account or another account.

단계5: API Key 붙여넣기

Data type

text

Value

gsk

Maximum length 4096 characters.

단계7: Create parameter

Tags — *Optional*
You can use tags to organize and restrict access to your parameter.

Add tag

Cancel

Create parameter

단계8: 파라미터 생성 확인

The screenshot shows the AWS Parameter Store console. At the top, a green banner displays the message "Your parameters were successfully deleted!". Below this, the breadcrumb navigation shows "AWS Systems Manager > Parameter Store". The "My parameters" tab is selected. The "My parameters" section includes a search bar, a filter bar with the text "Name: contains: /TEST/CICD/STREAMLIT/GROQ_API_KEY", and a "Clear filters" button. To the right of the search bar are buttons for "View details", "Edit", "Delete", and "Create parameter". Below the filter bar is a table with the following columns: Name, Tier, Type, and Last modified. The table contains one row with the parameter name "/TEST/CICD/STREAMLIT/GROQ_API_KEY", a "Standard" tier, a "String" type, and a last modified date of "Sat, 12 Apr 2025 11:04:19 GMT".

aws Services [Alt+S] Asia Pacific (Seoul) admin @ 4266-5374-2

IAM EC2 VPC S3 EMR Elastic Container Service CodePipeline Aurora and RDS CloudWatch Amazon EventBridge Lambda

☰ Your parameters were successfully deleted!

AWS Systems Manager > Parameter Store

My parameters Public parameters Settings

My parameters [Refresh] View details Edit Delete Create parameter

Search

Name: contains: /TEST/CICD/STREAMLIT/GROQ_API_KEY Clear filters < 1 > [Settings]

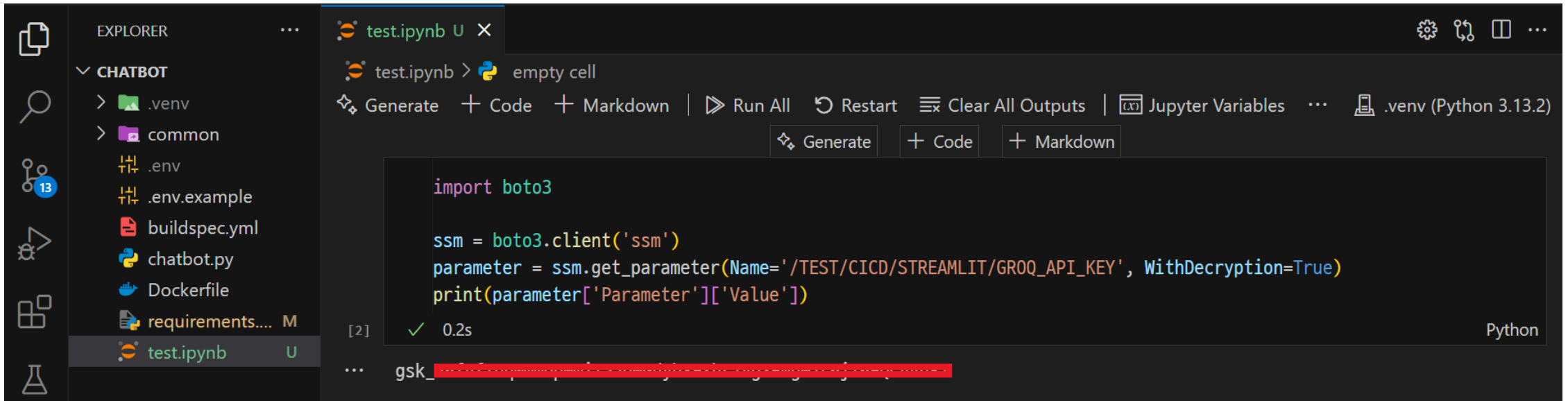
<input type="checkbox"/>	Name	Tier	Type	Last modified
<input type="checkbox"/>	/TEST/CICD/STREAMLIT/GROQ_API_KEY	Standard	String	Sat, 12 Apr 2025 11:04:19 GMT

단계9: CLI에서 파라미터 생성 확인

```
$ aws ssm get-parameter --name /TEST/CICD/STREAMLIT/GROQ_API_KEY
```

```
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> aws ssm get-parameter --name /TEST/CICD/STREAMLIT/GROQ_API_KEY
{
  "Parameter": {
    "Name": "/TEST/CICD/STREAMLIT/GROQ_API_KEY",
    "Type": "String",
    "Value": "gsk_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "Version": 1,
    "LastModifiedDate": "2025-04-12T20:04:19.458000+09:00",
    "ARN": "arn:aws:ssm:ap-northeast-2:123456789012:parameter/TEST/CICD/STREAMLIT/GROQ_API_KEY",
    "DataType": "text"
  }
}
```

단계10: Python에서 파라미터 생성 확인



The screenshot shows a Jupyter Notebook interface within a code editor. The left sidebar displays the file explorer with a project named 'CHATBOT' containing files like '.venv', 'common', '.env', '.env.example', 'buildspec.yml', 'chatbot.py', 'Dockerfile', 'requirements...', and 'test.ipynb'. The main editor area shows the 'test.ipynb' file with a single code cell. The code cell contains the following Python code:

```
import boto3

ssm = boto3.client('ssm')
parameter = ssm.get_parameter(Name='/TEST/CICD/STREAMLIT/GROQ_API_KEY', WithDecryption=True)
print(parameter['Parameter']['Value'])
```

Below the code, the execution output is shown: '[2] ✓ 0.2s' followed by a redacted string 'gsk_...' which is highlighted with a red box. The bottom right corner of the code cell indicates the language is 'Python'.