권한 제어 설정

EC2 IAM 정책 작성

- user01(sk407_002)에게 EC2 관련 권한 부여
- EC2 인스턴스 보기, 생성, 수정, 삭제 등 허용
- 정책 키워드 사용: Effect, Action, Resource

예:

```
{  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    }
  ]
}
```

S3 버킷 정책 작성 (업로드 허용)

- 특정 버킷에서 PutObject 권한 부여
- principal을 user01로 설정

예:

```
{  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::149491344651:user/sk407_002"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucketosaka00001/*"
    }
  ]
```

권한 경계 정책 작성 (특정 EC2 stop/start만 허용)

- EC2 전체 권한이 있더라도 경계를 통해 제한
- 특정 인스턴스 ID(sk407_002)에 대해서만 StartInstances, StopInstances 허용

예:

```
{  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
        "Action": [
          "ec2:StartInstances",
          "ec2:StopInstances"
        ],
 "Resource":"arn:aws:ec2:ap-northeast-3:149491344651:instance/i-0cc8d422467017770"
      }
   ]
}
```

결론 :

IAM 정책, S3 버킷 정책, 권한 경계 정책을 작성하여 user01 계정에 대한 접근 제어를 구현하였다.

먼저, IAM 정책에서는 EC2 서비스에 대한 광범위한 권한을 허용하도록 설정하였다. 이를 통해 user01은 EC2 인스턴스를 조회, 생성, 수정, 삭제할 수 있다. 정책 구문은 Effect, Action, Resource를 기반으로 작성되며 EC2 전체 자원(Resource: *)을 대상으로 ec2:* 액션을 허용하였다.

다음으로, S3 버킷 정책을 작성하여 user01이 지정된 버킷에 파일 업로드를 수행할 수 있도록 하였다. 이때 S3 버킷 정책은 리소스 기반 정책이므로 principal을 user01에게만 권한을 부여하였다. 구체적으로 s3:PutObject 권한을 허용하여 해당 버킷에 객체 업로드가 가능해졌다.

마지막으로, 권한 경계 정책을 통해 EC2 인스턴스 권한을 제한하였다. 비록 user01의 IAM 정책에서 EC2에 대한 모든 권한이 부여되어 있더라도, 권한 경계를 설정하여 특정 인스턴에 대해서만 StartInstances, StopInstances 액션을 허용하도록 제한하였다.