**□ 모듈프로젝트 산출물**

1. 4차산업기반 기본 기술 역량 강화

|  |
| --- |
| 모듈프로젝트 2-2 |

**2020년 08월 21일**

**클라우드 보안 융합 전문가**

이름 : 송정훈

1. **모듈프로젝트 내용**

**- AWS 아키텍처 구축**

**- RDS, Node.js, python 연동**

|  |
| --- |
| 프로젝트 산출물 (사진 첨부)  전체 아키텍처    CloudWatch  NewConnection으로 설정해서 count 3 이상 시 알람과 메일이 오도록 설정    이메일 확인    **CodeDeploy**    **Node, Python**  **Get**  nodejs    Python    **Post**  Nodejs    Python    **Delete**  nodejs    Python |
| AWS   1. 인스턴스 생성 후 인스턴스 안에 npm, node, express, mysql, body-parser를 설치합니다 2. 이미지 생성 후 만들어진 AMI 기반으로 템플릿을 생성합니다. 3. 템플릿 기반으로 Auto Scaling에 인스턴스 2개를 허용하고 생성합니다. 4. Auto Scaling그룹에 연결할 ELB를 만들어줍니다. 5. RDS생성 후 RDS와 연동할 database.js 파일을 만듭니다. database.js 안에 RDS 엔드포인트 및 mysql 아이디와 비밀번호를 입력합니다. 6. npm init을 해서 초기화 하고 index.js에 RDS와 연동되도록 dabases.js를 연결시킵니다. 7. S3를 생성하고 인스턴스 안에 CodeDeploy-agent를 설치하여 배포환경을 구성합니다. 8. CodeDeploy에 AllAtOnce로 설정해 한번에 배포되도록 했습니다. 9. CloudWatch와 AWS SNS를 ELB에 연결합니다. NewConnection으로 설정해서 count 3 이상 시 알람과 메일이 오도록 설정합니다. 10. 기존 RDS에서 읽기전용으로 slave를 만들고 인스턴스 안에 database-slave.js를 만들어 select는 slave에서, insert, update, delete는 Master에서 일어나도록 설정합니다.   Node, Python   1. Python에 Requests 를 임포트하여 인스턴스에 Request 요청을 보낼 수 있게 만듭니다. Pip3를 통해 설치합니다. 2. Python에서 node로 리퀘스트 보내는 get, post, delete를 작성합니다. 3. 인스턴스 안의 index.js에서 요청을 받을 수 있는 get, post, delete를 작성합니다. |

|  |  |
| --- | --- |
| 성명 | 프로젝트 후 소감 |
| 송정훈 | AWS의 다양한 기능을 사용한 아키텍처를 만들어 볼 수 있는 기회였습니다. 뿐만 아니라 node와 python을 활용하여 Hppt Method를 사용해볼 기회이기도 했습니다. 이러한 경험이 보안에 대해 배울 때 도움이 될 것이라 생각합니다.  3일간의 짧은 프로젝트였지만 시간을 느끼기 어려울 정도로 빨리 갔습니다. 나름 열심히 했다고 생각하지만 아쉬움이 많이 남습니다. 다음에는 더 나은 모습을 보여주고 싶은 생각이 든 프로젝트였습니다. |