Computer Network

Research

**목 차**

[1 [product]\_duplex에 대한 설명 4](#_Toc532306978)

[1.1 개요 4](#_Toc532306979)

[1.2 구성 방식: 4](#_Toc532306980)

[1.3 사용법 5](#_Toc532306981)

[1.3.1 .conf 파일 설정 5](#_Toc532306982)

[2 동작 방식 7](#_Toc532306983)

[2.1 프로세스 감시 & 재시작 7](#_Toc532306984)

[2.1.1 프로세스 감시 – 공유 메모리 확인 7](#_Toc532306985)

[2.1.2 프로세스 감시 – pid 파일 확인 7](#_Toc532306986)

[2.2 열린 FD의 비율 검사 8](#_Toc532306987)

**그림 목차**

[Figure 1‑1 watcher의 조망 4](#_Toc532306988)

[Figure 2‑1 pid 파일의 내용 7](#_Toc532306989)

[Figure 2‑2 열린 fd 개수의 비율을 구하는 수식 및 명령어 8](#_Toc532306990)

[Figure 2‑3 열린 FD의 비율이 지정 비율을 넘을 경우 기록되는 시스템 로그 9](#_Toc532306991)

# Layer

컴퓨터 네트워크의 Layer에 대해서 Top-Bottom 방식으로 설명합니다.

## OSI 7 Layer

* OSI : Open System Interconnection.
* 표준이 되는 모델이기는 하나 실제로 구현되는 예가 거의 없음

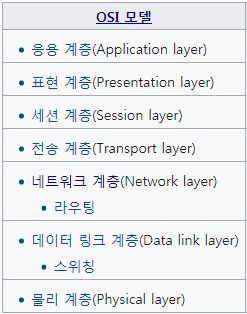


Figure 1‑1 OSI 7계층 모델

### 응용 계층

* 프로세스 간 정보 교환 / 파일 전송 등의 서비스 제공
* 데이터 전송 단위는 메시지(message)
* 사용자가 네트워크에 접근 가능하게 해줌

### 표현 계층

* // TODO

### 세션 계층

* // TODO

### 전송 계층

* // TODO

### 네트워크 계층

* // TODO

### 데이터 링크 계층

* // TODO

### 물리 계층

* // TODO

## TCP/IP 4 Layer

* 인터넷 개발 이후 꾸준히 표준화된 모델
* 실질적인 통신 프로토콜의 표준으로 사용



Figure 1‑2 TCP/IP 4계층

### 응용 계층

* // TODO

### 전송 계층

* // TODO

### 인터넷 계층

* // TODO

### 네트워크 접속 계층

* // TODO

# Protocol

각 계층별로 많이 사용되는 프로토콜에 대해 설명합니다.

## 응용 계층

### FTP

* // TODO

## 전송 계층

### TCP

* // TODO

### UDP

* // TODO

## 네트워크 계층

### IP

* // TODO

### ICMP

* // TODO

### ARP

* // TODO

## 데이터 링크 계층

### 이더넷

* // TODO

## 물리 계층

### // TODO

* // TODO

# 목적

개발의 목적 및 방향성을 정의합니다.

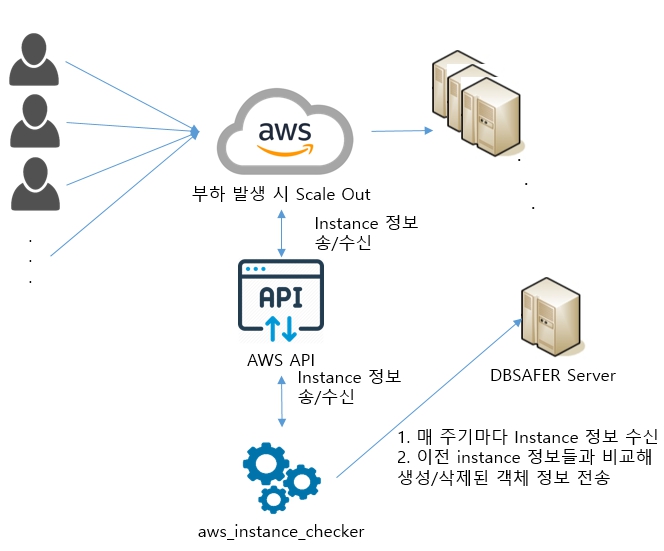
## 요구사항

* AutoScaling으로 인한 scale out 발생 시 새로운 인스턴스가 생성
  + 동일한 서비스가 더 늘어난 것
  + 기존 서버와 동일한 정책이 적용 필요 -> 해당 인스턴스의 서비스 등록 필요
  + 기존에는 새로 생성된 인스턴스를 수동으로 등록
* AutoScaling으로 인한 인스턴스의 생성/삭제 시 이를 자동으로 반영하는 기능 개발 요청
  + 1차 개발 : 인스턴스의 생성 시 신규 서비스 생성
  + 2차 개발 : 인스턴스의 삭제 시 해당 서비스 삭제
* 서비스 등록 시 다음과 같은 정보가 필요함(예상)
  + 서비스 종류(DBMS / FTP / Terminal)
  + 서비스 이름
  + 서버 IP 주소 / 포트

## 개발 예상 흐름

### API를 이용한 정보 전달

* AutoScaling으로 생성된 인스턴스를 주기적으로 검사해 변경 사항이 있을 경우 이를 DBSAFER 서버로 전달
* CLI를 사용하면 json / text / table 형식으로 반환되는 리턴 데이터를 별도로 파싱해야 하나 API를 사용하면 별도의 파싱 없이 필요한 데이터를 추출 가능



# 관련 조사 내용

개발을 위해 조사한 내용을 간략히 설명

## AWS Java API

* AWS에서는 CLI에서 제공하는 기능을 동일하게 API로도 지원하고 있으며 여러 언어에서 상ㅇ 가능
* 본 모듈은 JAVA를 이용해 개발할 예정

### 인증

* 특정 인스턴스의 정보를 얻기 위해서는 IAM 사용자 등록 후 발급받은 Key가 필요함

|  |
| --- |
| AWS SDK for Java를 사용하여 Amazon Web Services(AWS)에 액세스하려면 AWS 계정과 AWS 자격 증명이 필요합니다. AWS 계정의 보안을 증진하려면 루트 계정 자격 증명 대신 IAM 사용자를 사용하여 액세스 자격 증명을 제공하는 것이 좋습니다. |

* 이 자격 증명을 Java API에서 사용하는 방법은 여러 가지가 있음
  + <https://docs.aws.amazon.com/ko_kr/sdk-for-java/v1/developer-guide/credentials.html#credentials-file-format>
  + 환경 변수 값 : EnvironmentVariableCredentialsProvider 클래스 사용
  + Java 시스템 속성 : SystemPropertiesCredentialsProvider 클래스 사용
  + **기본 자격 증명 프로필 파일 : ProfileCredentialsProvider 클래스 사용**
  + Amazon ECS 컨테이너 자격 증명 : ContainerCredentialsProvider 클래스 사용
  + 인스턴스 프로파일 자격 증명 : InstanceProfileCredentialsProvider 클래스 사용
* 기본 자격 증명 프로필 파일
  + ~/.aws/credentials 파일
  + aws CLI가 설치된 환경이라면 aws configure를 수행하였을 경우 자동 생성
  + 또는 직접 파일을 편집해야 함
  + 만약 이 파일이 없으면 com.amazonaws.SdkClientException 발생.



Figure 4‑1 ~/.aws/credentials 파일

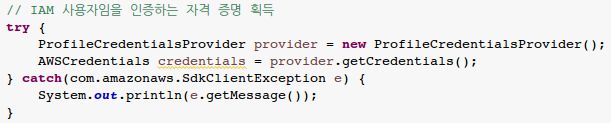


Figure 4‑2 자격 증명 획득 코드

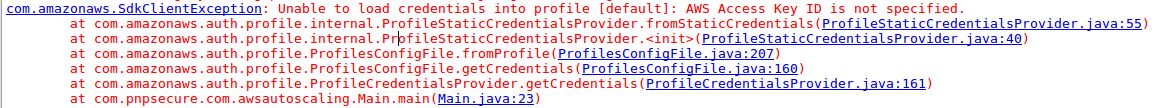


Figure 4‑3 자격 증명 획득 실패 시 발생하는 Exception

### AutoScaling 인스턴스 획득

* AutoScaling의 객체를 획득
  + DescribeAutoScalingInstancesRequest : AutoScaling의 객체를 요청하기 위한 Request
  + DescribeAutoScalingInstancesResult : Request에 대한 응답
  + AutoScalingInstanceDetails : Result에 담긴 객체를 하나씩 볼 수 있음
* //TODO 실행 코드
* //TODO 실행 결과 및 필요한 field 설명

### 해당 인스턴스의 상세정보 획득

* 3.1.2절의 결과로는 필요한 정보 중 Instance ID와 그룹명만 얻을 수 있음
* 서비스를 등록할 때 필요한 IP(도메인)이 필요
  + DescribeInstancesRequest
  + DescribeInstancesResult
  + Reservation //TODO 이 reservation이 사용되는 이유는?
  + Instance
* //TODO 실행 코드
* //TODO 실행 결과 및 필요한 field 설명

### 신규/삭제 인스턴스 반영

* // TODO 알고리즘 설명
* // TODO 실행 결과 설명