

# 요구사항 정의서

serverless-weather-app

프로젝트명	serverless-weather-app
작성자	김다정
작성일	2026년 1월 28일
버전	v1.0
문서 유형	요구사항 정의서 (SRS)

## 1. 프로젝트 개요

기상청 공공 API를 기반으로 전국 170여 개 지역의 날씨 정보를 수집·가공하여 제공하는 서버리스 날씨 웹 애플리케이션이다. AWS Lambda가 하루 2회 자동으로 데이터를 수집해 S3에 저장하고, 웹 프론트엔드는 S3에서 직접 데이터를 불러와 렌더링한다.

항목	내용
서비스 URL	https://weather-serv.duckdns.org/
개발 기간	2026년 1월 28일 ~
팀 구성	DN오토티브 DX팀 부트캠프 팀 프로젝트
배포 환경	AWS EC2 (t3.micro) + S3 + Lambda + RDS
주요 API	기상청 공공 API (apihub.kma.go.kr)

## 2. 시스템 아키텍처 요약

구성 요소	역할	비고
기상청 API	날씨 예보 데이터 제공	일 3회 발표 (05시·11시·17시)
AWS Lambda	API 호출 → JSON 가공 → S3 저장	Python, 하루 2회 실행 (05시·17시)
Amazon RDS	전국 170개 지역 코드·좌표 정보 저장	MariaDB, Lambda가 조회 전용으로 사용
Amazon S3	날짜별 날씨 JSON 파일 저장 및 서빙	202601290500.json 형식
Amazon EC2	Nginx 웹서버, HTML·CSS·JS 서빙	t3.micro, 퍼블릭 서브넷
GitHub Actions	코드 push 시 EC2 자동 배포	CI/CD 파이프라인
브라우저	Geolocation → S3 fetch → 날씨 렌더링	위치 거부 시 서울 기본값

## 3. 기능 요구사항

### 3-1. 데이터 수집 (Lambda)

ID	요구사항명	상세 내용	우선순위
FR-01	기상청 API 호출	전국 170개 지역 예보 구역코드(regid) 기반으로 기상청 옥상예보 API 호출	필수
FR-02	하루 2회 자동 실행	EventBridge를 통해 05시·17시 자동 트리거 (11시 데이터는 05시와 중복으로 제외)	필수
FR-03	RDS 지역 정보 조회	Lambda 실행 시 RDS region_view에서 지역코드·지역명·위도·경도 일괄 조회	필수
FR-04	JSON 가공 및 저장	API 응답 데이터를 numEf=1 기준으로 필터링 후 날짜별 JSON 파일로 S3에 저장	필수
FR-05	중복 방지	파일명을 발표시각 기반(YYYYMMDDHHMI.json)으로 동적 생성하여 덮어쓰기 방지	필수

### 3-2. DB 관리 (RDS)

ID	요구사항명	상세 내용	우선순위
FR-06	지역 마스터 테이블	regions 테이블: 예보구역코드(PK)·지역명·광역구분 저장	필수
FR-07	좌표 테이블	region_coordinates 테이블: 예보구역코드(PK·FK)·위도·경도 저장	필수
FR-08	뷰 테이블 제공	region_view: regions + region_coordinates JOIN 뷰로 Lambda 조회 최적화	필수
FR-09	확장성 확보	국내외 지역 추가 시 DB INSERT만으로 반영 가능하도록 설계	관장

### 3-3. 프론트엔드 (EC2 + 브라우저)

ID	요구사항명	상세 내용	우선순위
FR-10	위치 기반 날씨 표시	브라우저 Geolocation API로 현재 위치 파악 후 가장 가까운 예보 구역 날씨 표시	필수
FR-11	기본값 설정	위치 권한 거부 또는 조회 실패 시 서울(11B10101) 날씨 자동 표시	필수
FR-12	현재 날씨 표시	날씨 상태(맑음·구름많음·흐림·비·눈), 날씨 아이콘, 기온 표시	필수
FR-13	최저·최고 기온 표시	당일 최저 기온(04~05시 기준)·최고 기온(14시 기준) 표시	필수

FR-14	어제 날씨 비교	오늘 기온과 어제 기온을 비교하여 '어제보다 더 추움/더움' 정보 표시	필수
FR-15	반응형 UI	PC·모바일 화면 해상도 자동 대응	필수
FR-16	HTTPS 지원	SSL 인증서 적용 (duckdns.org)	필수

#### 3-4. 배포 (CI/CD)

ID	요구사항명	상세 내용	우선순위
FR-17	자동 배포	GitHub push 시 GitHub Actions를 통해 EC2에 자동 배포	필수
FR-18	S3 연결 테스트	배포 후 브라우저 콘솔에서 S3 연결 상태 확인 가능하도록 테스트 코드 제공	권장

#### 4. 비기능 요구사항

ID	분류	요구사항	기준
NFR-01	성능	Lambda 실행 완료	15분 이내 170개 지역 전체 처리 완료
NFR-02	성능	S3 응답 시간	JSON 파일 fetch 200ms 이내
NFR-03	가용성	서비스 업타임	EC2 단일 인스턴스 기준 99% 이상
NFR-04	보안	RDS 접근 제어	보안 그룹으로 Lambda 및 관리자 IP만 허용
NFR-05	보안	S3 퍼블릭 접근	날씨 JSON 파일만 퍼블릭, 원본 데이터는 프라이빗
NFR-06	보안	HTTPS 적용	전체 웹 트래픽 SSL/TLS 암호화
NFR-07	비용	Lambda 비용 최적화	하루 2회 실행으로 API 호출 비용 최소화
NFR-08	확장성	지역 데이터 확장	DB INSERT만으로 신규 지역 추가 가능
NFR-09	유지보수	코드 배포 자동화	GitHub Actions CI/CD로 수동 배포 제거

#### 5. 데이터 요구사항

##### 5-1. 기상청 API 응답 데이터

필드명	한글명	타입	비고
announceTime	발표시간	Number	년월일시분 (예: 202601290500)
numEf	발효번호	Number	1 = 오늘 대표 예보 (기준값)
regId	예보구역코드	String	11B10101 = 서울
ta	예상기온	String	단위: °C
wf	날씨	String	맑음·구름맑음·흐림·비·눈
wfCd	날씨코드	String	DB01:맑음, DB03:구름맑음, DB04:흐림
rnYn	강수형태	Number	0:없음, 1:비, 2:비/눈, 3:눈, 4:소나기
rnSt	강수확률	Number	0~100 (%)
latitude	위도	Double	region_view에서 결합
longitude	경도	Double	region_view에서 결합
region_name	지역명	String	region_view에서 결합

##### 5-2. DB 테이블 구조

테이블명	컬럼명	타입	제약 조건	설명
regions	regId	VARCHAR(20)	PK	예보구역코드
regions	region_name	VARCHAR(50)	NOT NULL	지역명
regions	broad_region	VARCHAR(50)	NULL	광역 구분
region_coordinates	regId	VARCHAR(20)	PK, FK	예보구역코드
region_coordinates	lat	DOUBLE	NOT NULL	위도
region_coordinates	lon	DOUBLE	NOT NULL	경도
region_view (뷰)	regId · region_name · lat · lon	-	-	regions + region_coordinates JOIN

#### 6. 인프라 요구사항

구성 요소	설정값	비고
VPC	weather-project-vpc (10.0.0.0/16)	DNS 호스트 이름 활성화
퍼블릭 서브넷	AZ 2a, AZ 2b 각 1개	IGW 연결, EC2 배치
프라이빗 서브넷	AZ 2c 1개	내부 리소스 보호
인터넷 게이트웨이	weather-project-igw	퍼블릭 서브넷 인터넷 연결
EC2	t3.micro, 퍼블릭 서브넷 2a	Nginx, HTTP(80), HTTPS(443), SSH(22) 허용
RDS	MariaDB, 퍼블릭 액세스 허용	서브넷 그룹: 2a+2b AZ, SSL 선택적 적용
S3	weather-project-s3-bucket-01	날씨 JSON 퍼블릭 서빙, 원본 프라이빗
Lambda	Python 런타임, 최대 15분	RDS 조회 + API 호출 + S3 업로드
EventBridge	하루 2회 (05:00, 17:00 KST)	Lambda 자동 트리거

#### 7. 제약 사항

--

항목	내용
API 호출 한도	기상청 API 일 최대 20,000건, 일 최대 5GB
Lambda 제한	최대 실행 시간 15분 이내 170개 지역 처리 완료 필요
인프라 비용	EC2 1대 고정 운영으로 비용 최소화 (ALB 미적용)
데이터 갱신	실시간 갱신 불가, 하루 2회 배치 업데이트
위치 정확도	예보 구역 단위 (읍·면·동 수준의 정밀 위치 불가)
보안	테스트 환경 특성상 RDS 퍼블릭 액세스 허용 (운영 시 프라이빗 전환 권장)

본 문서는 serverless-weather-app 프로젝트의 요구사항을 정의한 문서입니다. | 작성자: 김다정 | v1.0