Workload Automation

J-Jobs (Ver 2.0)

INTRODUCTION

LG CNS

Contents

Ⅰ J-Jobs 개요

Ⅱ J-Jobs 구성

Ⅲ J-Jobs 아키텍처

Ⅳ 주요기능

Ⅴ 부가기능

Ⅵ 지원 플랫폼

Ⅶ 적용사례

Why J-Jobs

제품개요

다양한 Template 을 통한 작업 자동화 지원
※ Command/Shell, DB, Java, FTP, Email, RestAPI 등

다양한 이벤트를 통한 workflow 자동화 지원
※ Token, 스케줄, 파일이벤트, SQL결과, 이메일 수신 여부 등

다양한 플랫폼 지원을 통한
이 기종 시스템 작업 자동화 지원
※ on-premise, 클라우드, 컨테이너, hybrid 환경 지원

Web 기반의 WYSIWYG workflow 기능을
이용한 손쉬운 작업 흐름 정의

주요기능

안정적 업무 운영

- 서버 고가용성 확보, 에이전트 그룹관리를 통한 부하 분산 및 장애격리
- 이벤트기반 작업 자동화 기능 제공
- 대용량 데이터 분산 파티셔닝 기능
- [Cloud/Container] 작업량 부하 감지에 따른 에이전트/Pod Auto Scaling

편리한 개발

- Job 개발 시 필요한 다양한 Template 제공
- 업무 간 배치 연계 지원
- REST API 기반의 타 시스템과의 쉬운 연계
- Excel 파일로 upload/download 가능
- Check-in/out 기능으로 배치 수행 중 수정 가능

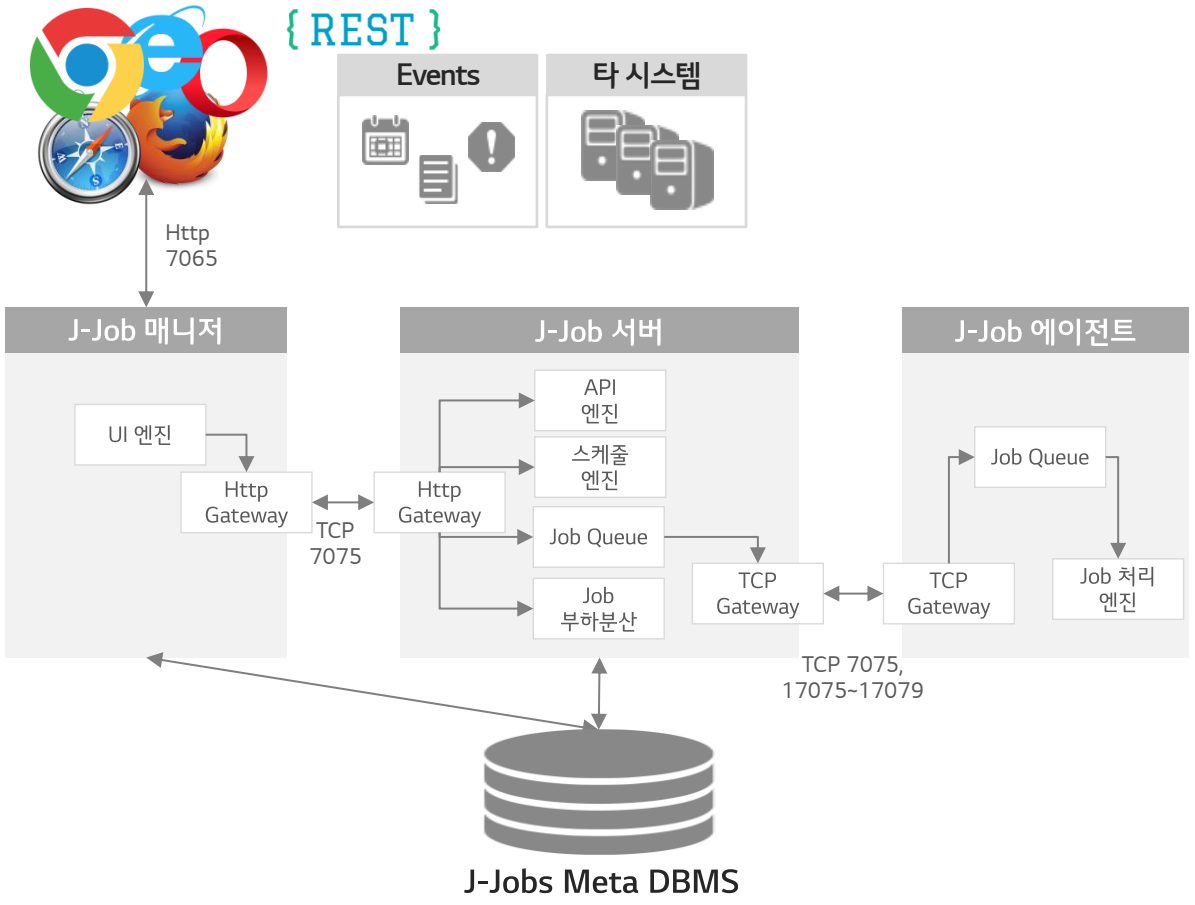
모니터링

- 에이전트 및 서버 상태 모니터링
- Job 실행 상태 모니터링
- Job 수행 통계 정보

실행 아키텍처

다양한 경로로 전달된 Job수행 요청을 적시에 가용한 환경에서 누락 없이 정확히 수행할 수 있습니다.

실행 아키텍처



구성 및 특징

J-Job 매니저

- 브라우저를 통해 다양한 수행요청 접수하고
- 가용한 서버를 검색하여
- 검색된 서버로 Http Gateway를 통하여 Job 요청을 전달

J-Job 서버

- 스케줄, 파티셔닝 등 필요한 Job수행 정보를 관리하고
- 가용한 에이전트를 검색
- 검색된 에이전트로 TCP Gateway를 통하여 Job수행 요청을 전달

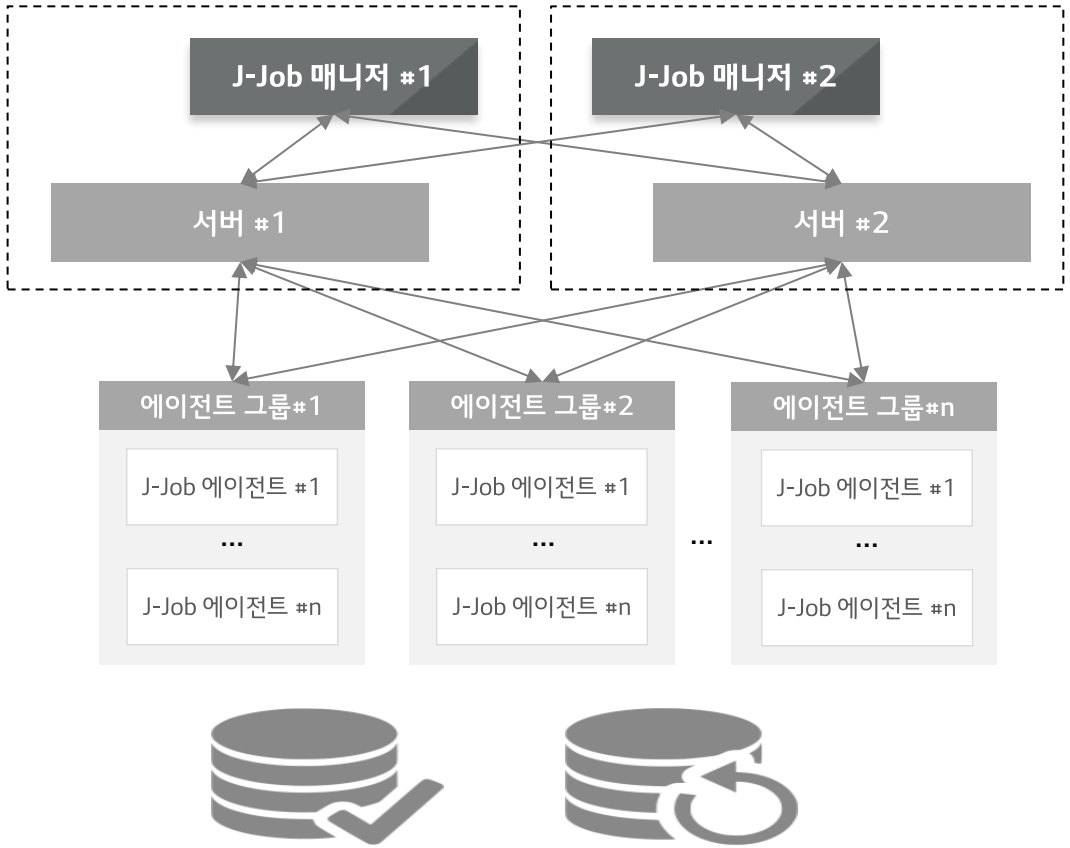
J-Job 에이전트

- Job 요청에 대한 정상 접수를 TCP Gateway를 통하여 응답을 보낸 후
- 전달 받은 Job수행 요청을 수행
- 수행된 결과를 저장

고 가용성 아키텍처

J-Jobs는 시스템 별 논리적인 서버그룹과 그 하위에 에이전트 그룹으로 구성되어 있고, HA로 구성된 자체 메타 DBMS을 내장하고 있어 고 가용성을 보장합니다.

고 가용성 지원 아키텍처



구성 및 특징

서버 그룹

- 시스템을 논리적으로 분리한 단위
- 여러 대의 서버를 논리적으로 묶을 수 있어 다중화 구성이 가능함
- 서버의 고 가용성을 보장

에이전트 그룹

- 업무 , 서비스 또는 어플리케이션 배포단위
- 여러 개의 에이전트를 논리적을 구성하여 작업 부하 분산 및 장애 시 고 가용성 보장

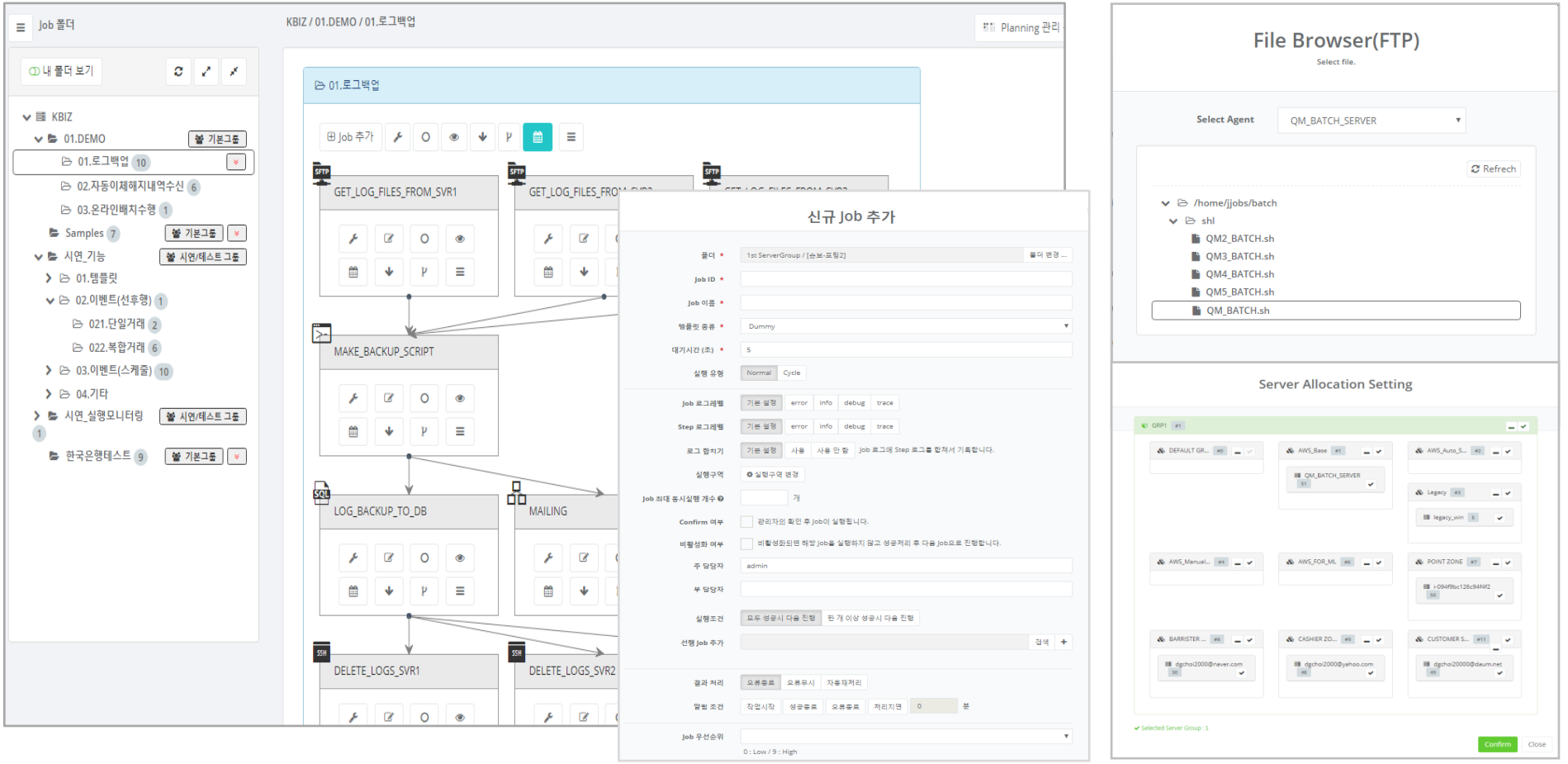
고 가용성 Meta DBMS

- HA를 지원하는 내장 DBMS 제공
- 설정기반 구성이 가능하여 필요 시 별도 구성 가능

작업편집 – 작업계획 (planning)

J-Jobs는 등록된 Job간의 선/후행을 지정하여 작업의 흐름을 제어 할 수 있습니다. 작업계획은 크게 속성값을 변경하여 처리하거나 그래픽기반으로 편집할 수 있고 대량의 경우 Excel 파일을 업로드 하여 정의할 수 있습니다.

Job 편집 화면

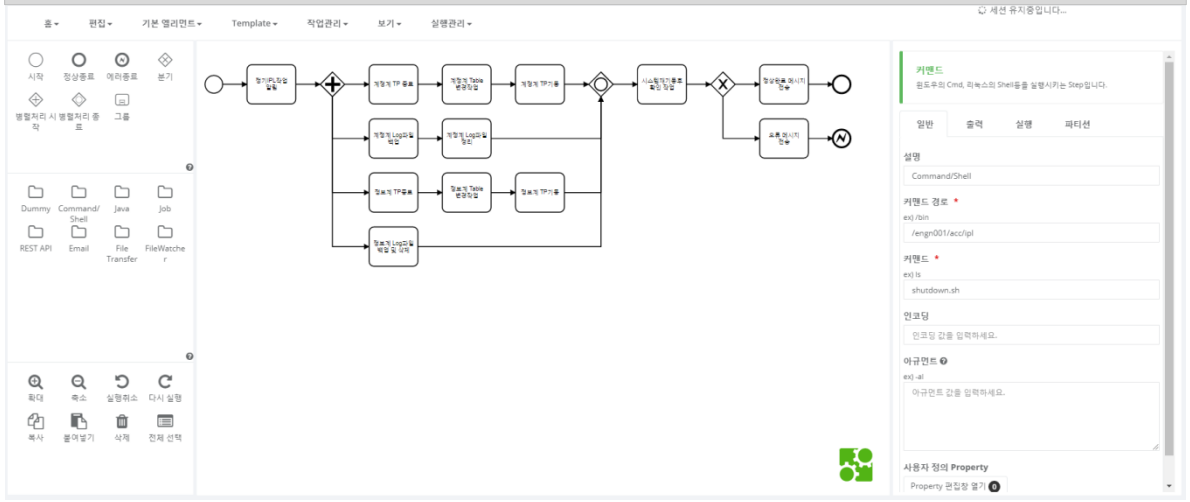


작업편집 – Web 기반 WYSIWYG 방식의 Job 편집

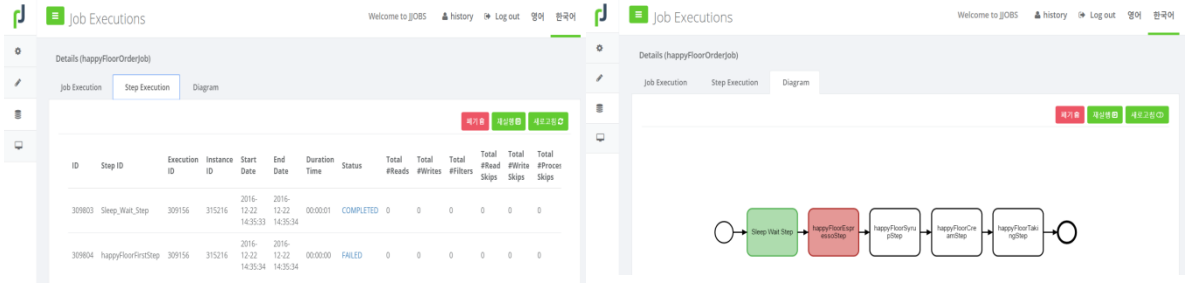
Job 편집기를 이용하여 Job의 처리 흐름을 정의하고 실행 시 정의된 흐름을 기반으로 단계적으로 확인할 수 있습니다. Job 편집화면에서는 툴바를 제공하여 간편하게 스케줄, 실행제한구역설정을 Job편집 시 사용할 수 있습니다.

Job 편집 화면

Job 편집기 화면



Job 실행 확인 화면



구성 및 특징

Job 편집기

- 다이어그램을 통해 Business Process 확인
- 드래그 방식의 직관적 UI로 Job 생성 및 수정 편의성 제공
- 툴바를 이용한 Job 설정 지원

편의 기능 제공

- Job 편집기 상의 Business Process 다이어그램을 파일로 저장 가능
- Job 편집기 화면에서 Job의 유효성 체크 가능

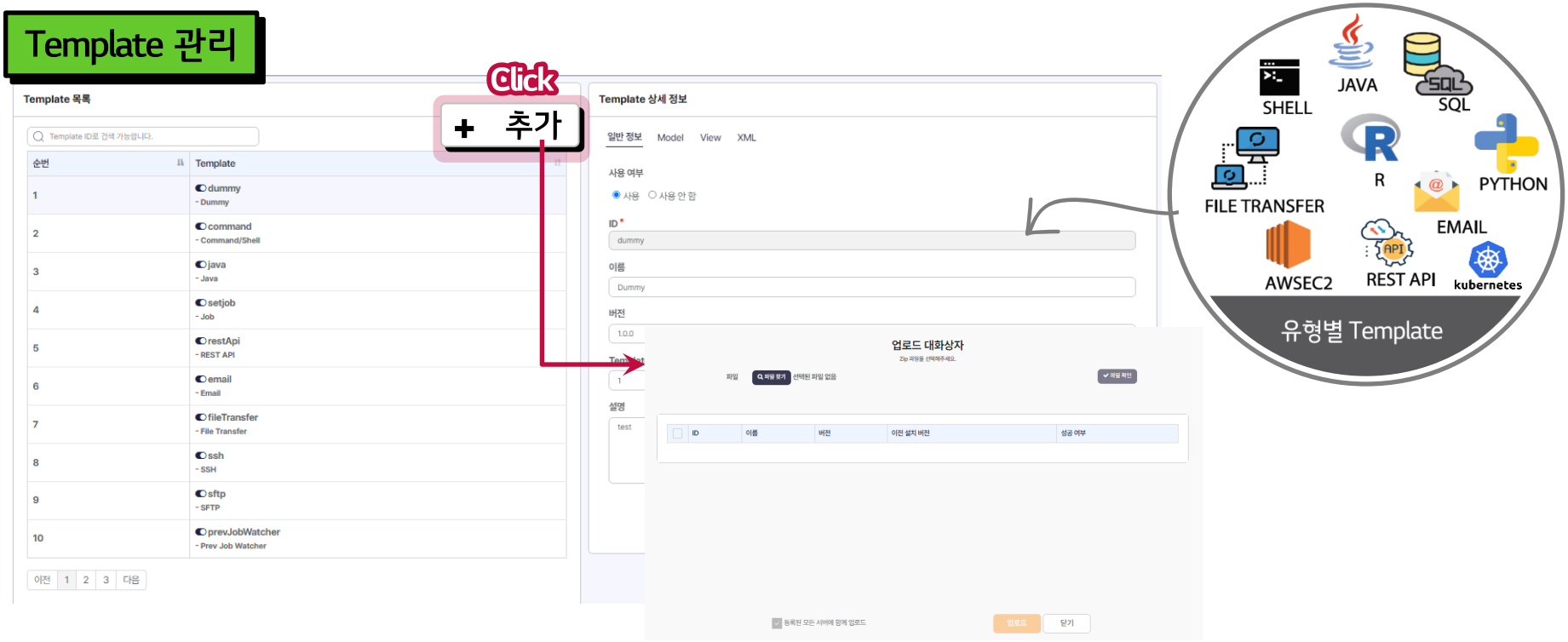
Job 실행 확인

- 전체 프로세스의 완료, 장애 여부 한눈에 파악 가능
- 각 스텝 별 Status 로그 확인 가능

작업편집 – 템플릿 관리

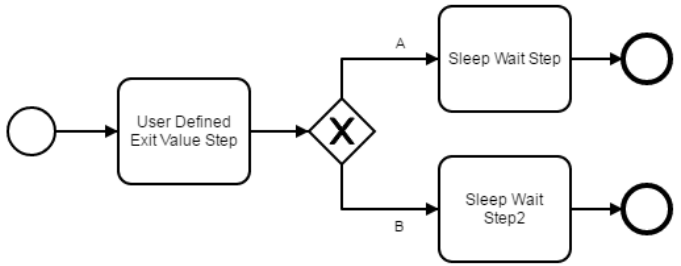
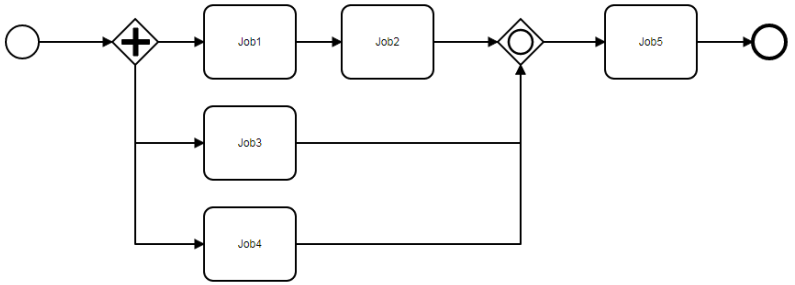
Job 생성을 용이하게 지원하기 위해 다양한 환경에 대한 템플릿을 활용함으로써 손쉽게 배치환경을 구성할 수 있도록 지원합니다. JSON 및 XML 파일을 동시에 제공하여 사용하던 배치환경의 용이한 이관이 가능합니다. 또한 템플릿 자체의 추가도 손쉽게 가능합니다.

스케줄링 작업 처리과정



작업편집 – Step 순차, 분기, 병렬 처리

Job은 한 개 이상의 Step으로 구성할 수 있습니다. 또한 각 Step을 순차, 분기, 병렬로 흐름을 제어할 수 있도록 구성할 수 있고 실시간으로 반영할 수 있는 기능을 제공합니다.

종류	다이어그램	특징	JSR-352 Spec
순차 처리		이전 Step이 종료되면 이이서 다음 후속 Step이 실행됨	Next
분기 처리		이전 Step의 상태 값이나 종료 값에 따라 실행할 다음 Step이 결정되어 실행	Decision
병렬 처리 (Split)		서로 다른 Step이 동시에 실행되며 모든 병렬처리가 끝나면 다음 Step으로 진행	Split
배포		수정된 Job 정보는 별도의 배포 작업 없이 다음 Job 실행부터 자동 반영	

이벤트기반 작업요청 – 스케줄링 기능

J-Jobs는 고 가용성을 보장하는 Time-Based의 Job 스케줄링 기능을 제공합니다. 시간/캘린더/간격 등에 대해 스케줄을 수립할 수 있습니다

스케줄링 작업 처리과정

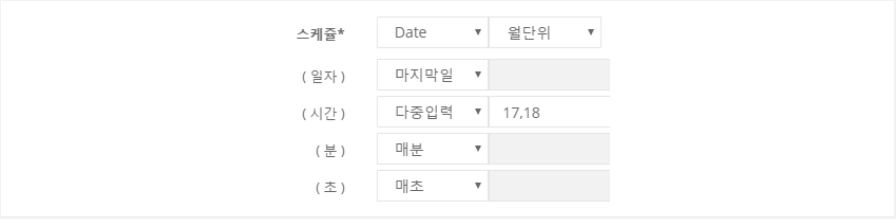
이벤트기반 작업요청 – 스케줄링 기능 (다양한 스케줄링 방식제공)

고객의 다양한 비즈니스 환경을 고려하여 날짜 지정, 실행 시간 간격 등록 및 캘린더 지정 방식의 기능을 제공하여 다양한 스케줄 유형을 지정할 수 있습니다.

구성 및 특징

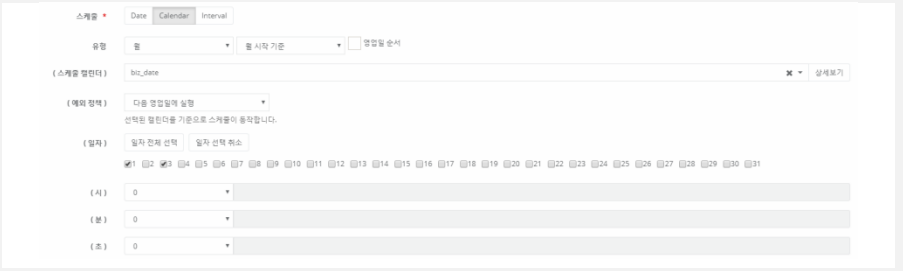
날짜 기준 (Date)

- 날짜 기준으로 실행 예약 가능
- 특정 일시, 마지막일 등 지정 가능
- 일 단위/주 단위/월 단위의 반복일정 등록 가능



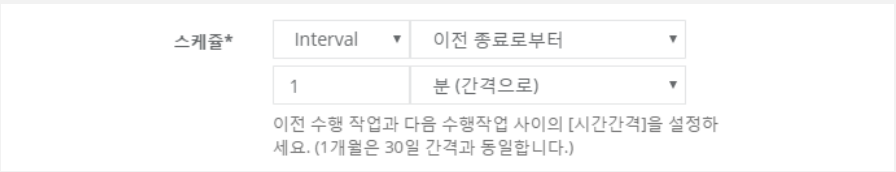
캘린더 기준 (Calendar)

- 특정 캘린더를 기준으로 실행 예약 가능
- 영업일 기준 예외 조건 추가 기능



실행 시간 간격 기준 (Interval)

- 이전 실행과 다음 실행 사이의 간격을 지정하여 실행 예약
- 이전 실행의 시작 시간이나 종료시간 기준으로 설정
- 초/분/시/일/주/월 단위의 반복일정 등록 가능



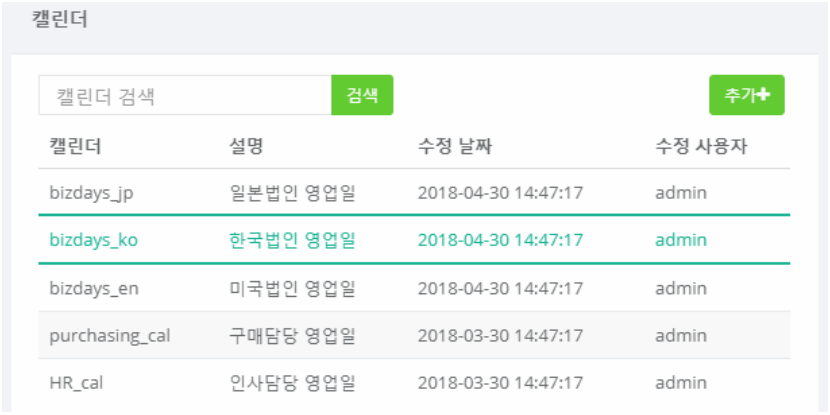
기타 기능

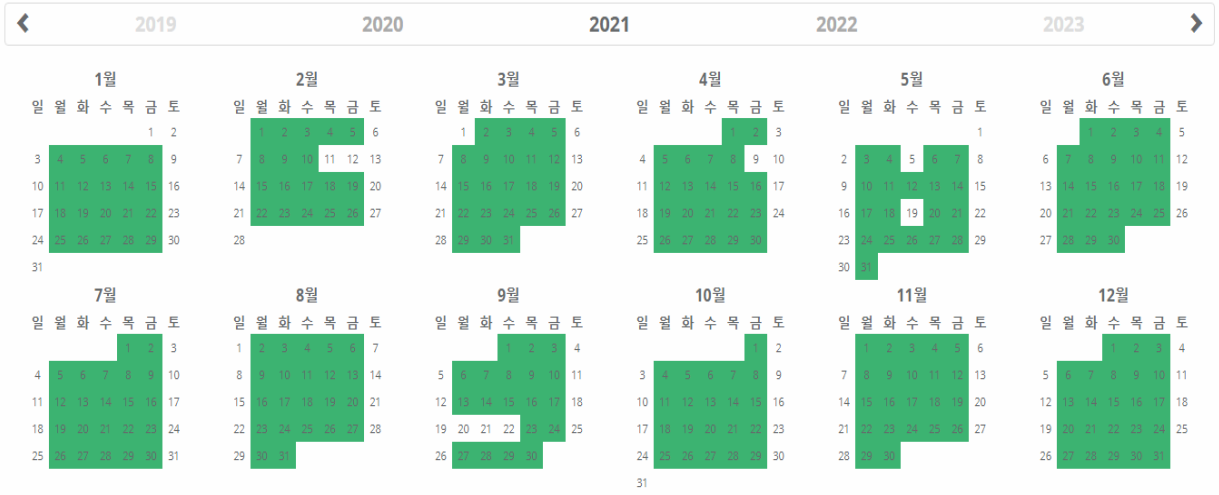
- 유효기간 및 유효시간을 설정하여, 설정된 기간과 시간 동안에만 배치 수행 가능함
- 스케줄 실행 이력 조회 기능을 제공함
- 작업 수행시간이 스케줄 간격보다 큰 경우 중복실행을 방지함

이벤트기반 작업요청 – 스케줄링 기능 (영업일 관리)

J-Jobs는 Calendar 기능을 제공하여 사이트, 국가 및 법인 등의 임의의 카테고리를 정의하여 영업일을 다중으로 관리할 수 있습니다. 등록된 영업일 Calendar는 스케줄과 연계해서 사용할 수 있습니다.

Calendar 관리





구성 및 특징

다중 Calendar 관리

- 영업일을 다양한 카테고리로 정의하여 관리하는 기능을 제공
- 사이트/국가/법인 별 Calendar 등록
- Rule Base Calendar 제공

스케줄 연계

- 등록된 Calendar를 사용하여 스케줄을 입력하면 영업일 기준으로 스케줄이 계산됨
- 매월 첫 영업일, 마지막 영업일 등의 스케줄 등록 가능

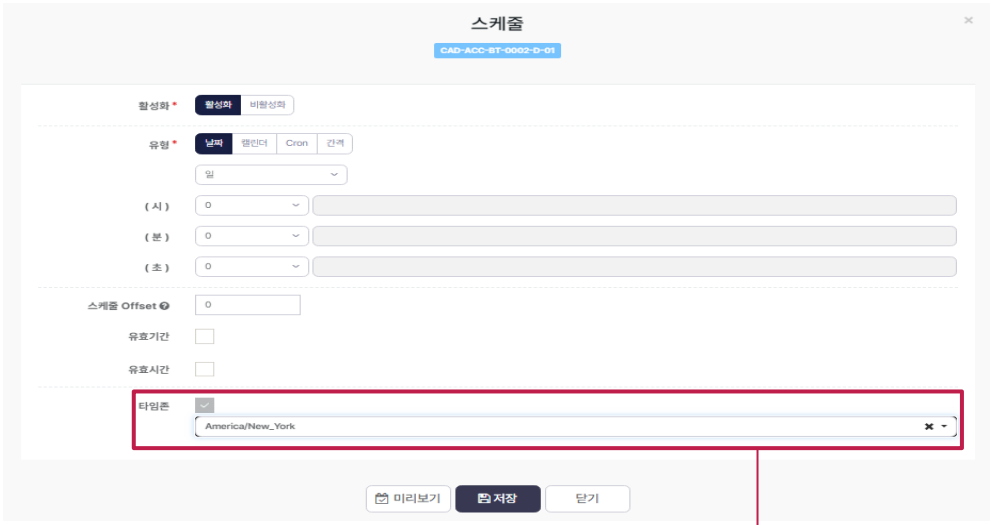
승인 Calendar 적용

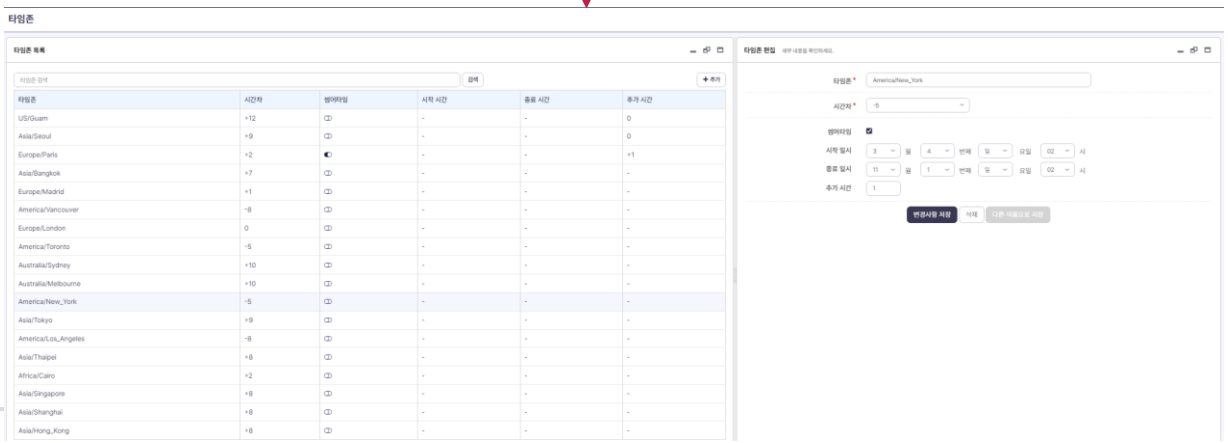
- 등록된 스케줄이 승인 Calendar를 기준으로 영업일이 아닌 경우에 대한 추가 처리 기능 제공
- 항상 실행/다음 영업일/이전 영업일/실행안함 등의 처리 옵션 제공

이벤트기반 작업요청 – 스케줄링 기능(Time-zone & Summer Time 설정 기능)

스케줄 작업 등록 시, Time-zone을 선택하면 등록된 Time-zone을 기준으로 Job이 수행되며 Time-zone 관리에서는 Summer Time 옵션 기능을 추가로 제공합니다.

J-Job Manager Time-zone 관리 화면





구성 및 특징

Time-zone 관리

- Time-zone 정보 추가/수정/삭제
- 스케줄 작업 등록 시, Time-zone을 선택하면 Time-zone을 고려하여 배치 처리 시간이 스케줄 됨
- 처리 기준 시간은 DB 시간임

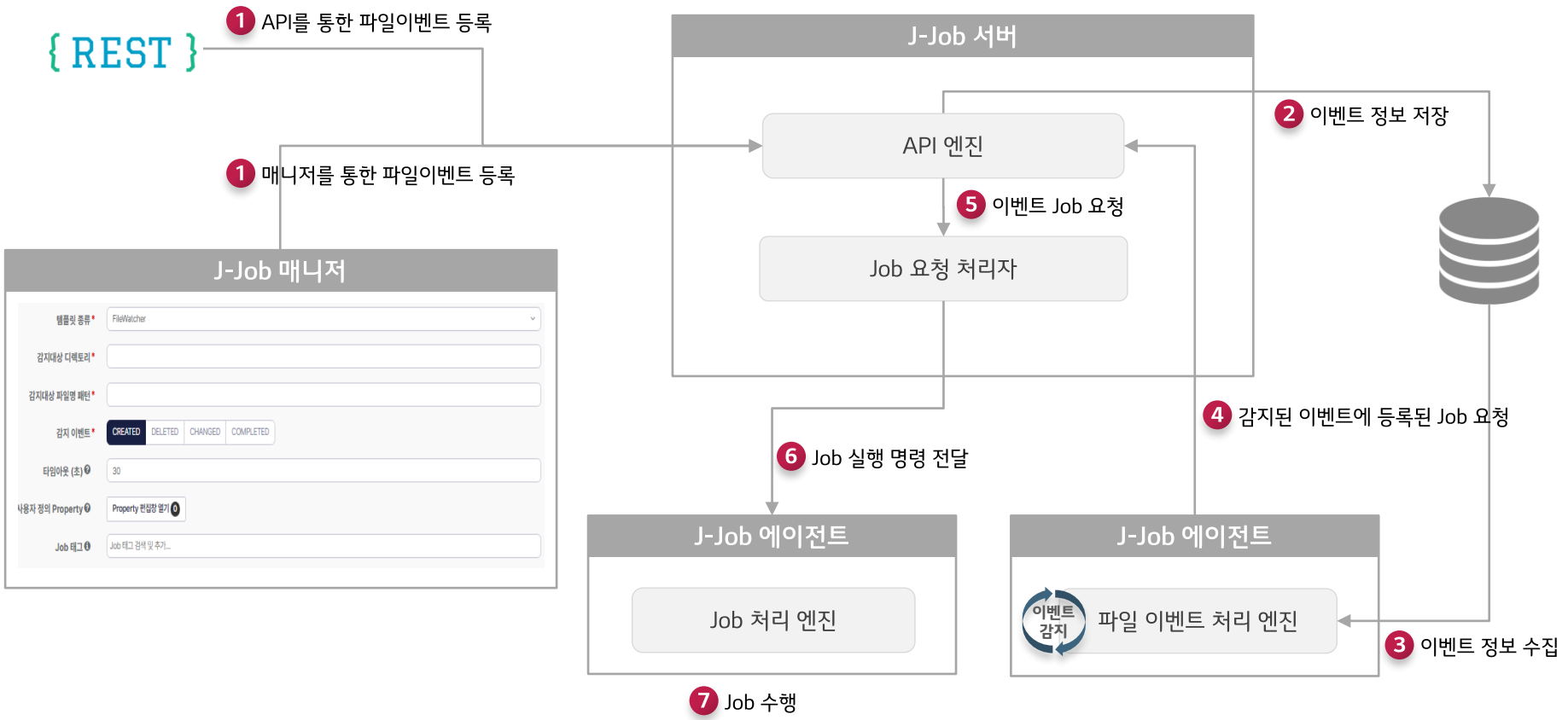
Summer Time 옵션 기능 제공

- 표준시간대별 Summer Time 적용 기간 설정
- Summer Time 적용 시, 추가되는 시간 설정 (예, 1시간)
- 스케줄 작업 등록 시, Summer Time 옵션이 사용된 Time-zone을 선택하면 Time-zone & Summer Time을 고려하여 배치 처리 시간이 스케줄 됨

이벤트기반 작업요청 – 파일이벤트

다양한 종류의 파일의 생성,삭제,변경 및 변경완료 이벤트를 감지하여 작업을 요청합니다.

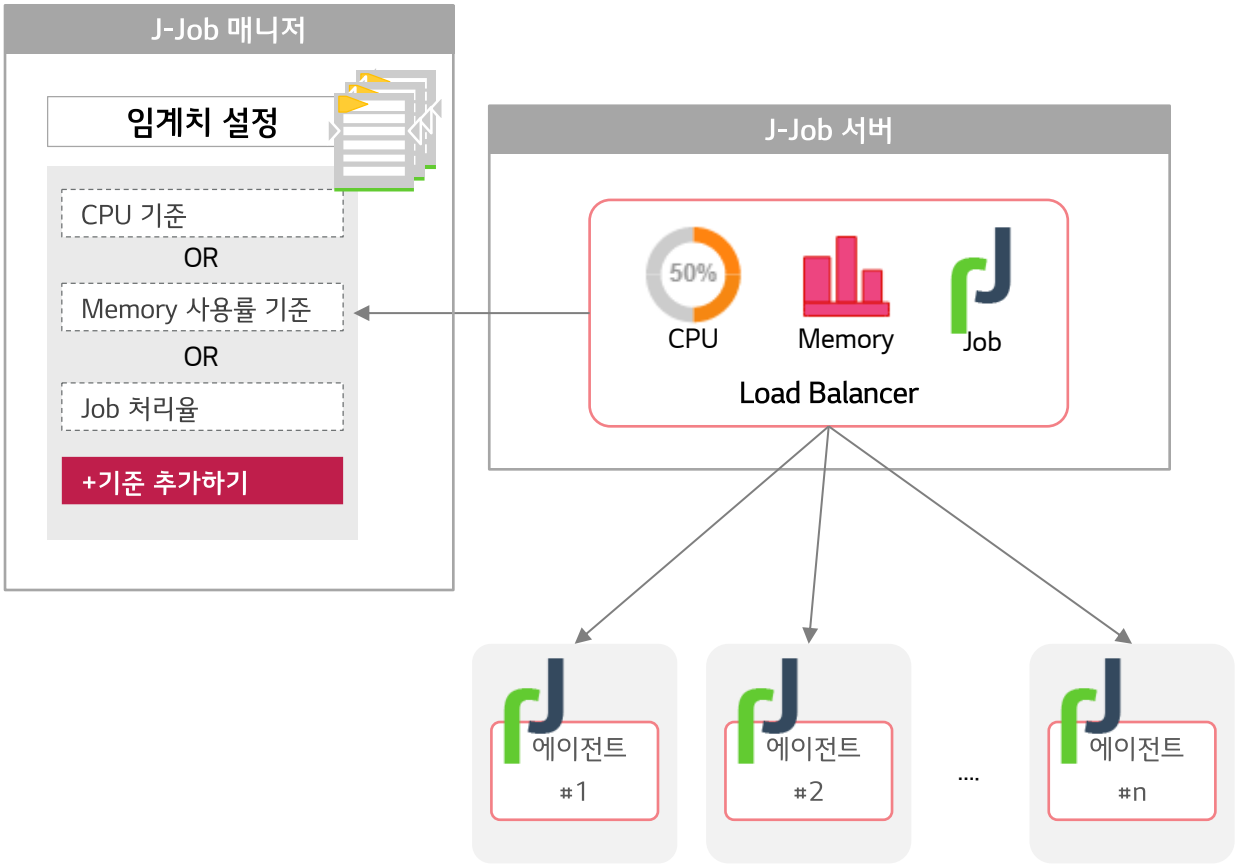
파일이벤트 작업 처리과정



부하분산 기능 – 시스템 자원상태 기반

J-Jobs 부하분산 기능은 에이전트 그룹내의 에이전트의 자원사용률 및 Job처리율을 실시간으로 파악하여 최적의 에이전트에서 작업이 실행될 수 있도록 부하를 조절을 수행합니다.

부하분산 메커니즘



구성 및 특징

부하분산

- 서버는 에이전트의 부하 상태를 동적으로 분석하여 Job을 분산하여 처리를 요청 함
- 에이전트 부하 상태 요소를 설정 값으로 조정하여 부하를 조절할 수 있음
- 에이전트 부하 상태 요소로는 CPU 사용률, 메모리사용률 또는 수행중인 Job 처리율 등이 있음

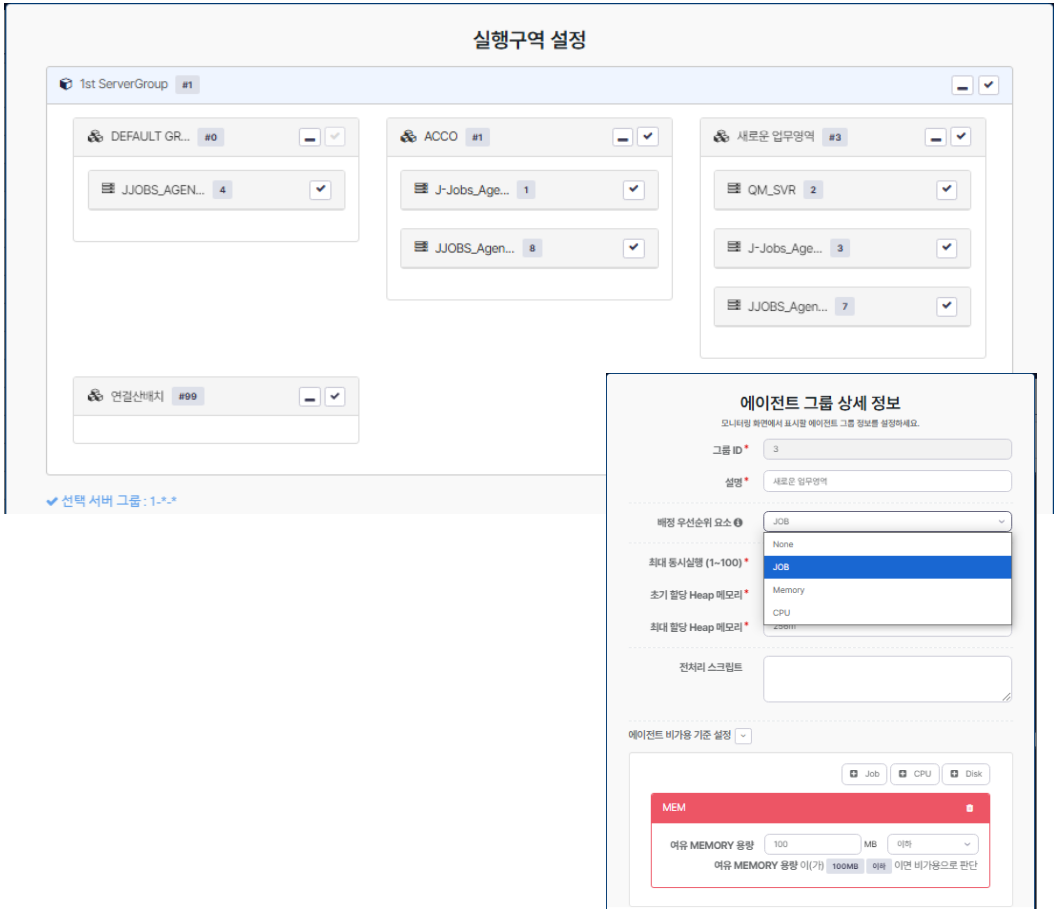
고 가용성 기능 제공

- 서버는 에이전트 상태를 모니터링
- 서버에서는 문제가 되는 에이전트로 작업을 할당하지 않고 정상적인 에이전트로만 작업을 할당 함
- 작업처리가 가능한 에이전트가 임계치를 초과하는 경우 사전에 경고를 보내고 Scaling기능과 연결된 경우 자동 Scaling 됨

부하분산 기능 – 다양한 부하분산 설정

Job 단위로 실행구역 제한, 최대 동시 실행 건수 제한, 우선순위 설정 등의 다양한 부하분산 관련 설정 기능을 제공합니다.

부하분산 설정 관리 화면



구성 및 특징

실행 구역 제한

- 그룹/에이전트그룹/에이전트 별로 작업 할당 가능
- 특정 Job을 특정 에이전트에서만 처리하도록 제한 가능

동시 실행 건수 제한

- 특정 작업의 작업 부하를 고려하여 최대 동시 작업 수행 건 수를 제한 하는 기능
- 동시 실행 건수가 제한된 부하분산 설정을 Job에 등록하면 해당 Job은 전체 에이전트에서 제한된 실행 건수로 제한되고 초과 건은 큐에서 대기됨

작업 우선순위 설정

- Job 단위 우선순위 설정 가능
- 실행 대기 큐 내부에서 우선 순위 설정에 따라 조정 됨

실행 – 입력 파라미터

J-Jobs는 작업 실행 시 개별 작업 실행에 필요한 파라미터 뿐만 아니라 공통으로 사용할 수 있는 파라미터 기능을 제공합니다. 공통 파라미터는 Global 파라미터와 폴더(그룹) 파라미터가 있습니다.

파라미터 유형 및 관리

파라미터 목록

항목의 구분 없이 입력하면 결과가 조회되어 나타납니다.

+ 추가 | 삭제

파라미터 이름	파라미터 값	수정 날짜	수정 사용자
PREV_OMONTH1_WDATE_S0	BIZ_DATE_ADD_DATE("def-calendar",CONCAT(USER_DATE_ADD_MONTH(PARAM(ODATE),"yyyyMM",-1),"01"),"yyyyMMd...	2024-02-19 10:52:46	root (root)
PREV_OMONTH3	USER_DATE_ADD_MONTH(PARAM(ODATE),"yyyyMM",-3)	2024-02-19 10:52:46	root (root)
PREV_OMONTH2	USER_DATE_ADD_MONTH(PARAM(ODATE),"yyyyMM",-2)	2024-02-19 10:52:46	root (root)
PREV_OMONTH1	USER_DATE_ADD_MONTH(PARAM(ODATE),"yyyyMM",-1)	2024-02-19 10:52:46	root (root)
OMONTH	USER_DATE_ADD_MONTH(PARAM(ODATE),"yyyyMM",0)	2024-02-19 10:52:46	root (root)

폴더 파라미터

실행 시 사용할 파라미터를 확인 및 입력값을 수정하시기 바랍니다.

CAD-회계정산

전역 / 폴더 파라미터 목록 | 파라미터 예약어 목록

파라미터 이름	파라미터 값
ACODE	12345
DCODE	$(DEPART_CODE)
P_DATE	$(PDATE)
N_DATE	$(NEXT_DATE)

키워드

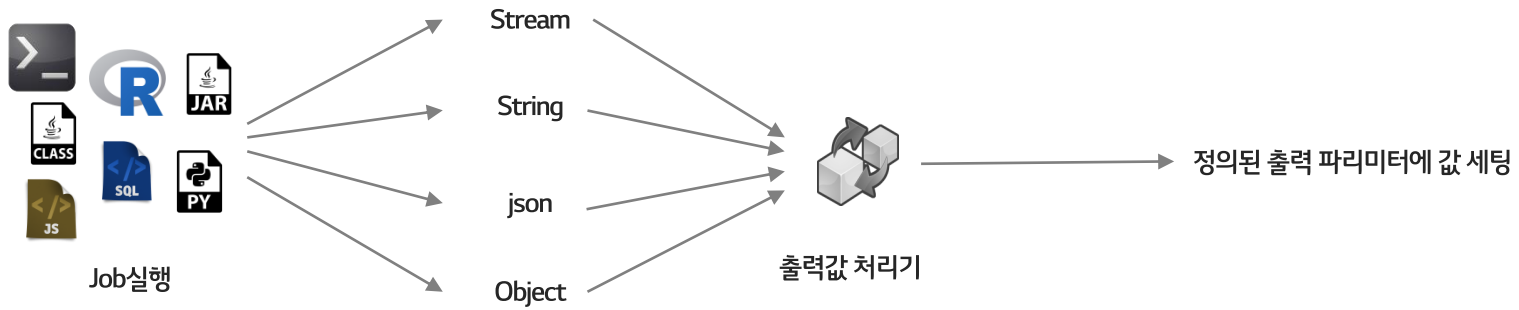
파라미터 타입의 Property의 값 부분에 아래 내용을 넣어두면 실행 시점에 값이 치환됩니다.

${SERVER_DATE_ADD_HOUR("yyyyMMddHHmmss",2)}	DB 현재시간에 2시간을 더한 시간 yyyyMMddHHmmss 형태로 반환
${SERVER_DATE_ADD_DATE("yyyyMMdd",2)}	DB 현재시간에 2일을 더한 시간 yyyyMMdd 형태로 반환
${SERVER_DATE_ADD_MONTH("yyyyMMdd",2)}	DB 현재시간에 2달을 더한 시간 yyyyMMdd 형태로 반환
${SERVER_DATE_ADD_YEAR("yyyyMMdd",2)}	DB 현재시간에 2년을 더한 시간 yyyyMMdd 형태로 반환
${AGENT_DATE("yyyyMMdd")}	에이전트 현재시간 yyyyMMdd형태로 반환
${AGENT_DATE_ADD_SEC("yyyyMMddHHmmss",2)}	에이전트 현재시간에 2초를 더한 시간 yyyyMMddHHmmss 형태로 반환
${AGENT_DATE_ADD_MIN("yyyyMMddHHmmss",2)}	에이전트 현재시간에 2분을 더한 시간 yyyyMMddHHmmss 형태로 반환
${AGENT_DATE_ADD_HOUR("yyyyMMddHHmmss",2)}	에이전트 현재시간에 2시간을 더한 시간 yyyyMMddHHmmss 형태로 반환
${AGENT_DATE_ADD_DATE("yyyyMMdd",2)}	에이전트 현재시간에 2일을 더한 시간 yyyyMMdd 형태로 반환
${AGENT_DATE_ADD_MONTH("yyyyMMdd",2)}	에이전트 현재시간에 2달을 더한 시간 yyyyMMdd 형태로 반환
${AGENT_DATE_ADD_YEAR("yyyyMMdd",2)}	에이전트 현재시간에 2년을 더한 시간 yyyyMMdd 형태로 반환

실행 – 출력 파라미터

Job 수행에 따른 다양한 결과 값을 파리미터로 처리 할 수 있고 이렇게 처리된 출력 파리미터는 다음 작업의 입력 파리미터 또는 분기작업의 판단 값으로 다양하게 사용할 수 있음

출력 파라미터



사용자 정의 Property 목록

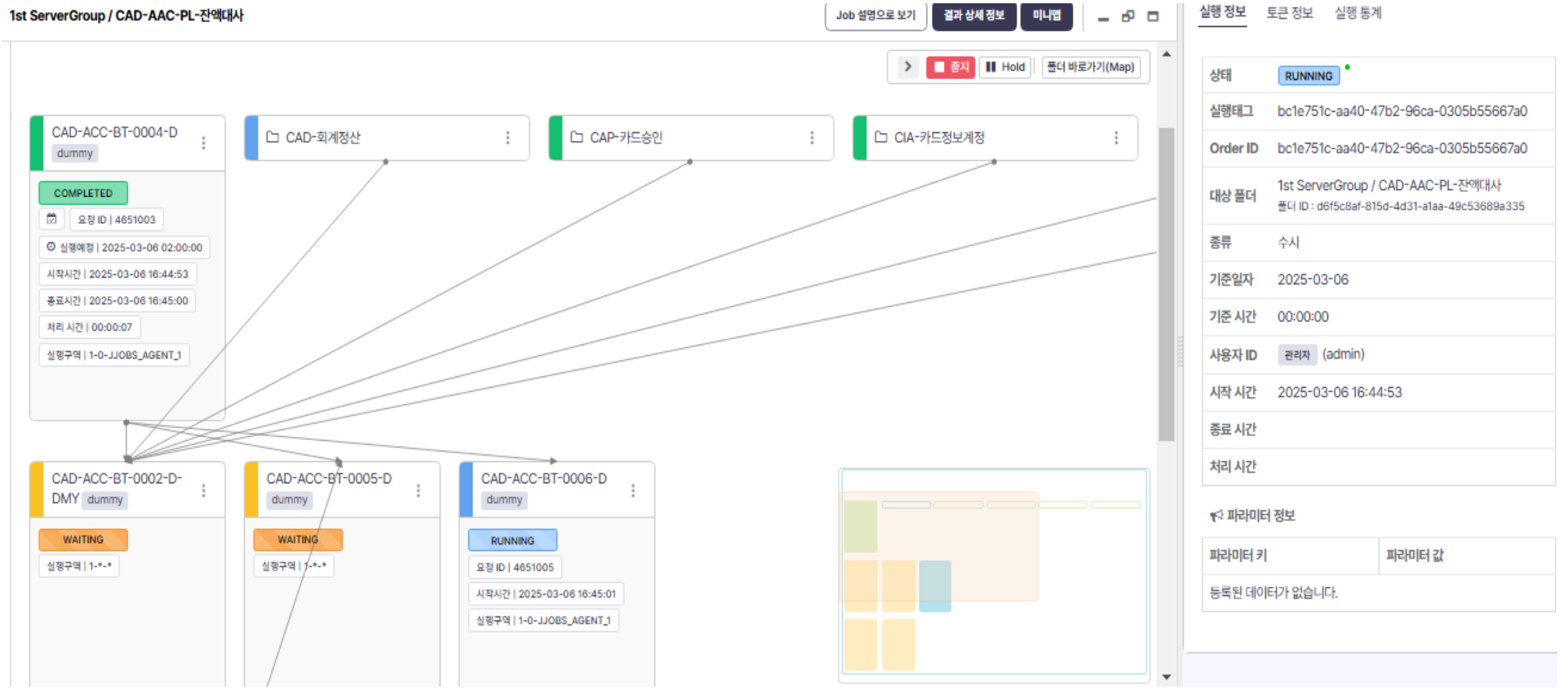
순번	종류	이름	값
3	파라미터	RESULT 대체	$(OUTPUT.JOB_RST)
2	출력변수	대체	
1	파티션	대체	

GET_VALUE_FROM_STDOUT(1,2)	표준출력으로 출력되는 내용 중 특정 위치 값을 추출해서 반환 (첫번째 인자:시작위치, 두번째 인
GET_FULL_OUTPUT()	표준출력의 내용을 모두 반환. (표준 출력의 크기가 최대 1MB 이내인 경우만 가능)
INDEX_OF("keyword",true)	표준출력으로 부터 입력한 키워드의 시작 위치 반환(일치한 값이 없으면 -1반환)(첫번째 인자:
SUBSTRING_BY_KEYWORD("keyword",1,true,true)	전체 표준 출력 결과 중 키워드가 존재할 경우 키워드 위치(혹은 키워드 제외)로부터 주어진 길이 는 최대 1000자까지 가능합니다.(첫번째 인자: 키워드, 두번째 인자 : 길이, 세번째 인자:키워드
GET_RESULT_TO_STRING(true)	Step수행 후 결과값을 문자열 형태로 반환(첫번째 인자:개행문자제거여부)
GET_RESULT_TO_JSON(true)	Step수행 후 결과값을 json형태로 반환(첫번째 인자:개행문자제거여부)
GET_RESULT()	Step수행 후 결과값을 Object형태로 반환
GET_RESULT_LIST_SIZE()	Step수행 후 결과값이 list인 경우 그 사이즈를 반환
GET_RECORD_AT_FROM_RESULT_LIST(1)	Step수행 후 결과값이 list인 경우 특정 위치의 레코드를 Object로 반환(첫번째 인자:추출하고
GET_FIELD_VALUE_FROM_RESULT("key")	Step수행 후 결과값이 Map형태이거나 Class형태인 경우 해당 key 또는 filed명에 해당되는
TO_STRING(object,true)	입력 받은 Object를 String값으로 반환(첫번째 인자:Object, 두번째 인자:개행문자제거여부)
TO_JSON(object,true)	입력 받은 Object를 json형태로 변환하여 반환(첫번째 인자:Object, 두번째 인자:개행문자제

실행 – 작업 실행 (기준일자 처리 지원)

J-Jobs는 선/후행 관계, 즉 전사적인 작업 계획을 수립하고 이를 실행/중지/재시작을 수행할 수 있습니다. 정기적으로 특정 시간 기준으로 자동 반복 작업이 가능하고 특정일을 기준일로 수시 작업도 가능합니다.

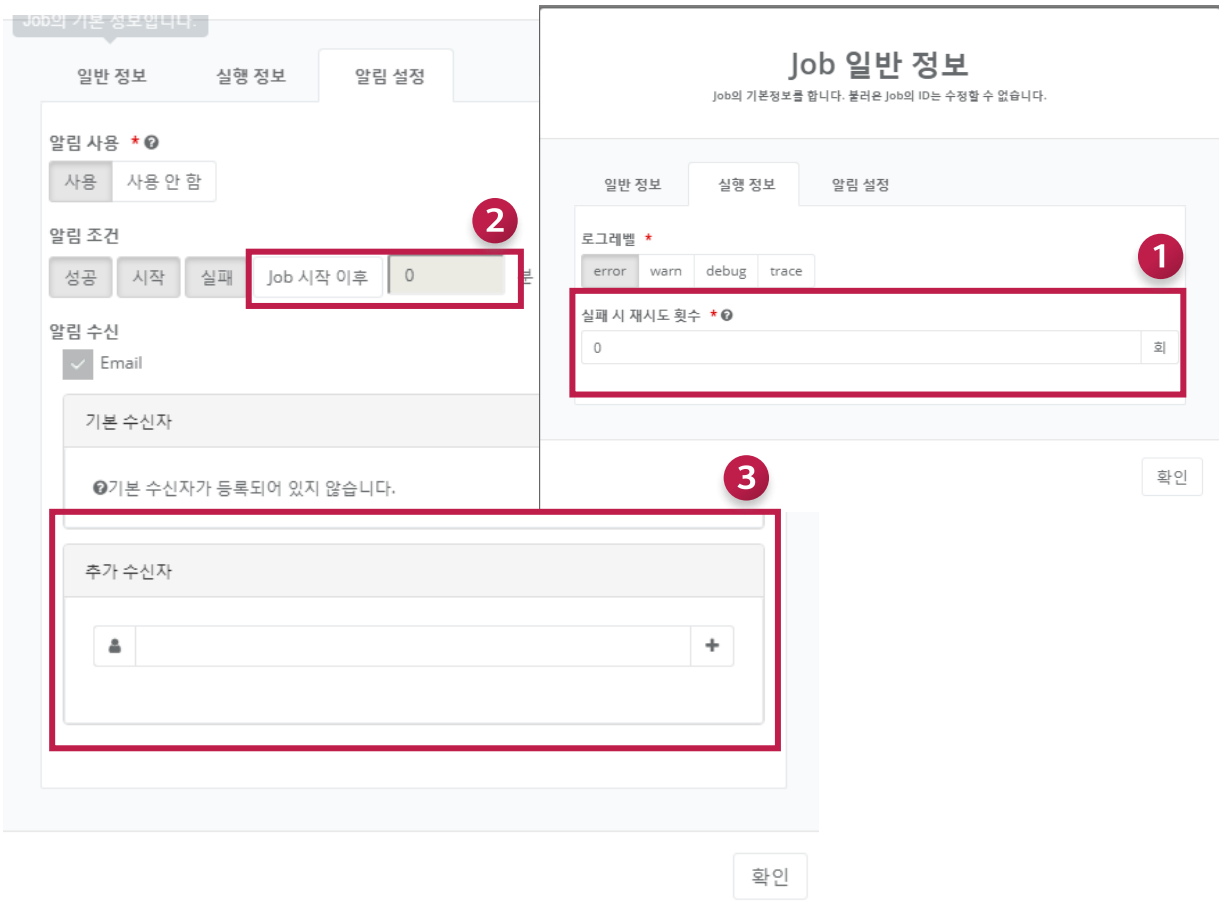
작업계획 실행/중지/재처리



실행 – 예외 발생 알림 설정

J-Jobs는 작업 실행 중 오류/시간 초과 등 예외 사항 발생 시, 지정된 재시도 회수만큼 실행 담당자에게 통보합니다.

예외 시 알림 설정



구성 및 특징

1 실패 시 재시도 횟수 설정

- 예외 사항 발생시 자동 재시도가 필요한 경우 재시도 회수를 지정하면 해당 회수만큼 자동 재실행 처리

2 시간 초과 설정

- 알림 조건 중, 시간 초과 시간을 설정하면 해당 시간이 지나도 작업이 끝나지 않을 경우 담당자에게 통지

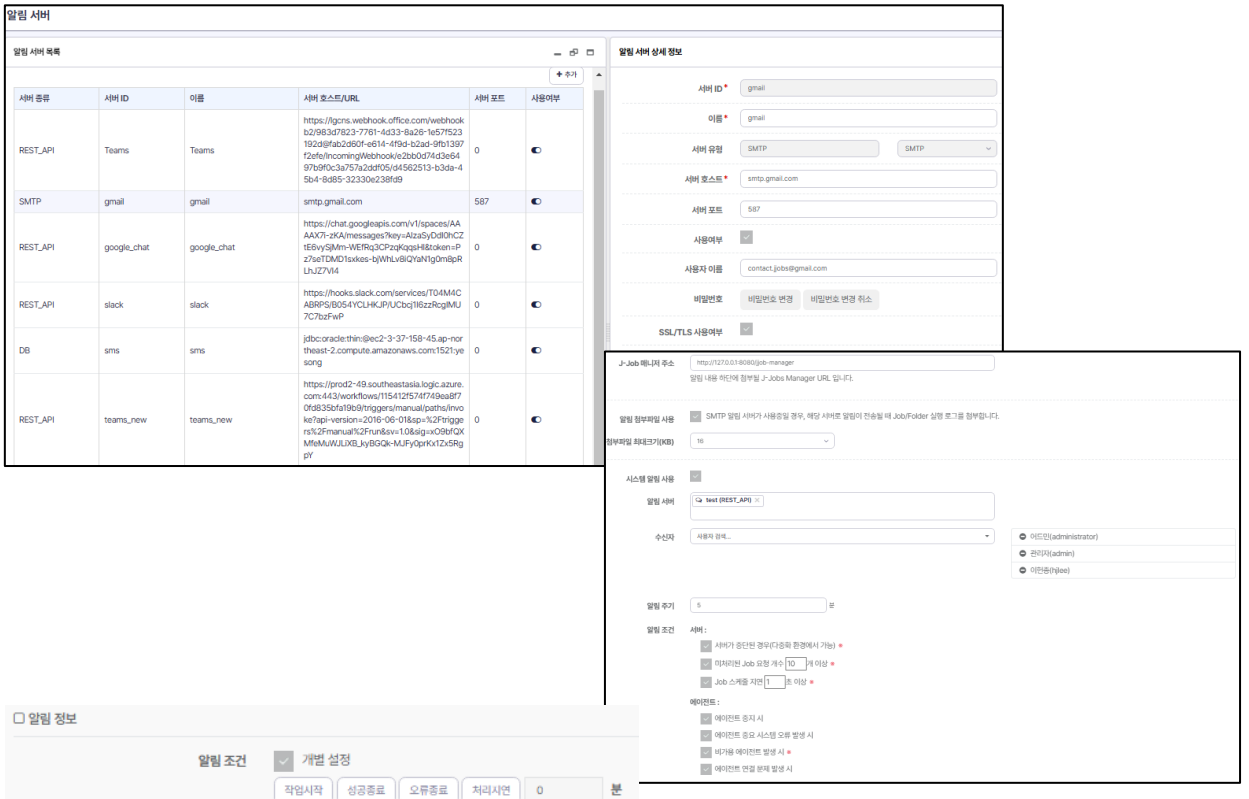
3 Job별 결과 통지

- Job 별로 일반 정보 등록 시 알림 설정 기능 제공
- 성공/시작/실패/지연의 조건을 중복 설정 가능
- Job별로 추가 수신자 설정 가능

실행 – 예외 발생 알림 설정 (알림 기능)

J-Jobs는 서버 및 에이전트 상태에 이상이 있을 경우 관리자에게 알림 메시지를 전달하고 관리화면에 알림 창을 띄워 현재 상황을 알려줍니다. Job의 성공/실패/지연처리에 따른 알림 기능도 제공합니다.

알림 설정 화면



구성 및 특징

전역 알림 설정

- 알림 조건을 중복으로 선택할 수 있음
- 알림 조건은 서버 이상, 에이전트 이상, 지연처리 Job건수, 스케줄 지연이 있음

알림 서버 설정

- 메일 서버나 SMS 정보를 등록하여 알림 설정 시 사용됨
- 알림 서버 별로 템플릿을 등록할 수 있어 다양한 형태의 메시지를 전송할 수 있음

Job별 결과 통지

- Job 별로 일반 정보 등록 시 알림 설정 기능 제공
- 성공/시작/실패/지연의 조건을 중복 설정 가능
- Job별로 추가 수신자 설정 가능

실행 – 작업 결과에 따른 후 처리

Job의 실행설정에서 각종 후 처리 기능을 이용해서 Job의 처리 결과에 따라 다양한 처리를 사용자 개입 없이 자동으로 수행되도록 정의하는 기능을 제공합니다

작업 결과 후행 처리

구성 및 특징

1 특정 오류 발생시 Skip처리

- 지정된 예외가 발생하면 자동 Skip처리 하여 정상적으로 다음 작업을 진행할 수 있도록 처리하는 기능

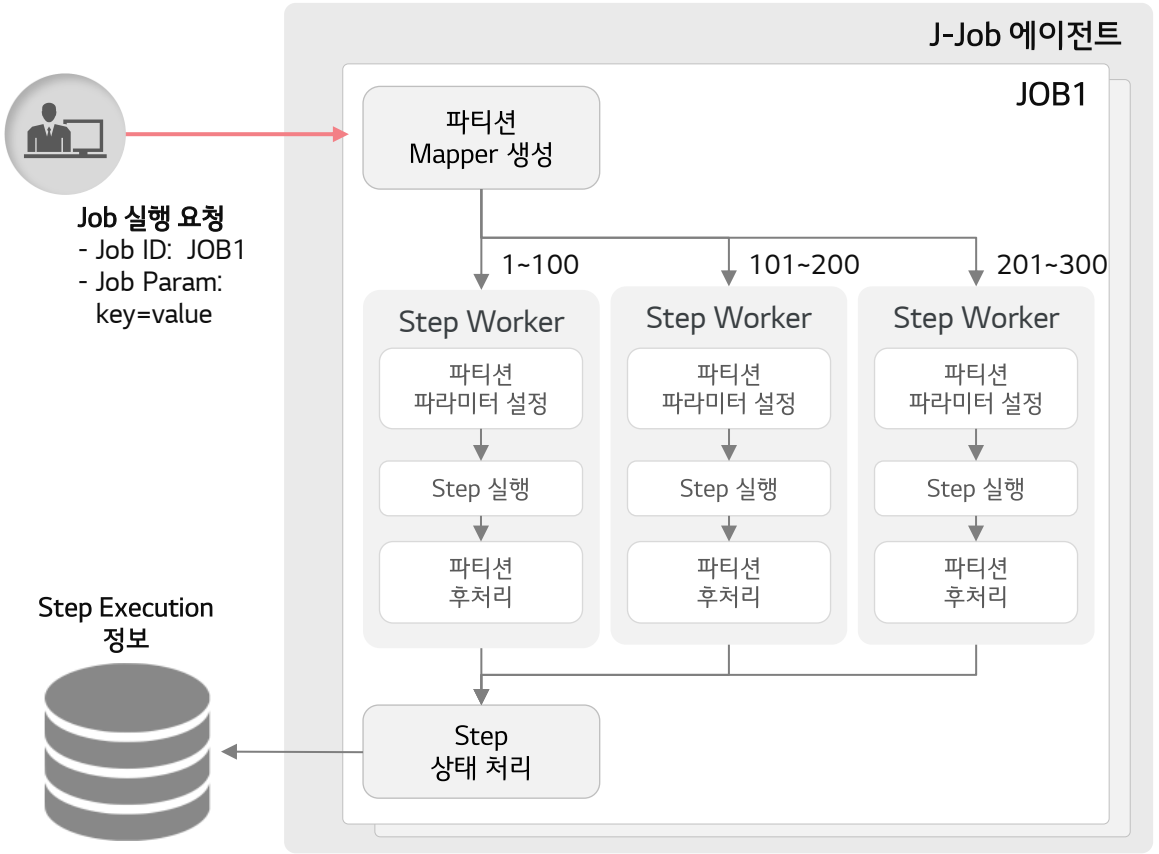
2 출력 유형 및 출력 값 비교

- 작업 시 발생되는 출력내용을 저장 할 수 있고, 이때 발생되는 출력 내용 중 일부를 추출 할 수 있음
- 추출된 값을 비교하여 후행 처리시 연계 가능

실행 – 대용량 데이터 처리 (파티션)

Step 파티션 기능을 이용하면 하나의 Step을 파티션을 나누고 Multi-Thread로 동시에 처리하여 대량건의 데이터를 처리하는 시간을 효과적으로 단축할 수 있습니다.

파티션 처리 메커니즘



구성 및 특징

파티션 Mapper

- 파티션 Mapper가 설정된 Step은 Mapper에 의해 파티션을 나누고 각 파티션에 파라미터를 할당

병렬처리

- Step Worker가 각자 파티션 파라미터를 이용해 Step을 동시에 수행
- 모든 파티션의 처리가 끝나면 해당 파티션의 상태를 처리하고 다음 스텝으로 진행

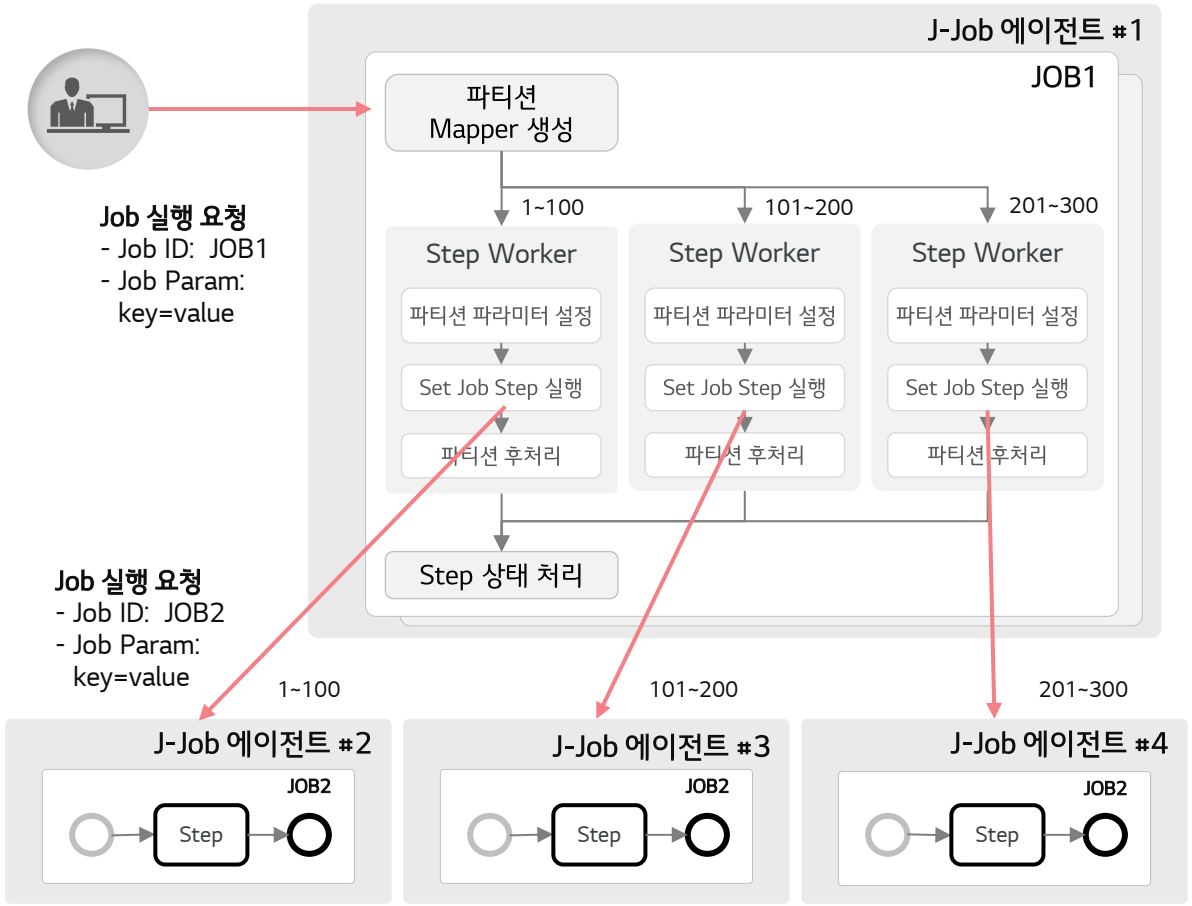
파티션 후처리

- Reducer를 통한 파티션 Lifecycle 제어 가능
- Collector를 통한 파티션 중간 결과 정보 전달
- Collector에서 전달한 정보를 수집하는 Analyzer를 통한 파티션 사용자 정의 종료 상태 정의

실행 – 대용량 데이터 처리 (Multi-Node 파티션)

Step 파티션 기능과 Set Job 모듈을 결합하면 대량건을 처리하는 Step을 구간을 나누어 여러 에이전트에서 분산하여 처리 할 수 있습니다. Set Job 모듈은 호출할 Job ID를 파라미터로 전달 받아 새롭게 Job 실행을 요청합니다.

Multi-Node 파티션 처리 메커니즘



구성 및 특징

파티션 Mapper

- 파티션 Mapper가 설정된 Step은 Mapper에 의해 파티션을 나누고 각 파티션에 파라미터를 할당

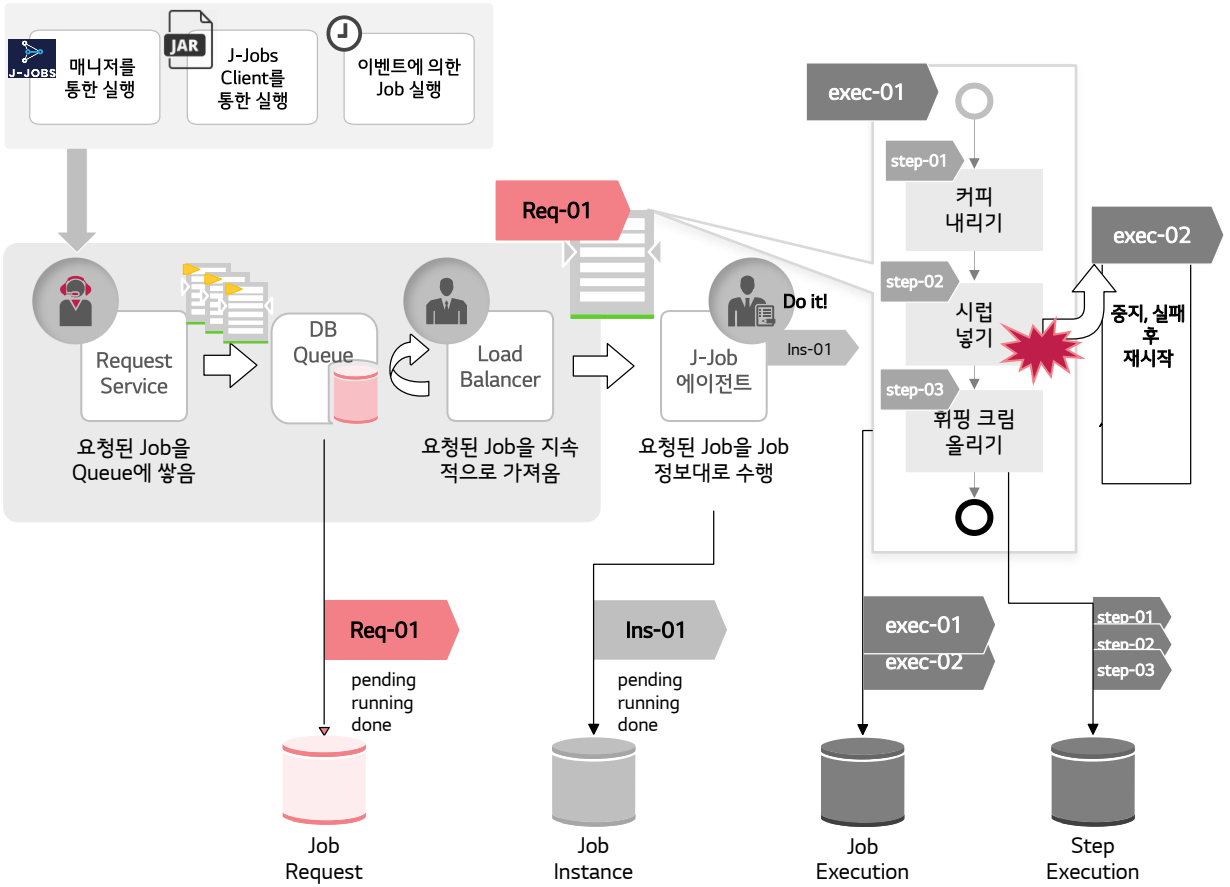
Multi-Node 병렬처리

- Step Worker가 각자 파티션 파라미터를 이용하여 Step을 동시에 수행
- Set Job Step의 경우 요청되는 Job은 다른 에이전트에서 실행 가능
- Set Job에서 자식 Job 은 비동기로 요청되지만 상태를 주기적으로 체크하여 자식 Job이 완료가 될 때까지 대기
- 모든 파티션의 처리가 끝나면 해당 파티션의 상태를 처리하고 다음 스텝으로 진행

실행 – 이력관리

Job의 요청, 실행, 재실행, Step의 실행 등을 관리하여 각각의 레벨 단위로 세부적인 상태를 조회할 수 있습니다.

실행 흐름



구성 및 특징

Job Request

- 요청된 작업을 DB Queue에 적재하는 역할을 수행
- 상태를 Running, Pending, Error, Complete로 관리

Job Instance

- 에이전트에 할당된 작업을 실행하기 전의 상태이며 하나의 요청에 하나의 Instance가 생성

Job Execution

- Instance 내에서 작업이 실제 실행될 때의 상태 값 관리
- 중지나 실패 후 재처리를 하게 되면 이 Job Execution 의 값이 증가

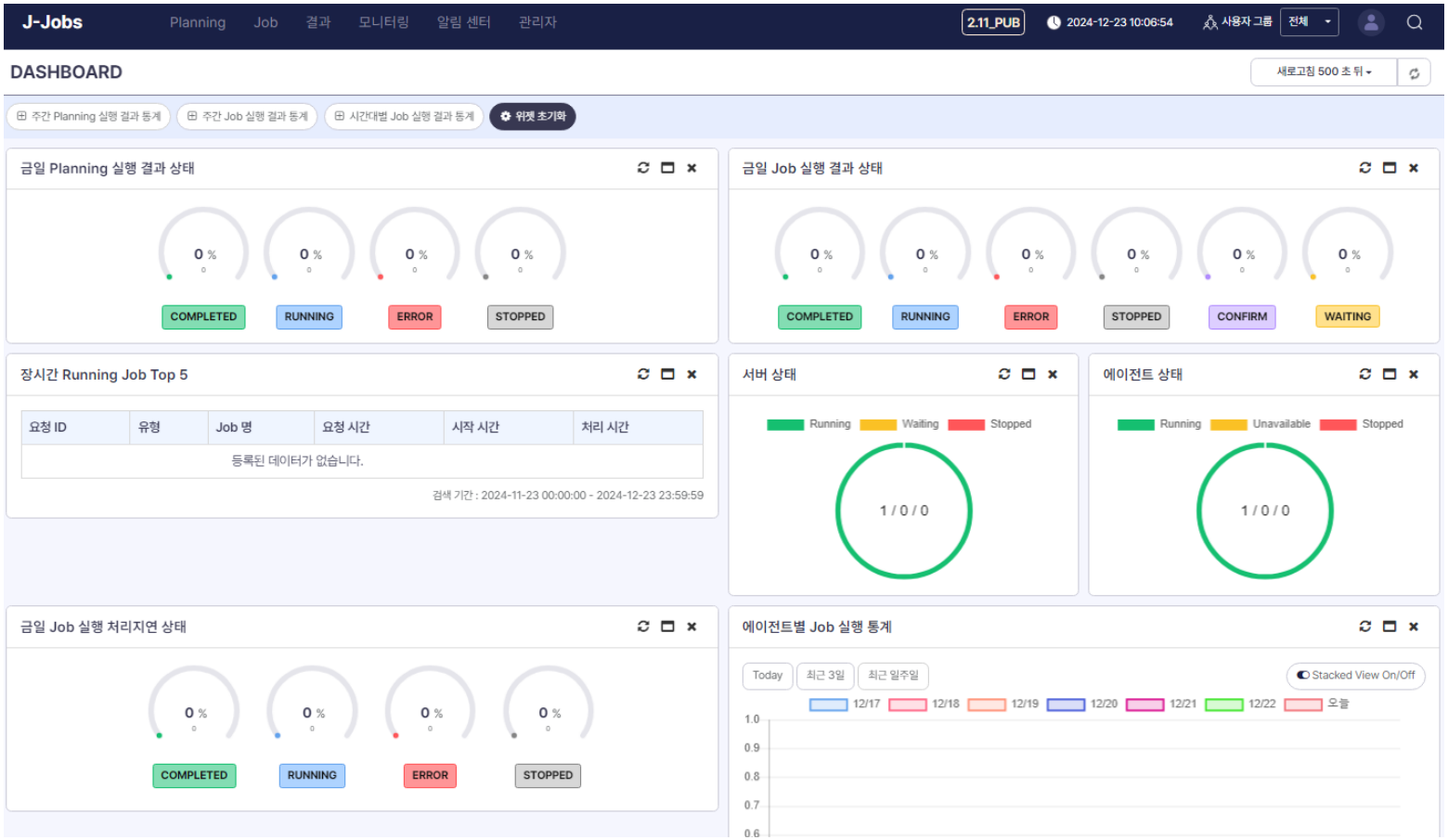
Step Execution

- Job 내의 여러 Step에 대한 실행 상태를 관리

모니터링 – 통합모니터링

J-Jobs 가 설치된 서버 및 에이전트의 상태, 현재 처리 중인 작업의 상태 및 처리된 Job의 통계 기능을 제공합니다.

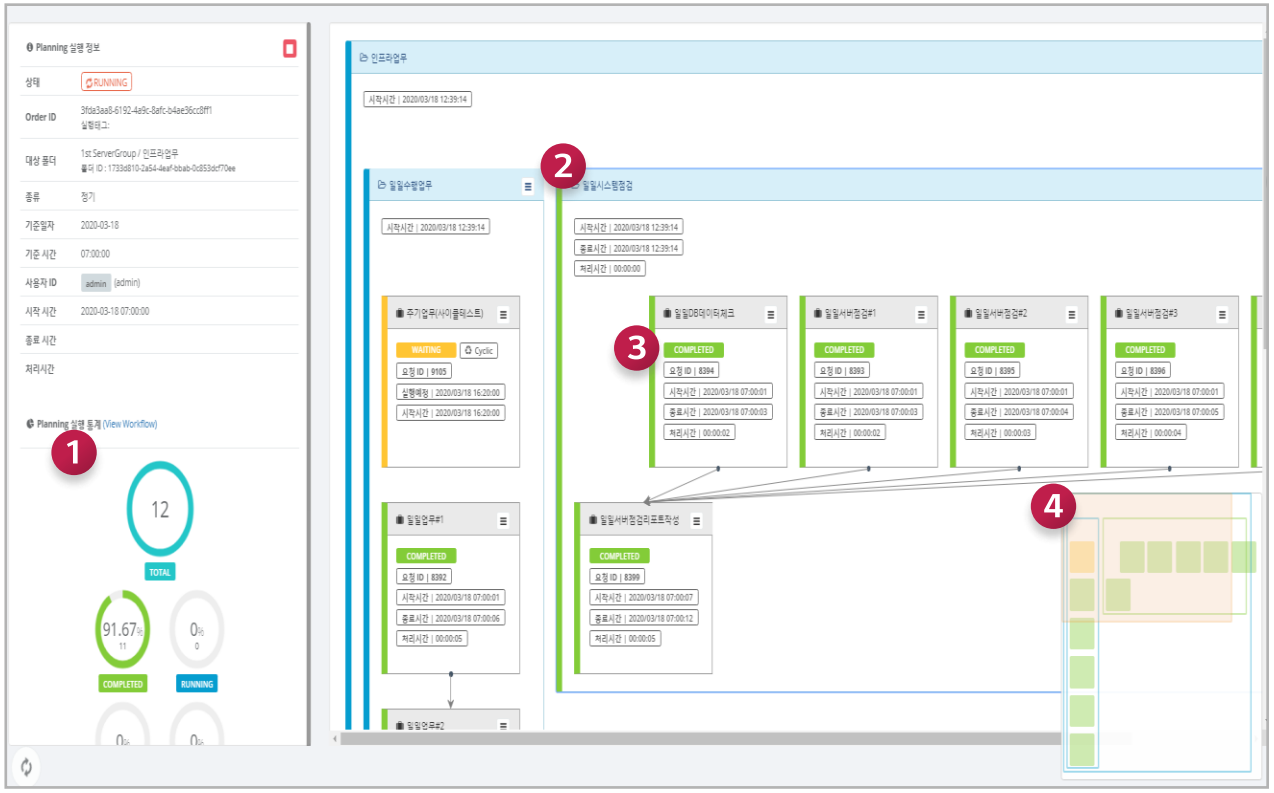
J-Job 매니저 모니터링 화면



모니터링 – 실시간 작업 처리 현황

기준일자(Order Date)에 처리되어야 할 작업처리 상태를 실시간으로 확인이 가능하여 어느 단계에서 문제가 발생하였는지 신속하게 확인이 가능합니다.

실시간 작업 처리 모니터링



구성 및 특징

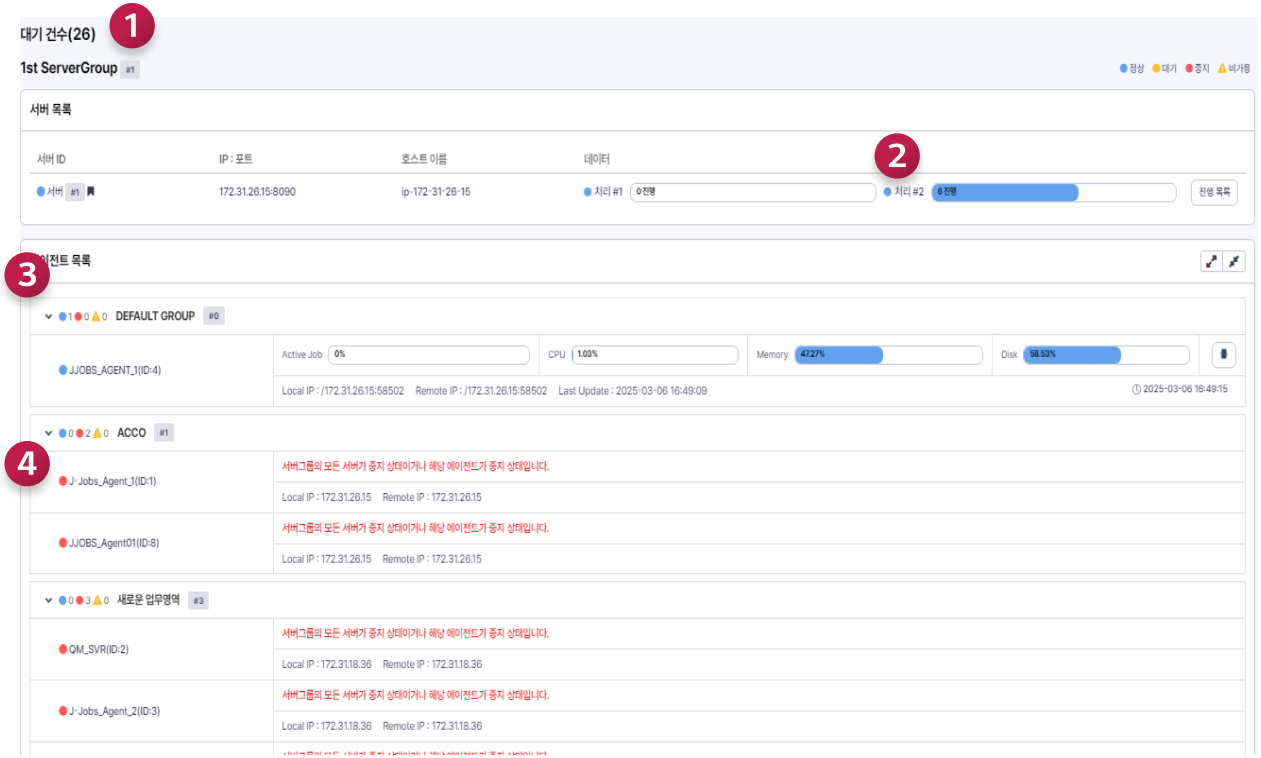
작업현황모니터링

- Visual하게 현재 작업의 선/후행 정의에 따라 처리되는 흐름을 직관적으로 확인 할 수 있음
- 현재 진행중인 작업 진행률 표시 1
- 각 상태 별 진행상황을 볼 수 있어 오류 발생시 즉시 확인 가능
- 폴더(그룹)내의 작업의 진행상태 표시 2
- 폴더의 상태로 내부 작업의 진행상태를 파악 할 수 있음
- 개별 작업의 진행상태 3
- 전체 현황을 볼 수 있는 MAP 4

모니터링 – 실시간 서버 및 에이전트 상태

J-Jobs는 실시간으로 서버 및 에이전트의 상태 및 진행되는 작업 상태와 자원 사용률을 모니터링 할 수 있습니다.

실시간 작업 처리 모니터링



구성 및 특징

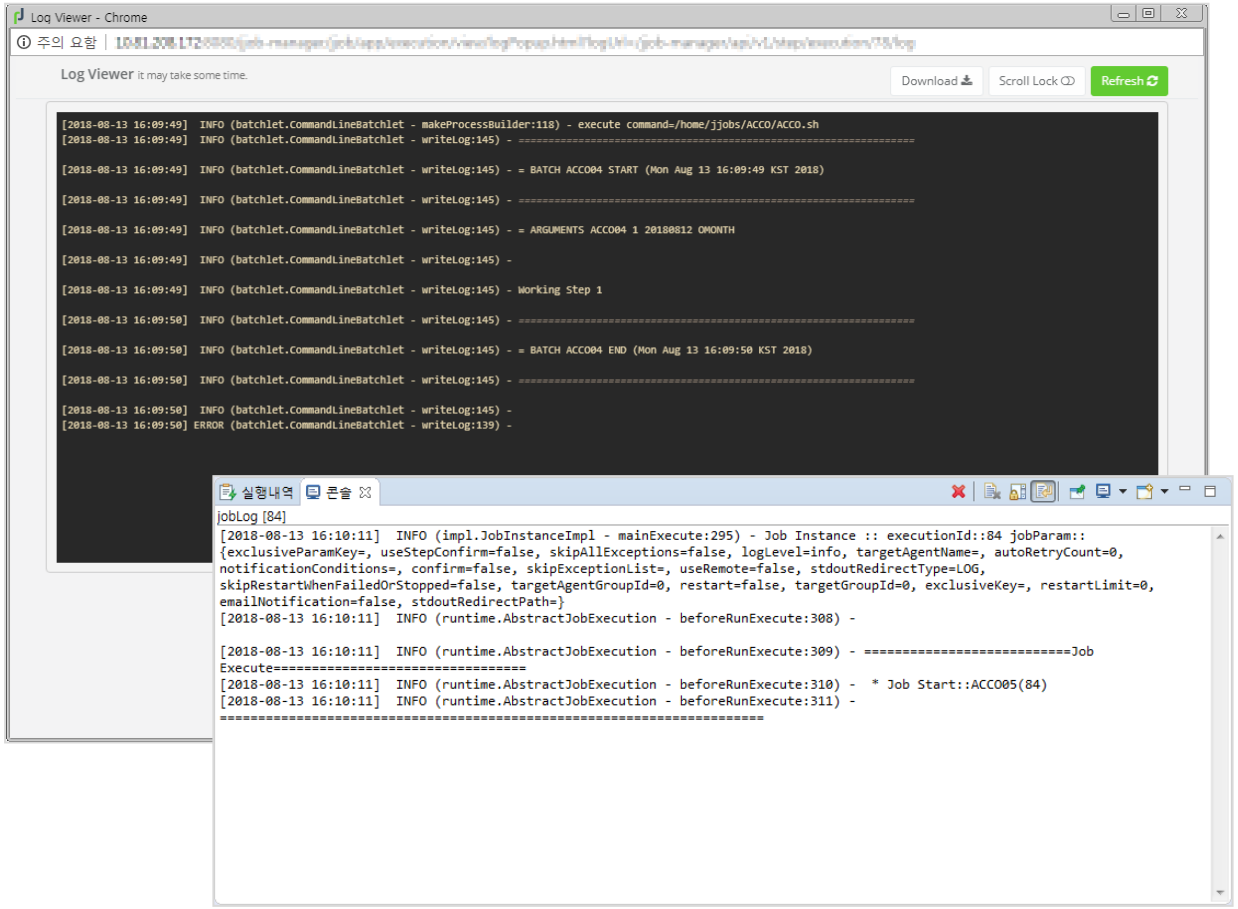
서버 및 에이전트 모니터링

- 서버 및 에이전트 상태, 작업 및 자원 사용률을 Visual하게 모니터링 할 수 있음
- 현재 작업 대기 건수 표시 1
- 서버의 상태 및 현재 서버가 처리하고 있는 작업 건수 표시 2
- 정상적인 에이전트가 작업을 수행하고 있는 작업 및 자원 사용률 표시 3
- 비정상적인 에이전트 상태 표시 4

모니터링 – 실시간 로그 조회

J-Jobs는 배치작업 수행 시 발생되는 log를 실시간으로 취합하고 이를 통합된 웹 매니저에서 확인 할 수 있습니다.

실시간 로그 확인



구성 및 특징

실시간 로그 뷰어

- 실시간으로 작업 로그 확인 가능한 전용 로그 콘솔 및 뷰어 제공
- 로그 레벨 설정 가능
  - TRACE
  - DEBUG
  - ERROR
  - WARN
  - INFO

실시간 제어 – 작업 제어

J-Jobs는 실시간 제어 기능을 제공하여 현재 진행중인 작업의 현황을 확인하여 상황에 맞게 실시간 제어를 수행할 수 있습니다.

실시간 작업 제어 기능

구성 및 특징

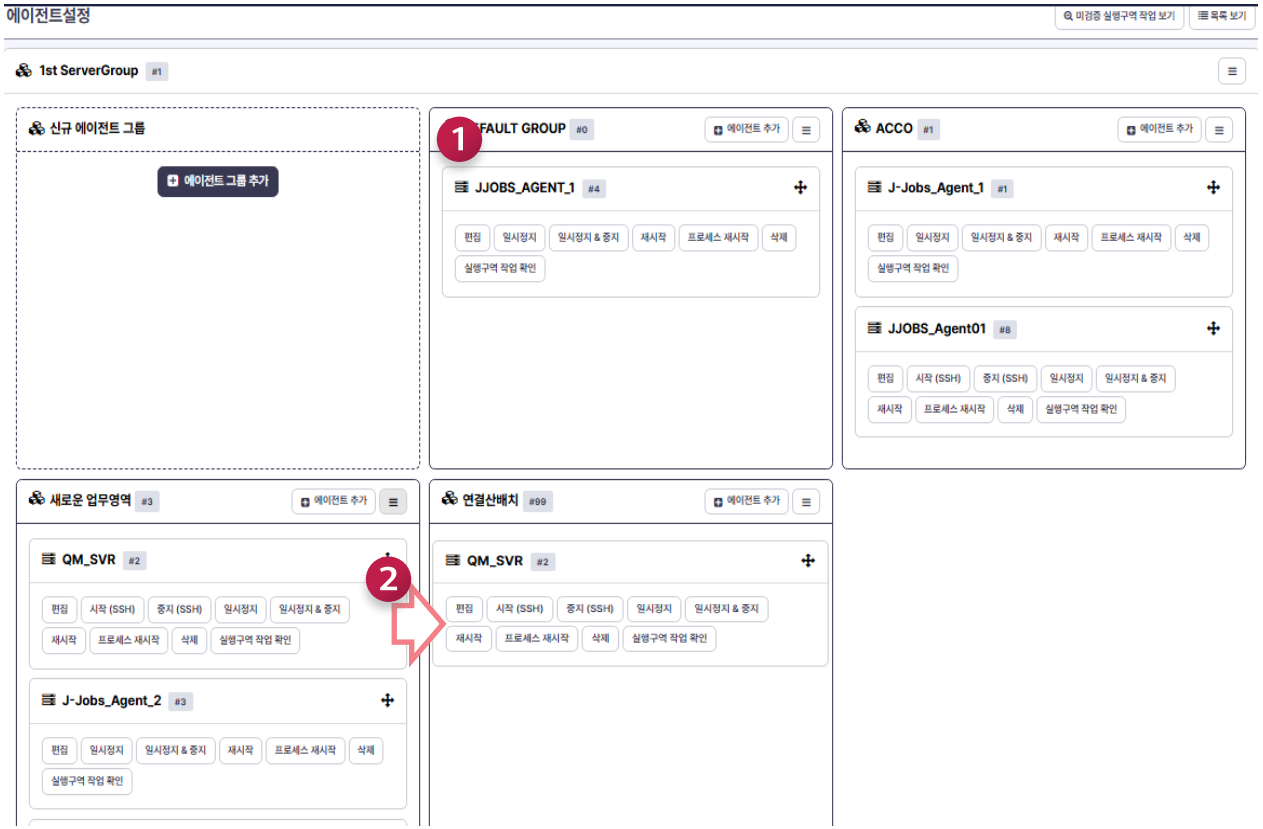
실시간 작업 제어

- 현재 진행 중인 작업을 실시간으로 제어 할 수 있음
- 중지 1
  - 진행중인 작업을 중단할 필요가 있는 경우
- 재시작 2
  - 오류가 발생하거나 중단 되었을 때 재시작
- 넘어가기 3
  - 오류가 발생하거나 중단 되었을 때 무시하고 다음으로 넘어가기
- 대기 4
  - 작업을 수행하지 않고 대기
- 파라미터수정 5
  - 입력된 파라미터 수정이 필요할 경우

실시간 제어 – 에이전트 제어

J-Jobs는 에이전트에 직접 접속하지 않더라고 에이전트 기동, 중지, 재기동 및 그룹이동을 쉽고 빠르게 실시간으로 제어 할 수 있어 장애 및 기타 상황에 빠르게 대응할 수 있습니다.

실시간 에이전트 제어



구성 및 특징

실시간 기동 제어

- 실시간으로 에이전트를 기동,중단 및 재기동을 수행 할 수 있음 1

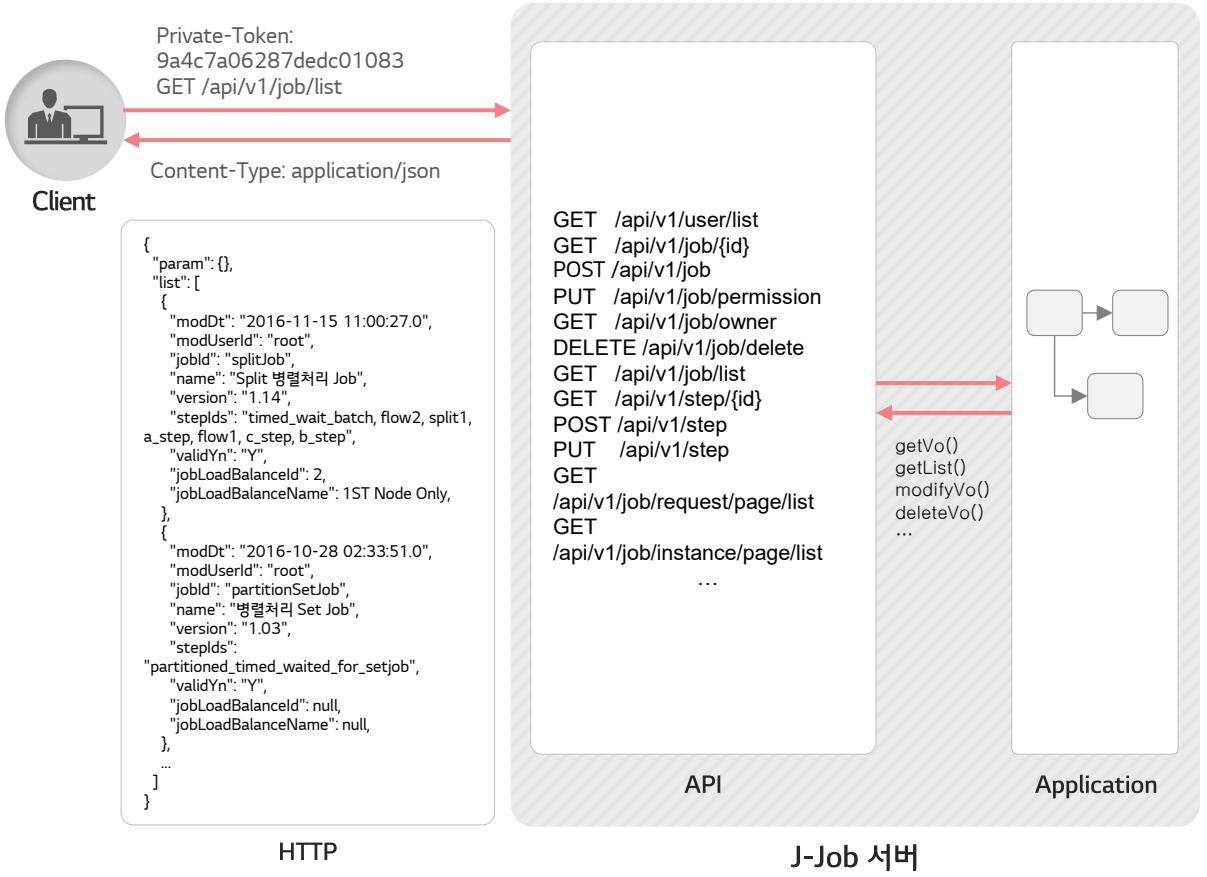
실시간 그룹 이동

- 특정 에이전트 그룹에 장애가 발생하거나 가용한 에이전트가 없는 경우
- 상대적으로 가용한 에이전트 그룹의 에이전트를 Drag & Drop으로 그룹을 이동시킬 수 있음 2
- 긴급 장애 복구 및 부하 분산에 사용할 수 있음

배포 및 연계 – RESTful API 를 통한 외부 시스템 연계

J-Job 서버에서 제공하는 RESTful API는 각종 조회부터 Job실행까지 다양한 서비스를 제공하고 있습니다. Http기반의 RESTful API를 제공하므로 외부 시스템과 쉽게 연계가 가능합니다.

RESTful API 처리 메커니즘



구성 및 특징

RESTful API

- HTTP로 전송하여 요청하면 JSON 타입으로 응답
- Private-Token 방식 인증
- 응답 결과 HTTP Status Code 사용

RESTful API 규칙 준수

- 직관적인 URI로 자원을 표현
- 자원에 대한 행위는 HTTP Method (GET, POST, PUT, DELETE)로 표현

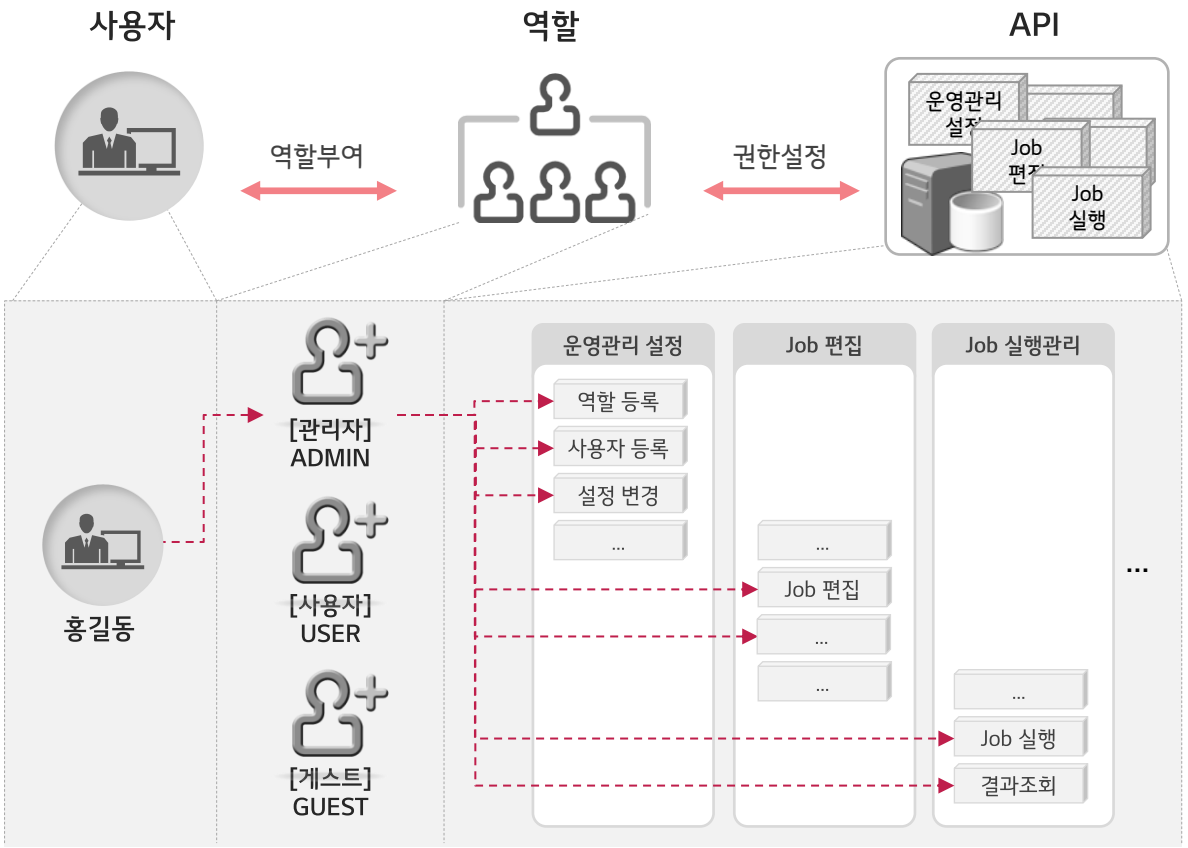
다양한 API 지원

- Manager 에서 보이는 모든 기능이 RESTful API
- Job 정보, Step 정보, 실행 이력 정보, 사용자 정보, 에이전트 정보, 스케줄 정보 등

권한 및 통제 – 사용자 관리와 권한 관리

역할을 관리하여 운영관리에 호출 API에 해당 역할의 권한을 부여하는 RBAC (Role Based Access Control) 방식으로 권한을 관리합니다.

권한 관리



구성 및 특징

API 권한 관리

- RBAC 기반
- 기본 역할 (root, administrator, user, guest) 외에 추가적으로 역할 등록 가능
- 각 사용자 별로 호출 가능한 API 매핑 별도 가능

그룹 및 사용자 연계

- LDAP, ACF2 등 사용자 및 기반 OS 연계를 하기 유연한 구조로 설계

최상위 관리자 (Root)

- 운영관리 설정
- 그룹, 사용자, 역할 관리

관리자 (Administrator)

- 서버 및 에이전트 설정
- 부하분산 설정 및 관리

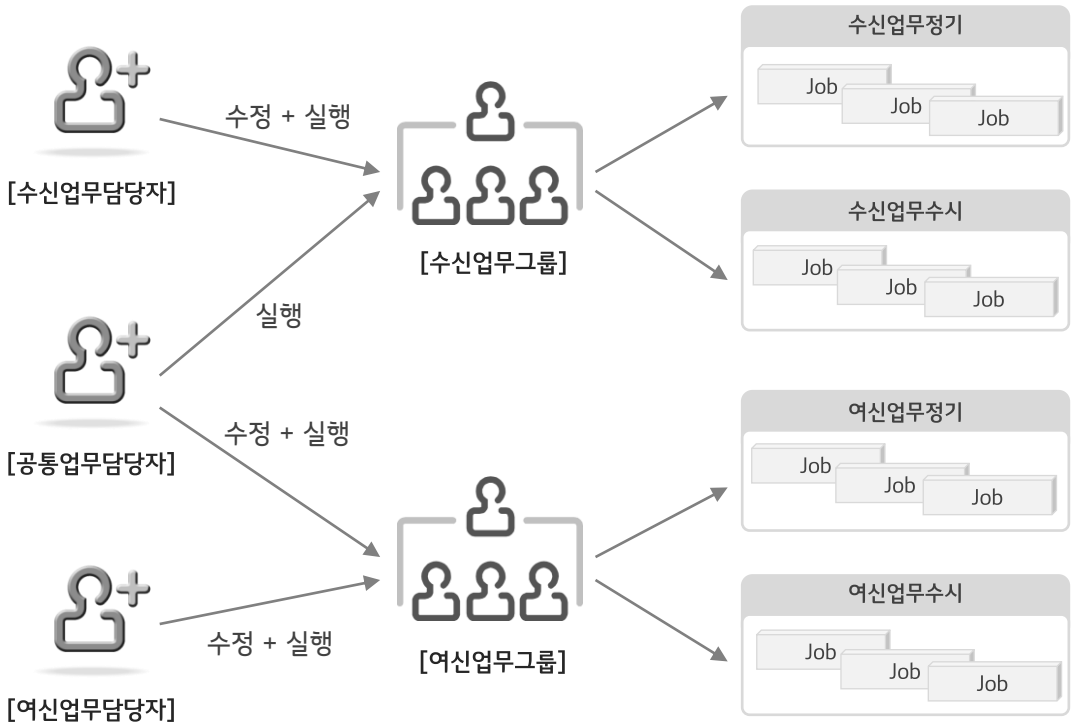
사용자

- Job 편집 및 실행
- 실행 결과 조회

권한 및 통제 – 사용자그룹

모든 Job은 특정 폴더(작업그룹)에 속하게 되며 특정 폴더는 하나의 사용자그룹과 매핑이 됩니다. 사용자그룹에는 다수의 사용자를 등록할 수 있으며 등록된 사용자는 폴더에 속한 Job을 수정 및 실행할 수 있는 권한을 부여 받게 됩니다.

사용자그룹



구성 및 특징

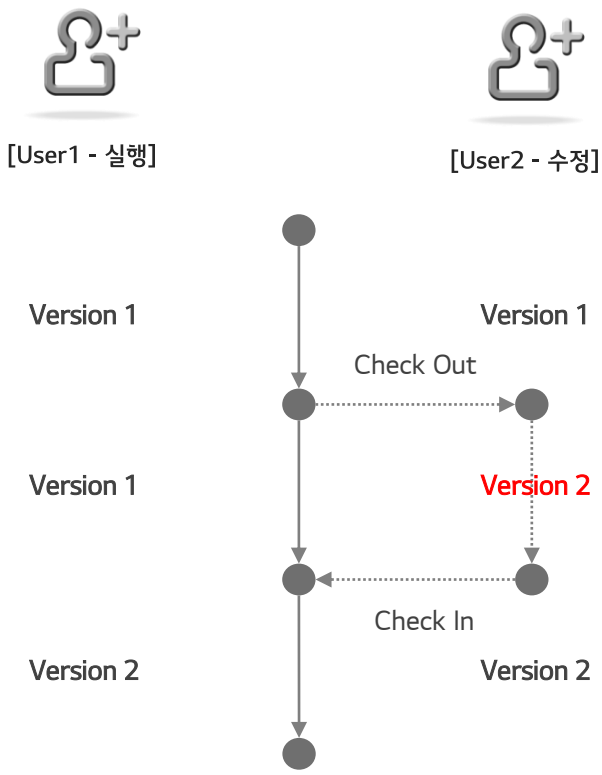
유연한 권한 관리

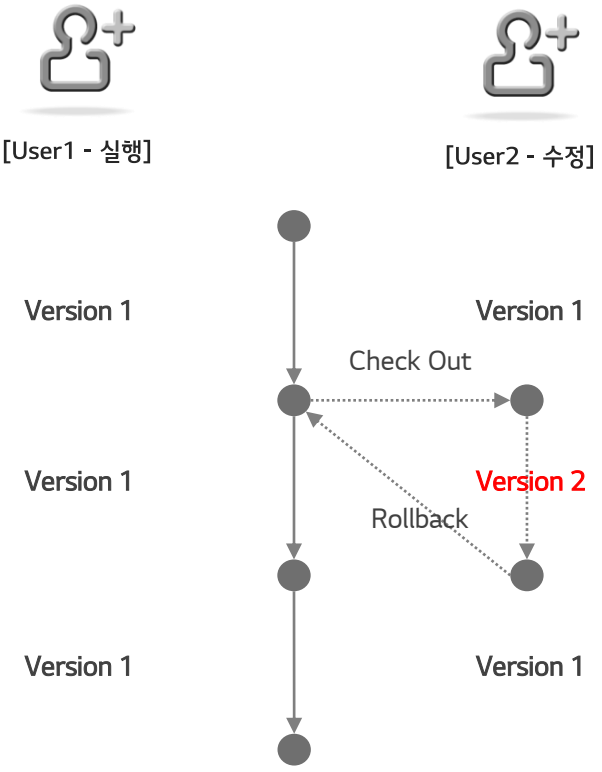
- 사용자 그룹을 통한 사용자 권한 관리
- 사용자를 다수의 사용자 그룹에 포함시킬 수 있는 구조로 한 사용자가 여러 Job을 쉽게 관리 할 수 있음
- 개별 Job에 권한 부여가 아닌 사용자 그룹을 통한 일관 자원관리
- Windows 권한 관리 개념을 참조하여 복잡하지 않고 간결함

권한 및 통제 – Check In/Out

폴더(그룹)단위 Job편집 시 권한이 있는 여러 사용자의 동시 변경작업을 방지하기 위하여 Check In/Out 기능을 제공하여 작업의 정합성을 보장합니다.

Check In/Out 처리 흐름

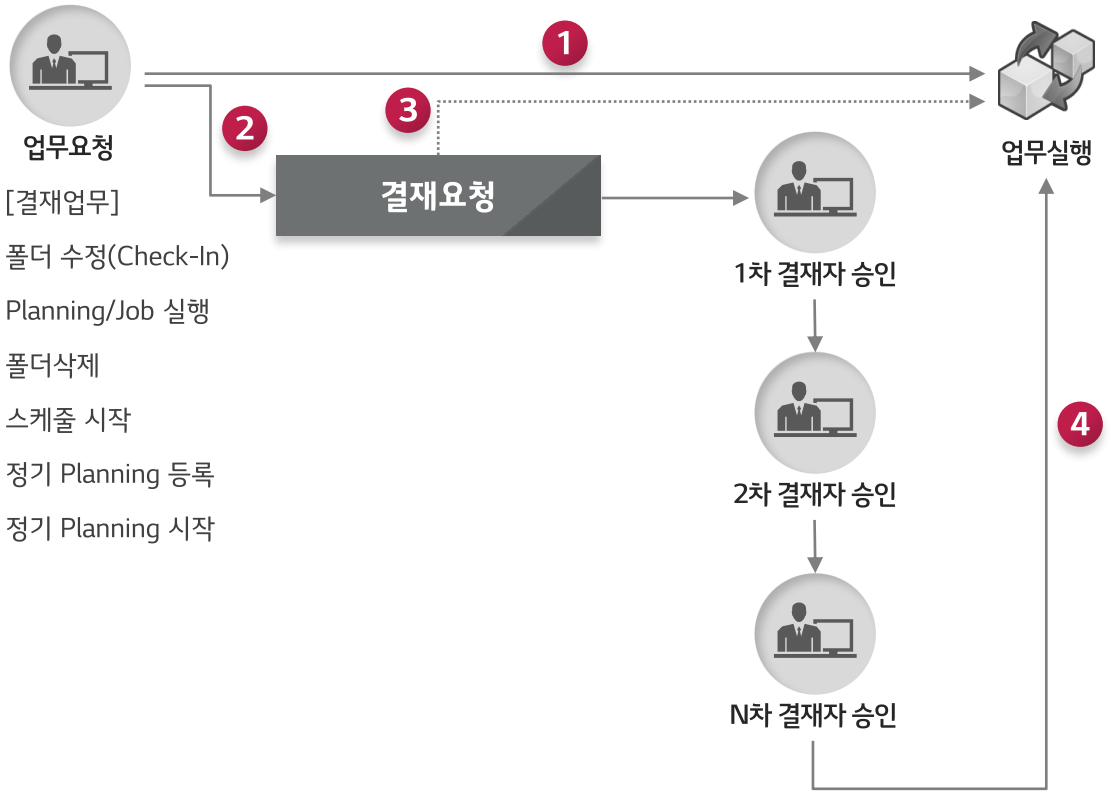




권한 및 통제 – 결재 기능

J-Jobs는 작업수정 및 실행과 같이 수행 시 민감한 업무를 결재기능을 통하여 통제할 수 있는 기능을 제공하고 있습니다. 결재기능을 이용하면 다중 승인권자의 확인을 통한 비 인가된 작업 또는 잘못된 작업 수행을 통제할 수 있고 향후 감사에 활용할 수 있습니다.

결재처리 흐름



구성 및 특징

결재 Feature 기능

- 결재 Feature 설정에 따라 결재기능을 켜고 끌 수 있음
- 결재 Feature가 비활성화 된 경우 즉시 실행됨 1
- 결재 Feature가 활성화 된 경우 결재요청이 수행됨 2

결재요청

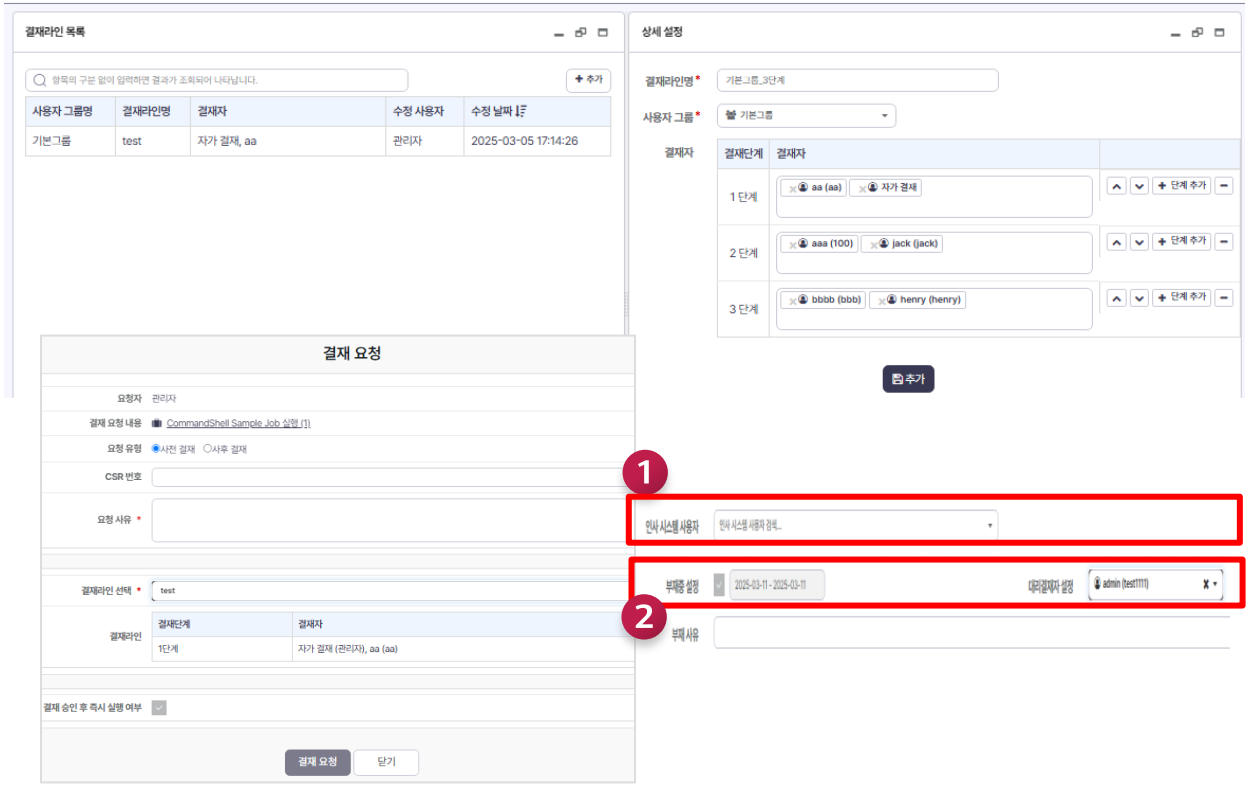
- 사후처리로 결재요청을 수행하면 즉시 업무가 실행되고 실행 이후 결재자에게 통지 3
- 사전처리로 결재요청을 수행하면 결재라인의 모든 결재자의 승인이 되어야 업무가 수행됨 4

권한 및 통제 – 결재기능 (계속)

J-Jobs는 결재를 위한 결재선 관리, 결재시간 관리, 부재중 대리결재, 사전/사후 결재 및 인사시스템 연동기능 등 다양한 기능을 제공하고 있습니다.

결재기능

결재라인



구성 및 특징

결재라인

- 사용자 그룹별 여러 개의 결재라인을 관리할 수 있음
- 결재라인 1단계 이상으로 관리 할 수 있으며, 자가결재 및 병행결재 기능으로 결재대기시간을 단축할 수 있음

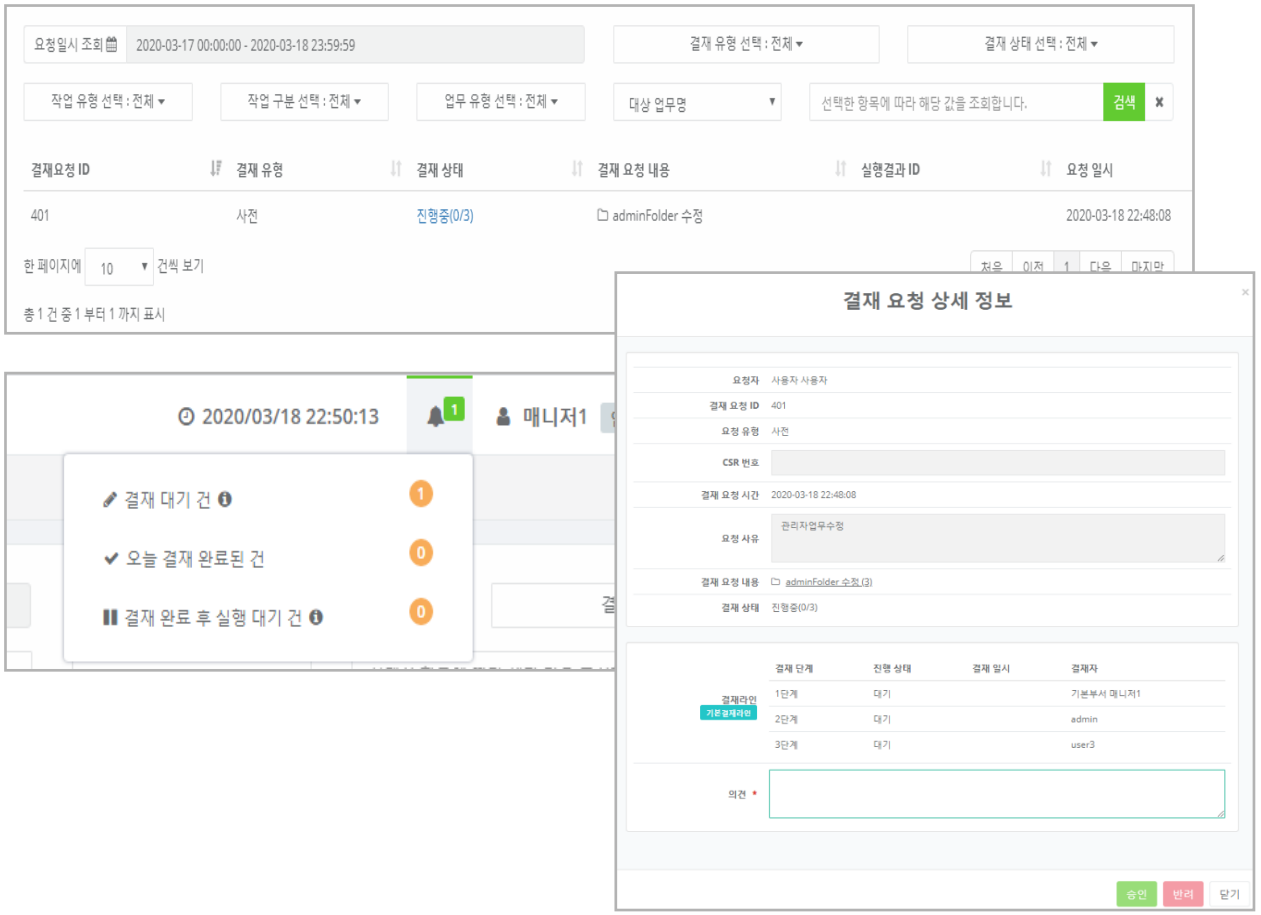
사용자관리

- J-Jobs 사용자는 인사시스템 사용자와 연계하여 추가 정보를 사용할 수 있음 1
- 부재중 설정 시 지정된 대리결재자로 결재라인이 자동으로 변경 됨 2

권한 및 통제 – 결재기능 (계속)

J-Jobs는 결재를 위한 결재선 관리, 결재시간 관리, 부재중 대리결재, 사전/사후 결재 및 인사시스템 연동기능 등 다양한 기능을 제공하고 있습니다.

결재기능



구성 및 특징

결재이력관리

- 결재요청 및 승인대한 모든 이력을 조회 할 수 있음

결재알림

- 웹 콘솔의 알림으로 결재에 관련된 진행상황을 확인 할 수 있음
- J-Jobs 알림서버기능에 알림서버 매체를 등록하면 Email 또는 SMS로 통지 처리가 가능함

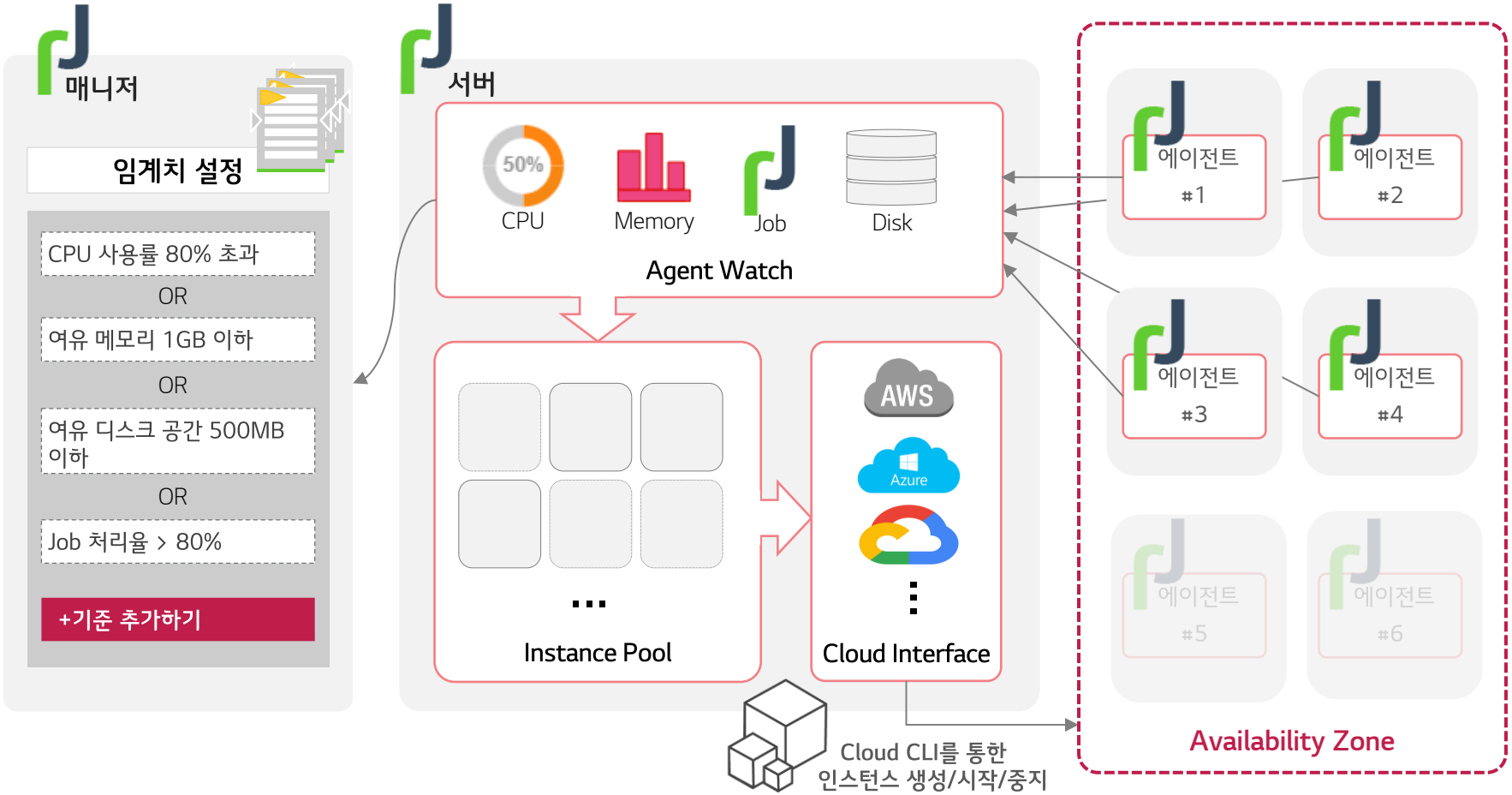
결재승인

- 결재라인에 설정된 결재자 순서에 따라 승인이 진행되고 최종단계 결재자의 승인전까지는 작업이 실행되지 않음
- 결재요청내용을 통하여 승인자는 상세한 변경 및 실행내용을 확인 할 수 있음

Cloud 기반 Auto Scaling 기능

J-Job 에이전트의 상태를 실시간 측정하여 사용자가 정의한 임계 기준(Job 처리율, CPU 사용량, 여유 Memory, Disk 공간)에 따라 에이전트가 자동으로 Scale In/Out 기능을 제공합니다.

J-Jobs Auto Scaling 아키텍처

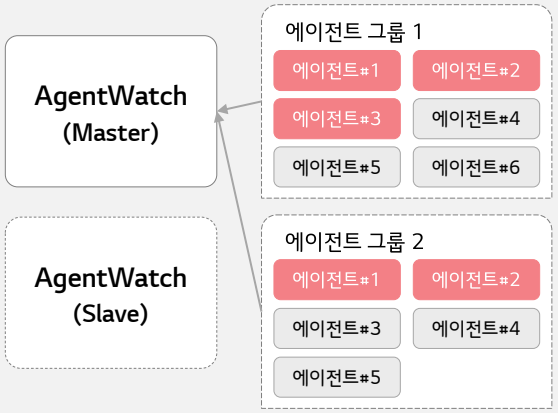


Cloud 기반 Auto Scaling 기능 (계속)

Agent Watch

"J-Job 에이전트들의 자원을 모니터링 하고 사용자가 설정한 기준에 따라 **에이전트의 Auto Scaling을 수행**한다."

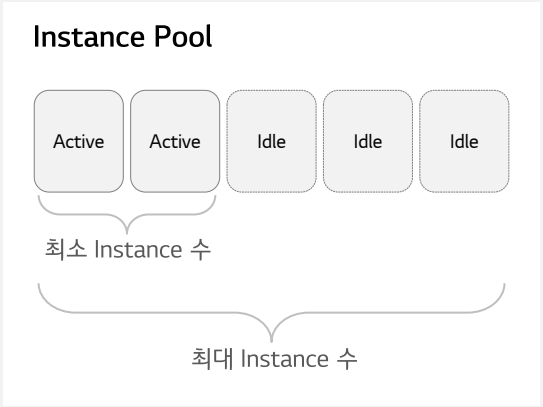
- 각 에이전트와 TCP 통신을 통한 실시간 자원 상태 모니터링
- 사용자 정의 임계치에 따라 가용 Agent를 구분
- 에이전트 그룹 내 최소 가용 에이전트를 유지하도록 Scale Out 혹은 Scale In



Instance Pool

"Cloud Interface와 연동하여 Pool 방식으로 인스턴스를 관리하고, **Agent의 기동 상태를 제어**한다."

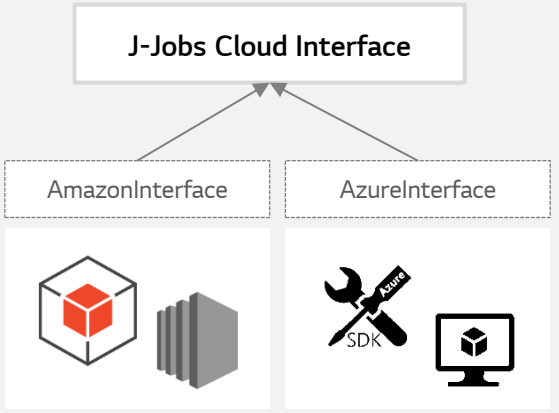
- Agent Watch로부터 Scale In/Out 요청에 따라 Instance 자동 생성 및 삭제
- 한번 생성된 인스턴스들을 Pool에서 일정 기간 동안 유휴 상태로 관리
- 매 Scale Out 요청 시, 유휴 인스턴스 재활용으로 인해 신규 인스턴스 생성 보다 빠른 Agent 기동 시간 확보 가능



Cloud Interface

"AWS(Amazon Web Service) 및 Azure 등 **다양한 클라우드 동작 수행**"

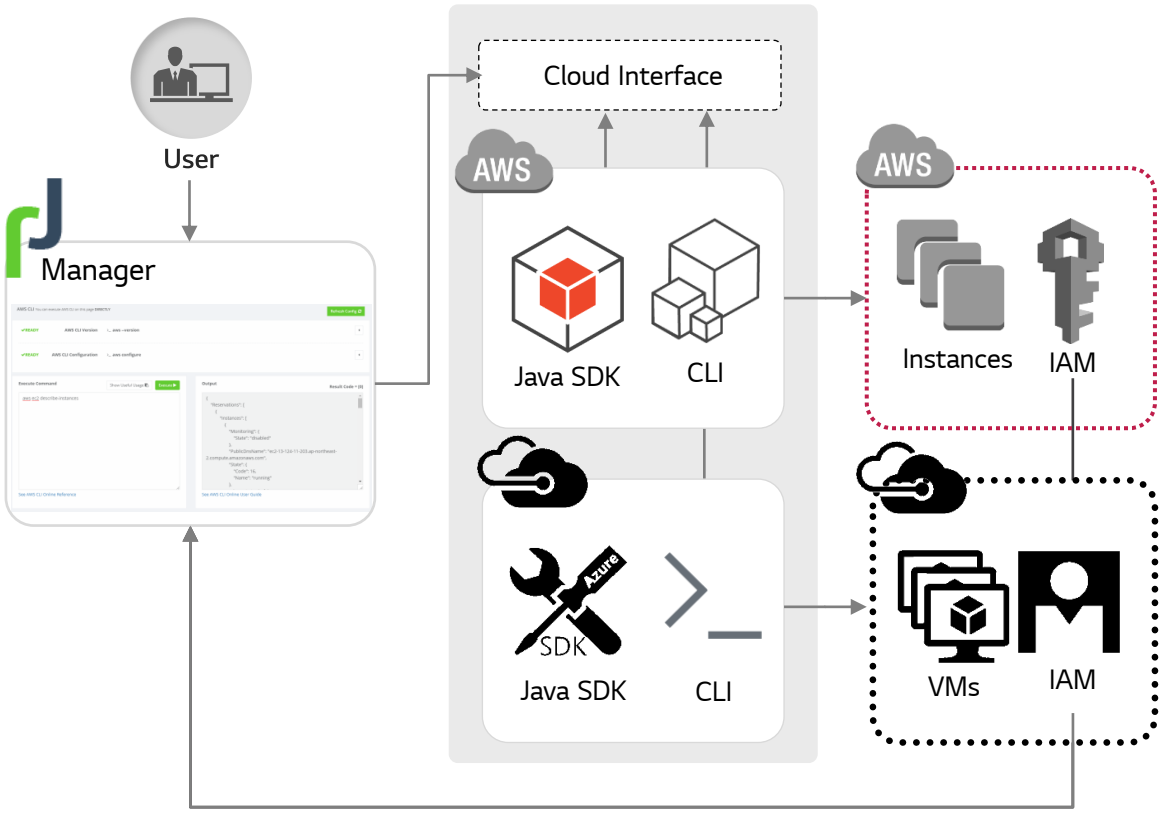
- 인스턴스 기동/중지/제거등의 기본Action API 호출
- 기본 Action API에 없는 경우는 Custom CLI 직접 설정 가능
- Cloud Interface를 이용한 통합 클라우드 관리 화면 제공



Cloud 관리 기능

AWS EC2의 기본적인 관리화면을 제공하여 별도의 AWS Console EC2를 사용할 필요가 없습니다. Azure, GCP를 위한 통합 관리화면도 제공하고 있습니다.

Cloud 관리



구성 및 특징

AWS CLI 직접실행

- Manager가 설치되어 있는 인스턴스의 AWS CLI 를 직접 호출하여 테스트

계정 및 기본정보 관리

- AWS, Azure 등의 Cloud향 Account를 관리
- 인스턴스가 구동될 Region 정보
- 인스턴스 생성시 사용될 Template 또는 Image 정보 설정

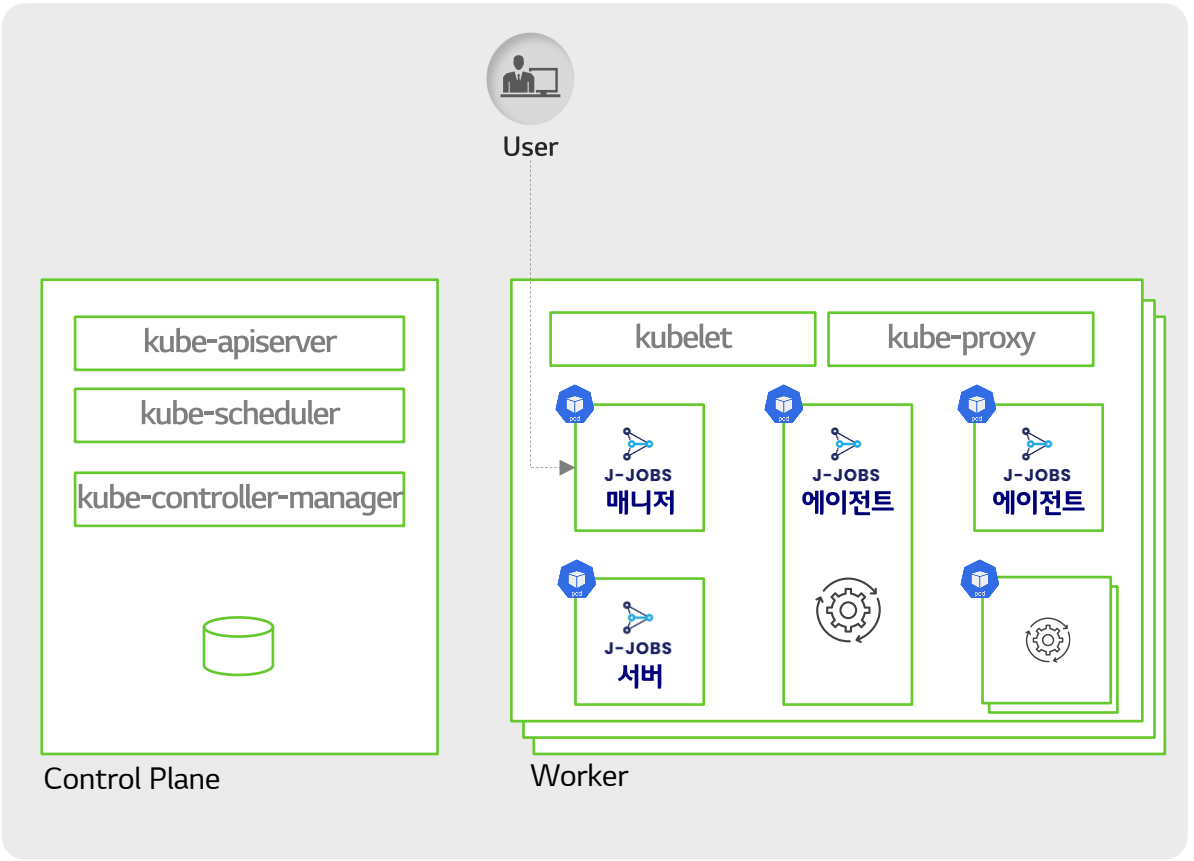
인스턴스 관리

- 인스턴스 시작, 정지, 중지(terminate)
- 인스턴스 생성(Launch)
- 목록조회

Fully Container 구성 지원

매니저부터 서버, agent, 배치 application 까지 컨테이너로 구성하여 관리할 수 있습니다.

컨테이너 환경 구성



구성 및 특징

아키텍처 일관성 유지

- 컨테이너 오케스트레이션 아키텍처에 최적화 구성으로 MSA 환경에 적합한 구성 가능
- VM/컨테이너가 혼용된 환경 뿐 만 아니라 Full Container 환경 지원
- 개발/테스트/운영 환경으로 배포체계 관리가 가능함

구성의 용이성

- Manager만 외부 통신을 허용
- Agent 증가나 업무 변경 시에도 컨테이너 환경에 대한 추가작업 없음
- 배치 pod에 대한 통합/분리, 병렬 구성 등 업무별 다양한 구성환경 지원

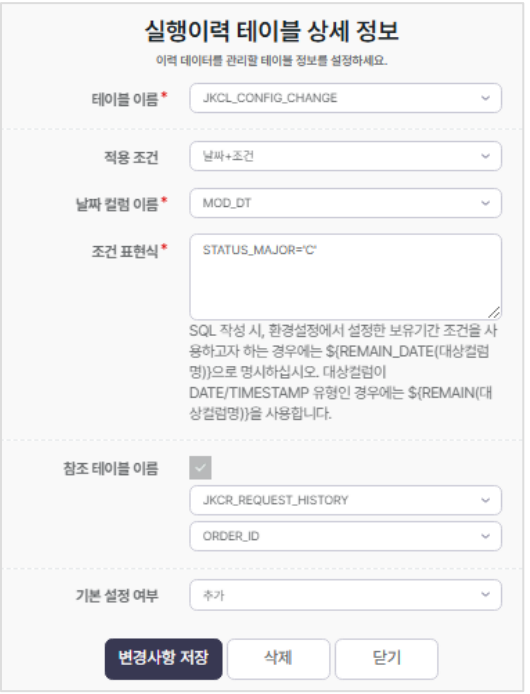
높은 장애 저항성

- 신속한 기동/정지로 불필요한 이중화 구성 지양
- Agent Pod에 어플리케이션을 탑재하여 신속한 배치처리 지원 가능
- 어플리케이션 pod를 병렬 수행하는 병렬 partitioning 지원

관리 - 데이터 백업 자동화

지속적으로 쌓이는 이력 데이터에 대하여 주기적으로 백업/삭제하는 기능 제공하여 데이터 증가에 따른 오류 및 성능 저하 현상을 방지 할 수 있습니다.

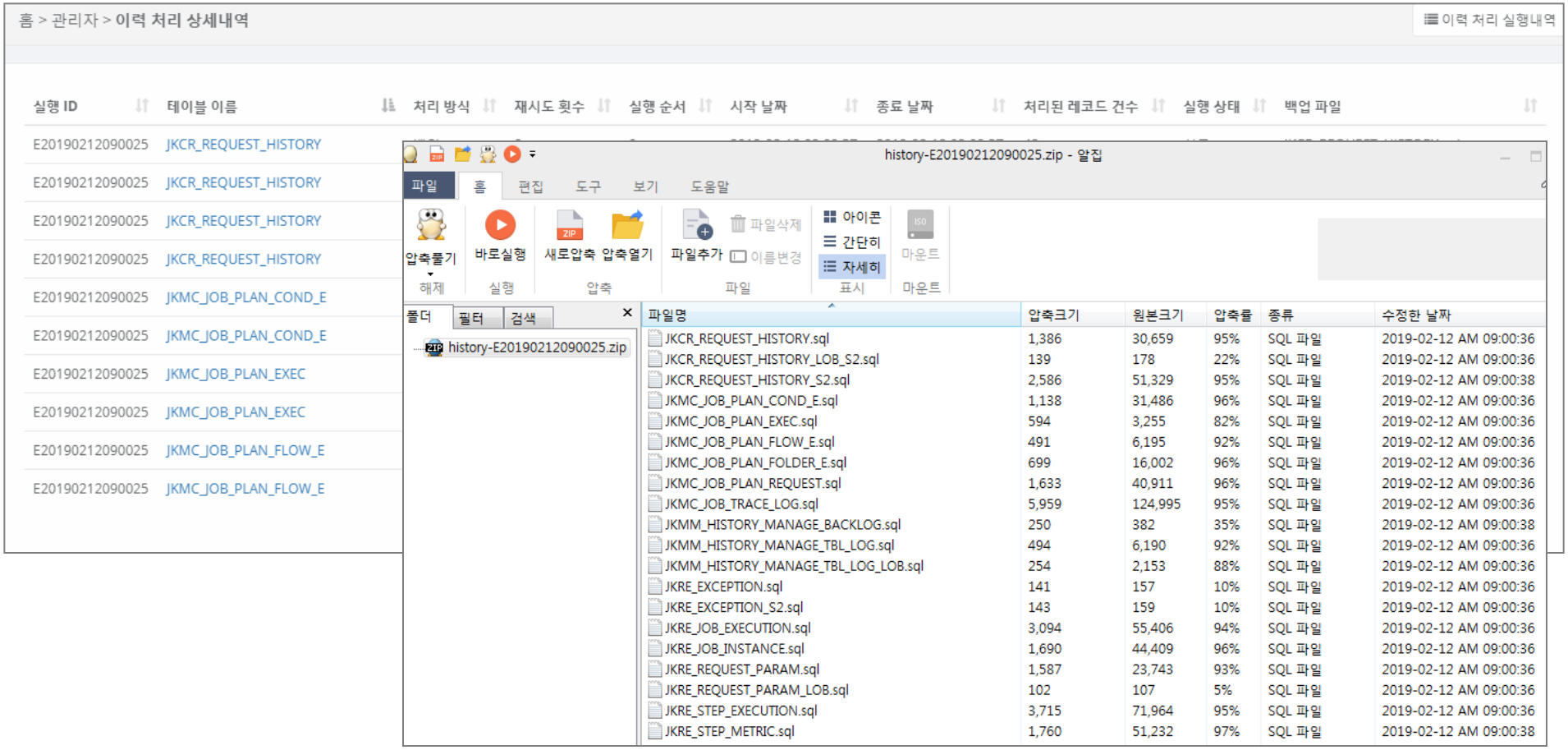
데이터 백업 자동화



관리 - 데이터 백업 자동화 (계속)

주기/수시로 실행된 이력 데이터 처리 내역 확인 기능 제공 및 백업파일 제공으로 향후 문제 발생시 해당 내역을 복구할 수 있습니다.

데이터 백업 이력

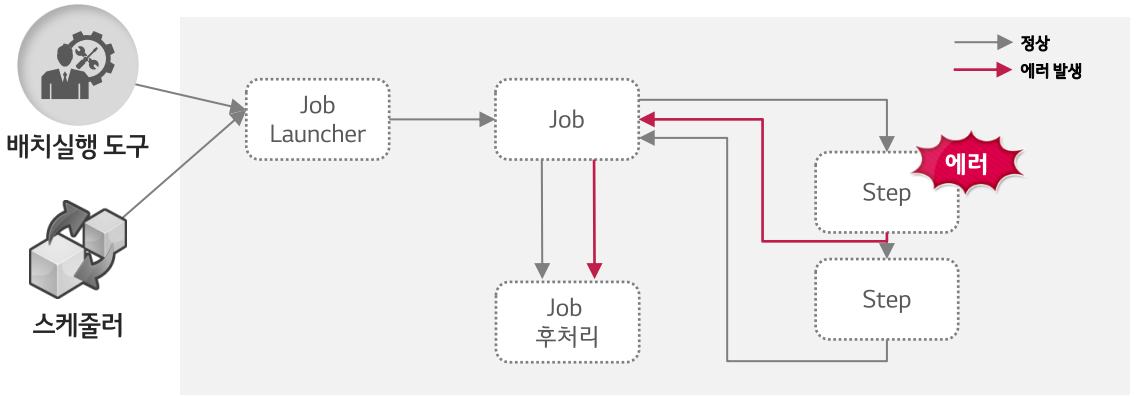


관리 - Job 실행 결과 알림 연계

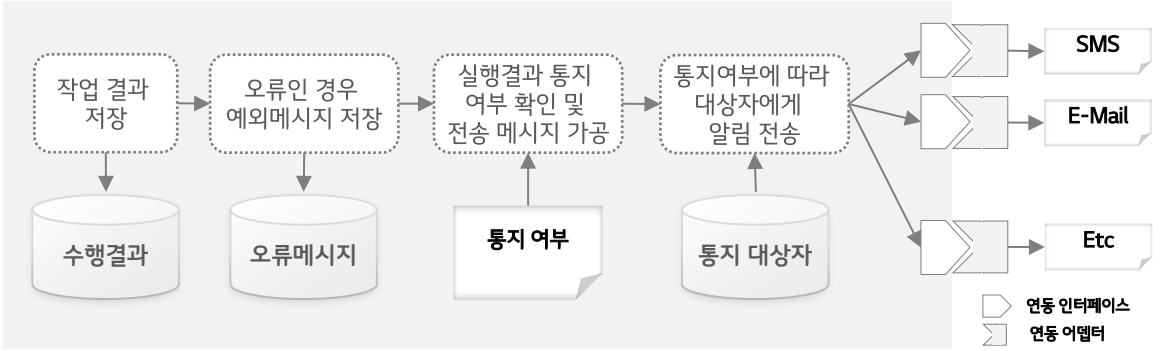
J-Jobs는 Job수행 후 처리된 결과를 메일 또는 SMS등으로 관리자에게 통지 할 수 있는 인터페이스를 제공합니다.

실행 결과 알림 개념도

[Job 실행 절차]



[Job 후처리 절차]



구성 및 특징

Job 후처리

- Job이 실행된 이후 결과를 처리하는 프로세스
- 오류인 경우 별도의 오류 메시지를 저장
- 결과 통지 여부에 따라 메시지 전송대상자에게 메시지가 전송 됨

통지 여부 및 대상

- 통지 여부는 Job 별 특성에 저장되고 알림 조건에 맞을 때 통지됨
- 대상자는 별도 DBMS에 저장하고 등록된 전송 매체 타입에 따라 메시지가 전송 됨

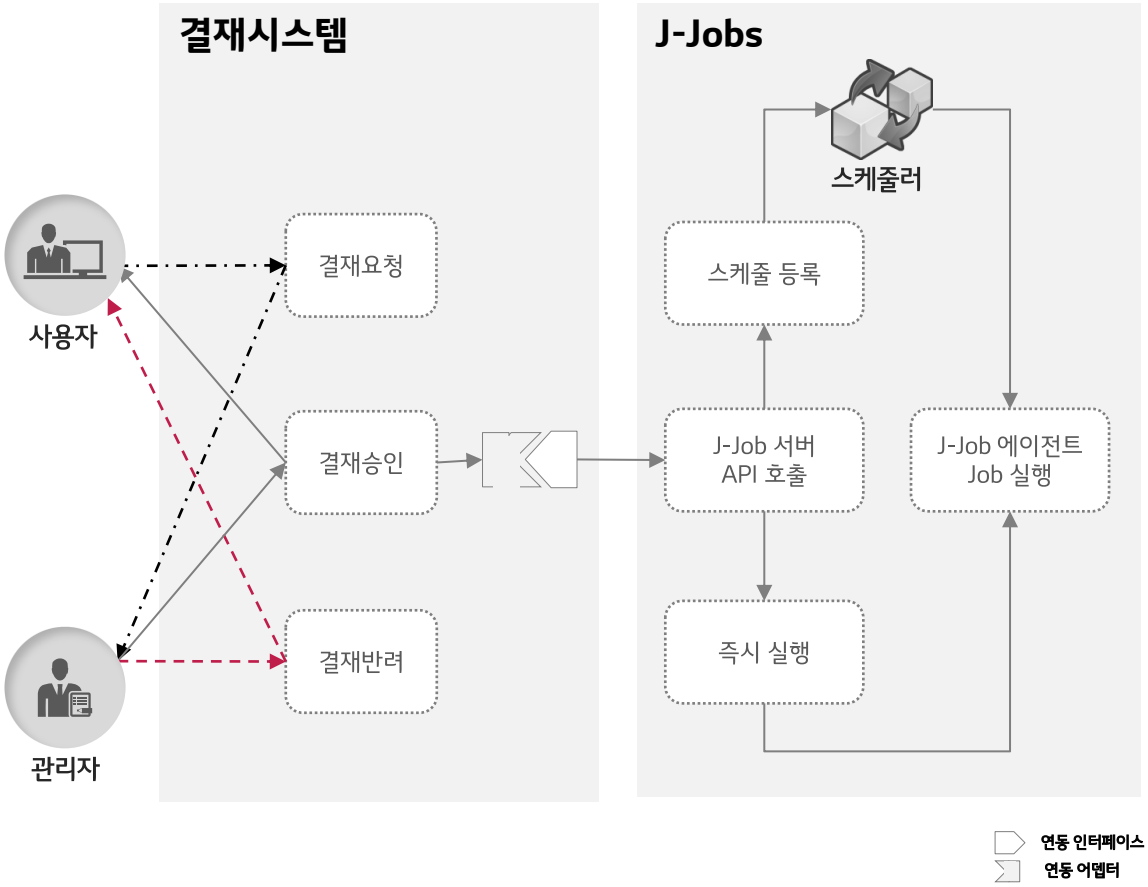
연동 인터페이스

- J-Jobs는 결과 메시지 전송을 위한 인터페이스를 제공
- 매체 타입에 따라 인터페이스를 사이트에 맞도록 연동 어댑터를 구현

관리 - 결재시스템 연계(내장된 결재 시스템을 사용하지 않을 경우)

J-Jobs는 운영환경에서 비 인가된 임의의 스케줄 등록 및 즉시 실행을 방지하기 위해서 결재시스템과 연계하여 승인된 작업만 등록 및 실행될 수 있도록 연계 기능을 제공합니다.

결재시스템 연계 메커니즘



구성 및 특징

결재시스템

- 형상관리나 기타 작업에 대한 결재 프로세스가 제공되는 시스템
- J-Jobs가 제공하는 인터페이스와 연계 필요
- 결재승인 시에만 J-Jobs 연계 인터페이스를 통한 요청 전달

결재연계 인터페이스

- J-Jobs는 결재시스템과의 연계를 위한 별도의 인터페이스를 제공
- 인터페이스는 J-Jobs와 연계를 위하여 인증기능 제공

연계처리

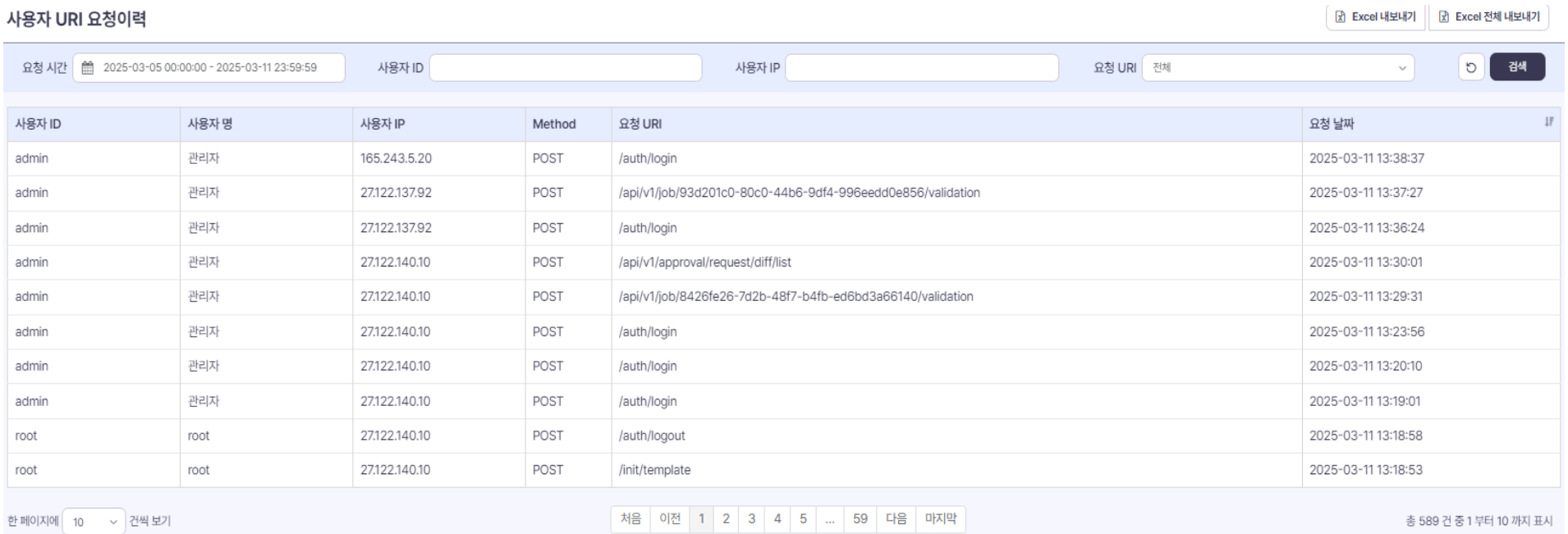
- 연계 인터페이스로 들어온 요청은 J-Job 매니저에서 처리되는 절차와 동일하게 수행

관리 - 패스워드 관리 및 사용자 요청 이력 관리 기능

J-Jobs는 패스워드 관리 규칙을 설정 기반으로 관리하고 있어 고객사 보안 규정에 따라 자유롭게 수정할 수 있으며, 사용자가 매니저 화면을 통해 사용한 API를 기록합니다.

키 값	설명
jjob.ac.password.valid.timestamp.admin jjob.ac.password.valid.timestamp.user	어드민 패스워드 유효시간 일반, 게스트 패스워드 유효시간
jjob.ac.password.temp.check	임시 비밀번호 사용 여부
jjob.ac.password.rule	비밀번호 규칙 (정규식)

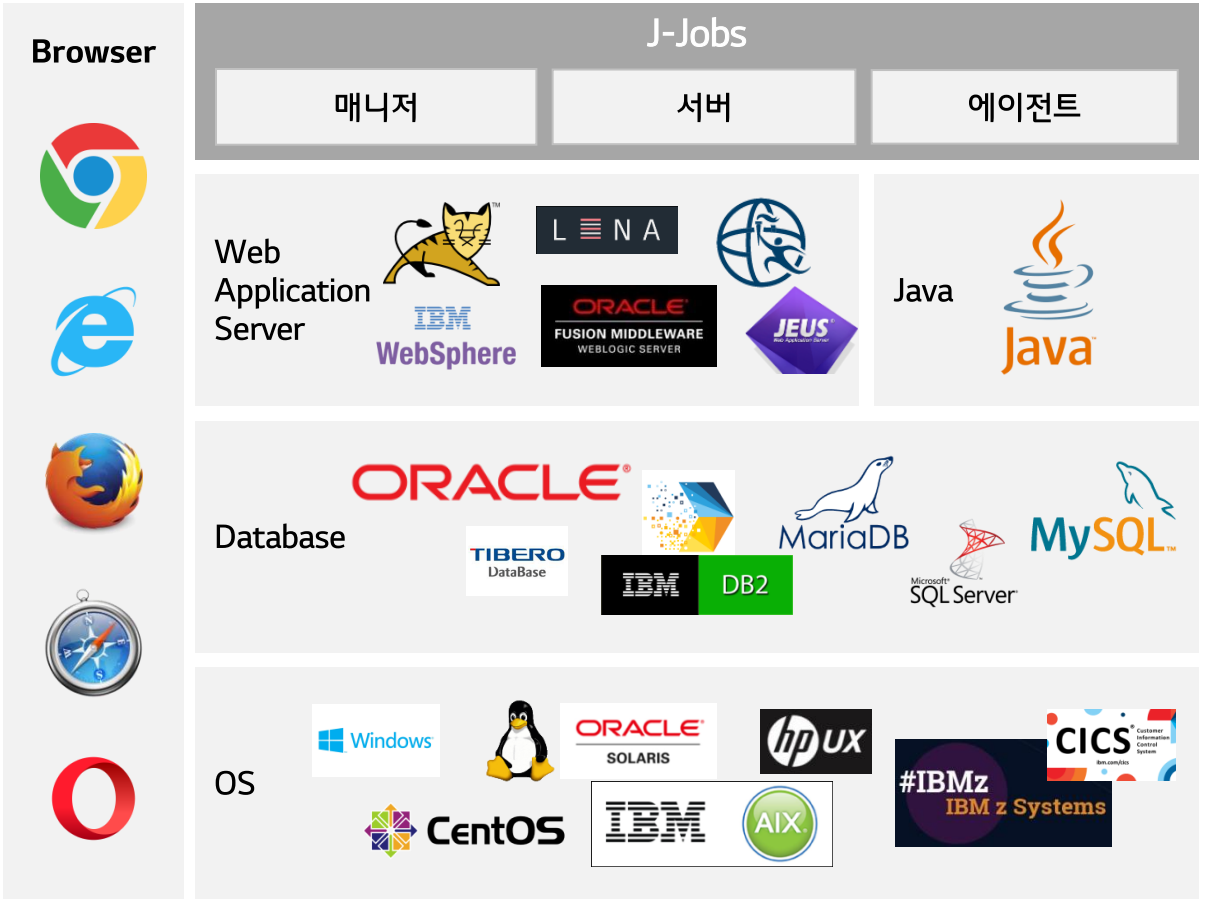
키 값	설명
jjob.ac.password.rule.repeat	동일한 문자값 반복 체크 여부
jjob.ac.password.rule.history.check	이전 패스워드 체크 여부
jjobs.ac.request.uri.history	사용자 요청 이력 관리 여부



지원 플랫폼

표준 배치 스펙(JSR-352) 준수하고 Java기반 엔진으로 구성되어 범용적인 플랫폼을 지원합니다. 또한 최소한의 WAS기능을 사용하여 WAS 종속성을 배제하였습니다.

지원 플랫폼



구성 및 특징

Any OS

- Windows, linux 설치 installer 제공
- z/OS CICS 호환성 테스트 완료

Any Database

- PostgreSQL 9 이상
- Oracle, Tibero 지원
- MariaDB , MySql
- SQL Server 2008 이상
- DB2 for z/OS, Windows, Linux

Any Was

- Weblogic, Tomcat, Websphere
- Websphere Liberty (CICS Liberty)
- LENA (LG CNS WAS)

Any Browser

- html5 지원하는 최신 브라우저

적용사례

공공 및 제조 고객사 시스템에 대한 자사 솔루션인 J-Jobs 배치 스케줄러 구축사례로 개발 생산성 및 운영관리 편리기능을 극대화하여 제공합니다.

공공 및 제조 구축 사례

사업명	고객사	구축시기
시스템 개선	J항공사	2024
디지털교과서 허브시스템	K교육기관	2024
차세대 행정시스템	N부처	2024
업무시스템 개선	J진흥원	2024
비대면 정책지원시스템	S진흥공단	2023
DX시스템 개선	D항공사	2023
업무시스템 개선	S에너지사	2023
스마트시티 구축사업	S지자체	2022
코어시스템	H물류사	2022
신뢰플랫폼	H공공기관	2021
차세대 사회보장정보시스템	B부처	2021
메인 배치솔루션 도입	S이커머스	2021
디지털 공제시스템 구축	K중소기업단체	2021
계열사 시스템 구축 다수	L전자 외	2021
계열사 시스템 구축 다수	L물류사 외	2021
고용산재보험료 퇴직정산시스템	G공단	2020
차세대시스템	J공제회	2019
통합커머스 시스템	G유통사	2019
통합 HR 시스템	A유통사	2018

적용사례

한국은행 구축사례를 포함하여 생명 및 보험사 IFRS 구축 등의 금융사 구축사례를 바탕으로 금융사 배치 스케줄러 기능요건을 충족시키고 있습니다.

주요 금융구축 사례

사업명	고객사	구축시기
시스템 고도화	D저축은행	2025
카드시스템 구축	K인터넷은행	2023
인사시스템 구축	D손해보험사	2022
업무시스템 개선	S캐피탈	2022
차세대 시스템 구축	H평가기관	2020
차세대 시스템 구축	H국책은행	2019
IFRS 시스템 구축	B보험기관 외 10개사	2019