

참여형 기후서비스 플랫폼 구축

**Developing a participatory climate service
platform**

**이현록, 한정민, 김상철, 신지현, 정주형, 정임국
기후사업본부 예측운영과**

2020.12.

APEC기후센터

평문 초록

APEC 기후센터에서는 아시아태평양 지역에서 기후정보가 좀 더 가치 있게 사용될 수 있도록 다양한 기후정보서비스를 개발하여 온라인에서 서비스하고 있다. 기존에 개발되어 서비스되고 있는 기후정보서비스는 다음과 같이 세 가지로 분류할 수 있다. 첫째, 기후예측정보 자체를 생산하는데 중점을 둔 서비스, 둘째, 기후예측정보를 다양한 형태로 가공할 수 있도록 지원하는 서비스, 셋째, 기후자료 자체를 디지털 형태로 전송 받아갈 수 있는 서비스로 분류할 수 있다. 이와 같은 서비스 구분을 바탕으로 사용자들이 요구하는 특정 목적 혹은 연구를 위해서 독립적으로 기능들을 집약하여 개발 및 서비스를 실시하고 있는데, 이런 목적지향 서비스들은 기후자료 수집, 기후자료 처리, 기후자료 표현 등을 기본적으로 포함하여 개발되는 것이 대부분이다. 목적지향 서비스들이 증가할수록 비슷한 기능들이 조금씩 다른 형태로 개발되고 서비스되어 가는 현시점에서 중복을 최소화하고 인적 물적 자원의 낭비를 줄일 수 있는 방법의 제시가 필요한 상황이었으며, 이를 위해 플랫폼 기술을 바탕으로 비슷한 기후정보서비스의 중복 기능을 통합하고 운영 효율을 높일 수 있도록 참여형 기후서비스 플랫폼으로의 전환을 제안하였다. 기후정보서비스들의 플랫폼 구축은 증가하는 기후정보 수용에 대응하여 기후예측자료 및 예측자료 활용 기술을 중심으로 통합하여 예산 절감효과와 함께 기후정보 수집부터 가공까지 사용자의 요구를 적합하게 충족시킬 수 있는 최적의 방안이다.

해당 과제는 기후자료 서비스를 중심으로 구축된 플랫폼에 APCC에서 현업으로 운영 중인 다중모델앙상블 계절예측 및 검증시스템을 플랫폼에 적합하도록 설계하고 개발하였으며, 이를 지원하기 위한 프로그램 인터페이스 개발과 웹 서비스 개발을 진행하였다. 또한 기후자료에 대한 활용성을 높이기 위한 기후자료처리 서비스를 확장 구축하기 위해 플랫폼 기반 기후자료 추출 기능과 표출 기능을 개발하여 서비스를 실시하였다. 또한 기후예측 연구 분야의 관학연합협력체계를 구축하기 위한 예측시스템 프로그램 소스코드 공유, 모델 자료 공유를 비롯하여 상호 정보교환을 위한 연구개발허브 시스템을 공동 구축하고 서비스 운영을 실시하였다.

본 과제를 통해 최신의 클라우드 플랫폼 기술을 기반으로 계절예측 및 검증, 기후자료 제공 및 처리서비스를 실시할 수 있는 APCC 기후정보도구를 새롭게 구성하여 서비스를 실시하였으며, 이는 기존의 개별적인 기후정보서비스를 점진적으로 통합해 하나의 플랫폼으로 발전해 나갈 수 있는 기반 기술을 확보한 것에 그 의의가 매우 깊다. 또한 기후예측시스템 연구개발허브를 구성하는 기후예측시스템 파일럿 현업화 시스템 및 스토리지를 도입하여 국립기상과학원과 함께 공동 구축하고 운영을 실시함으로써 기후예측시스템 연구 분야의 관학연합협력체계 구축을 위한 기초적인 시스템 및 기반기술을 확보하였다. 향후 해당 분야 연구개발 성과확산을 비롯하여 현업화 기간 단축 등에 크게 기여할 수 있을 것으로 판단된다.

Executive Summary

The APEC Climate Center provides various services for the production, processing, and provision of climate prediction data through individual climate information service which is tailored to the specific purpose, in order to effectively utilize the technologies and outcomes accumulated through the R&D project for climate services. However, proper integration and system of the corresponding information services are required for the development of climate services to reflect the need of various users. In other words, a climate service platform based on cloud technology, which is composed of a series of services including Multi-Model Ensemble(MME) prediction, presentation, and processing of the climate model data, should be established systematically.

The climate service system from developed countries such as the US and Europe has also provided climate information produced by observation, monitoring, research, modeling, and forecasting through the Climate Information Service System which is oriented by the provider. ECMWF C3S(Copernicus Climate Change Service) has recently been establishing the for providing a platform that can encourage user participation. Therefore, APCC climate information service system also should be turning into a system that can lead to new service contents with the participation of users, rather than supplying climate data unilaterally. This can be realized through the development and utilization of a participatory climate service platform that induces various climate information providers, modelers, and other climate data users. In addition, the establishment of an R&D Hub for supporting the climate prediction system development by utilizing the relevant cloud technology will solidify the cooperation system of the university, research institute, and government organization. The Hub can be as being a catalyst to share information for the climate prediction system R&D through model source code sharing with version control, exchange of opinions via Wiki page, and issue management by an issue tracking system.

The final goal achieved through this project is to build a platform-based climate service and primitive service for the gradual integration of APCC's climate information services. Through the project for this year, the online user-customized seasonal prediction and verification system, which is centered on the APCC multi-model ensemble deterministic/probabilistic seasonal prediction and verification, was started to serve web service through the platform. Also, climate data processing service that can be clipping regional area and specific variables from climate data has been developed and integrated into the platform. In addition, a research and development hub using the climate prediction system pilot operation server was jointly established with the NIMS(National Institute of Meteorological Science), and technical support such as network configuration, computer room rearrangement, etc for the installation was provided.

For the developing seasonal prediction and verification, the first detailed goal, we have

analyzed the production system of APCC DMME (Deterministic Multi-Model Ensemble) / PMME (Probabilistic Multi-Model Ensemble) in the operational system and existing CLIK service. The basic architecture for the platform modularization is summarized as follows: 1) seasonal prediction production component, 2) verification data production component, 3) user GUI for the data service, 4) user request input component, 5) data download component. For managing climate data, mongoDB and MySQL were selectively used according to the data properties, and RabbitMQ was used for message queue/job queue purposes to establish unity of processing. After modularization of the newly constructed climate prediction and verification system, experimental data were actually produced and compared with actual operation or CLIK production data to confirm that the system was consistent. The method of handling various task requests, which are divided into user request input, processing, status inquiry, data production, image request, packaging API, etc., finished development using Open API (Application Programming Interface). But also authenticated users have also completed the development of Java and Python libraries that can use the platform's corresponding API to directly transfer the work they want to the platform and receive the result. The web service of the seasonal prediction and verification service platform was integrated into the existing climate data service platform. In particular, the verification methods in the existing CLIK system are SR (Success Rate) and HSS (Heidke Skill Score). But additional two verification methods, which is ACC (Anomaly Correlation Coefficient) and ROC Curve (Relative Operating Characteristics Curve), were additionally developed and applied to provide various information to users. In order to achieve the second detailed goal, the climate data processing service on the platform, we firstly built a program for the data integrity check and synchronization, then we developed clipping and visualization modules that extract and visualize the variables with regions selected by the user. The climate data format conversion function was also implemented so that the result values of the corresponding climate data processing service can be downloaded in various file formats such as NetCDF, PNG, xls. Lastly, the joint construction of the climate prediction system R&D hub is a plan to resolve the delay in the process of operational version and the late response to the GC (Global Couple Model) development version. Through the R&D hub operation, it is possible to manage its own technology and accumulate competence, and it is a way to shorten the period of operationalization process by sharing knowledge among development groups and accelerating research and development using the wiki and cooperative development tools. The R&D hub starts with technical support for the transfer and installation of the climate prediction system pilot operational server and storage from the Korea Meteorological Administration's Super Computer Center. And the standard and sensitivity experiment data will share through the establishment of the issue management and version control system called TRAC and Subversion, respectively. It can be of great significance that it has built a foundation for sharing experiment data, and it will gradually operate as a core part of the collaborative development system for the climate prediction system while operating/managing the formal service following the pilot service.

국문 요약

기후예측정보 활용 기술개발을 통해 축적된 기술과 산출물을 기후서비스에 효과적으로 활용하기 위해서 APEC기후센터에서는 목적에 맞는 개별 기후정보서비스를 통해 기후예측자료의 생산, 가공, 제공 서비스를 실시하고 있다. 하지만, 다양한 사용자의 요구를 반영하기 위한 기후서비스를 개발하기 위해서는 해당 정보서비스들의 통합적인 활용이 필요하며 이를 반영한 체계가 필요하다. 즉, 다중모델 기반의 기후예측자료 제공, 해당 자료의 표출 및 가공을 포함하는 일련의 서비스가 연계되어 구성된 클라우드 기술 기반의 기후서비스 플랫폼이 체계적으로 구축되어야 한다.

미국, 유럽 등 선진국의 기후서비스 제공 체계도 관측·감시 및 연구·모델링·예측 등을 통해 생산된 기후예측정보를 기후정보서비스시스템(Climate Information Service System)을 통해 공급자 중심으로 제공하여 왔으며, 최근에 들어서야 ECMWF C3S(Copernicus Climate Change Service)를 필두로 사용자의 참여를 촉진할 수 있는 서비스 제공 기반을 마련하고 있는 실정이다. 따라서 APCC의 기후서비스 체계도 일방적으로 고정된 자료를 공급하는 형태가 아닌 사용자들의 참여로 새로운 서비스 콘텐츠로 이어질 수 있는 체계로의 전환이 필요하며, 이는 다양한 기후예측정보 생산자, 모델 개발자 및 관련분야 기후예측자료 사용자들의 참여를 유도하는 참여형 기후서비스 플랫폼의 개발 및 활용을 통해 실현될 수 있다. 또한 해당 클라우드 기술을 활용하여 기후예측시스템 연구개발의 지원을 위한 허브 구축은 관학연의 협력 체계를 공고히 하고 실질적인 소스코드 공유, 의견교환, 쟁점관리 등을 통해 기후예측시스템 연구개발에 필요한 정보들을 공유하여 실제 연구개발의 촉진제 역할을 할 수 있도록 구성할 필요가 있다.

해당 과제를 통해서 달성한 최종적인 목표는 APCC의 기후정보서비스의 점진적인 통합을 위한 기후서비스 플랫폼 및 기반이 되는 서비스를 구축하는 것이다. 당해연도 과제를 통해서 APCC 다중모델앙상블 결정론적/확률론적 계절예측 및 검증 중심의 한 온라인 사용자 맞춤형 계절예측 및 검증 시스템을 웹 서비스를 통해 사용자에게 서비스를 시작하였으며, 기후자료의 지역추출 기능 및 자료표출 기능을 개발하고 서비스를 실시하였다. 또한 기후예측시스템 파일럿 현업화 서버를 활용한 연구개발허브를 기상과학원과 공동으로 구축하고 해당 구축을 위한 기술지원을 실시하였다.

첫 번째 세부 목표인 계절예측 및 검증서비스 플랫폼 구축을 위해 현업 및 기존 서비스인 CLIK에서 제공 중인 APCC DMME (Deterministic Multi Model Ensemble)/PMME(Probabilistic Multi Model Ensemble) 예측자료/검증자료 생산 시스템을 분석하여 플랫폼 모듈화를 위한 기본 구조를 다음과 같이 정리하였다. 1) 계절예측 생산 컴포넌트, 2) 검증자료 생산 컴포넌트, 3) 자료서비스 사용자 GUI, 4) 사용자 요청 입력 컴포넌트, 5) 자료 다운로드 컴포넌트. 해당 기본 구조에 기후자료 관리용 데이터베이스로는 mongoDB, MySQL을 자료 속성에 따라 선별적으로 사용하였으며, 메시지 큐/작업 큐 용도로 RabbitMQ를 사용하여 처리과정의 통일성을 구축하였다. 신규로 구축된 기후예측 및 검증 시스템 모듈화 후에는 자료를 실제로 생산하여 실제 현업

혹은 CLIK 생산 자료와 비교하여 시스템이 일치함을 확인하였다. 실제 여러 가지 작업 요청을 처리하는 방식은 Open API(Application Programming Interface)를 활용하여 사용자 요청 입력, 처리, 상태조회, 자료생산, 이미지요청, 패키징 API 등으로 분류하여 구축 완료하였으며, 해당 시스템뿐만 아니라 추후 인증된 사용자는 플랫폼의 해당 API를 활용하여 자신이 원하는 작업을 직접 플랫폼으로 전송하여 그 결과를 받아올 수 있는 Java, Python 라이브러리 또한 개발을 완료하였다. 계절예측 및 검증 서비스 플랫폼의 웹 서비스는 기존 기후자료서비스 플랫폼에 통합하여 실시하였으며, 특히 검증기법은 사용자에게 다양한 정보를 제공하기 위해 기존 CLIK에서 제공하는 SR(Success Rate), HSS(Heidke Skill Score) 두 종류 이외에도 ACC(Anomaly Correlation Coefficient), ROC Curve (Relative Operating Characteristics Curve) 추가 개발하여 적용하였다. 두 번째 세부 목표인 기후자료 처리서비스를 플랫폼에 구축하기 위해서 먼저 자료 무결성 검증 및 자료 동기화 프로그램을 개발하고, 사용자가 선택한 변수 및 지역을 추출하여 지도기반 시각화를 제공할 수 있는 지역자료추출 서비스와 계절예측 자료 및 관측자료를 이용하여 사용자가 원하는 기간에 대한 합성 결과를 제공해주는 기후자료합성 서비스를 구축하였다. 해당 기후자료처리서비스의 결과 값들은 NetCDF, PNG, xsl 등 다양한 파일 포맷으로 다운로드 받을 수 있도록 기후자료 포맷 변환 기능 또한 구현하였다. 마지막으로 기후예측시스템 연구개발허브 공동 구축은 현재 기후예측시스템 연구개발의 문제점으로 지적되어온 GC(Global Couple model) 개발 버전에 대한 늦은 대응 및 현업화 과정 지연을 해결하기 위한 방안으로, 공동형상관리 체계 운영을 통해 자체 기술관리 및 역량 축적을 할 수 있으며 위키(Wiki) 및 협력 개발도구를 이용하여 개발 그룹 간 지식 공유 및 연구개발 가속화하여 현업화 기간을 단축할 수 있는 방안이다. 해당 연구개발허브는 기후예측시스템 파일럿 현업화 서버와 스토리지를 기상청 슈퍼컴퓨터센터로부터의 이전설치에 대한 기술지원을 시작으로 이슈관리시스템인 TRAC과 형상관리시스템인 Subversion의 구축을 통해 표준실험 및 민감도 실험자료를 공유할 수 있는 기반을 구축했다는 것에 큰 의미를 둘 수 있으며, 시범서비스에 이은 정식서비스에 대한 운영/관리를 진행하면서 점차 기후예측시스템에 대한 협업개발 체계의 핵심 파트로 운영예정이다.

Table of Contents

평문 초록	2
Executive Summary	3
국문 요약	5
Table of Contents	7
표 차례	9
그림 차례	10
1. 서론	1
2. 참여형 기후서비스 플랫폼 구축 개요	1
가. 플랫폼 서비스 구성 및 관리체계	1
(1) 클라우드 컴퓨팅	2
(2) 개방형 클라우드 플랫폼 파스-타	2
(3) PaaS-TA를 적용한 APCC 기후서비스 플랫폼	3
(4) 기후서비스 플랫폼 요구사항 및 Use Case 분석	7
(5) 기후서비스 플랫폼 내 메시지 큐 구축	11
(6) 기후서비스 플랫폼 내 메시지 큐 활용	13
(7) 기후서비스 플랫폼 내 DBMS(Database Management System) 구축	18
(8) PaaS-TA 내 DBMS 활용을 위한 매뉴얼	21
(9) 기후서비스 플랫폼 내 형상 및 배포관리	26
(가) Subversion	27
(나) Git	28
(10) 기후서비스 플랫폼 내 형상 및 배포 관리의 활용	28
3. APCC 계절예측 및 기후자료 처리서비스 플랫폼 구축	38
가. 플랫폼 기반 온라인 다중모델 앙상블 계절예측/검증시스템 개발	38
(1) 시스템 개발 필요성	38
(2) 시스템 설계	38
(가) 요구사항	38
(나) 요구사항 분석	39
(3) 계절예측 및 검증 시스템 구성 및 운영 환경	42
(4) 현업 시스템 모듈화	43
(5) 데이터베이스 설계 및 구축	56
(6) Open API 및 자료 생성 모듈 개발	59
(7) 계절예측 및 검증 웹 서비스 개발	69
(8) 계절예측 및 검증 서비스 실행	71
나. 플랫폼 기반 APCC 기후자료처리 시스템 개발	74
(1) 개요 및 필요성	74
(2) 플랫폼기반 기후자료서비스 개발	74
(가) 계절예측자료 추출 서비스 웹 인터페이스 개발	74
(나) 기후자료 및 관측자료 합성 서비스 웹 인터페이스 개발	75
(다) 계절예측 자료 지역 및 변수 추출 서비스 개발	76
(라) 서비스를 위한 계절예측 자료 관리 프로그램 개발	80
(마) 개별모델의 예측자료에 대한 편차 계산 프로그램 개발	81
(바) 가공자료의 지도기반 시각화 서비스 개발	82

(사) 계절예측 자료 합성서비스 개발	84
(아) 관측 자료 합성서비스 개발	85
(자) 가공자료의 파일포맷 변환 제공 서비스 개발	87
4. 기후예측시스템 연구개발허브 구축 및 기술지원	88
가. 연구개발허브 공동구축 및 운영	88
(1) 개요 및 필요성	88
(2) 이슈관리시스템 (TRAC)	88
(가) Apache 설치	88
(나) 형상관리 설치 및 환경설정	89
(다) Apache와 Subversion 연동	93
(라) TRAC 설치	95
(마) TRAC 자료 이관	96
(바) TRAC과 Apache 연동	97
(사) Plug-in 설치	102
(아) 연구개발허브 시범 운영실시	108
나. 연구개발허브 구축을 위한 기술지원	109
(1) 개요 및 필요성	109
(2) APCC 전산실 장비 배치 및 기술지원	110
5. 결론 및 향후 과제	111
사사	112
References	113
부록 A. CLIK (Climate Information Toolkit) 사용자 매뉴얼	1

포 차례

Table	Page
1. List of PaaS-TA add-on services installed on climate service platforms	5
2. PaaS-TA built-in app templates, app development environments	6
3. PaaS-TA User Portal Requirements	7
4. PaaS-TA Administrator Portal Requirements	8
5. List of Exchange Types supported by RabbitMQ, https://www.rabbitmq.com/	17
6. cf service-brokers command queries the list of service brokers	22
7. cf create-service brokers and query with the create-service-brokers command	22
8. cf enable-service-access Mongo-DB command after cf service-access	23
9. Requirements of MME prediction and verification.	38
10. Add environment variables for web interface	45
11. Verification skills for hindcast	48
12. Database table: clic_provider (list of model providers)	56
13. Database table: clic_model (list of models)	56
14. Database table: clic_variable (list of variables)	56
15. Database table: clic_verification_method (list of verification skills)	57
16. Database table: clic_data (list of model combinations)	57
17. Database table: clic_job (list of jobs to produce prediction and verification data)	58
18. Three areas of the system	59
19. List of API to support other services	67
20. List of API to support the others	67
21. Apache Installation	89
22. Subversion Installation and Configuration	89
23. Apache and SVN Configuration	93
24. Trac Installation	95
25. Trac glosea project migration	96
26. Trac and Apache Configuration	97
27. Python Configuration	98
28. Trac and Apache Configuration	102
29. Essential Plug-in list for Trac	102
30. Useful Plug-in list for Trac	102
31. Auxiliary Plug-in list for Trac	103
32. Install account manager plug-in	103
33. Install subtickets plug-in	103
34. Install masterticket plug-in	104
35. Install Timing and estimation plug-in	104
36. Install Gantt and calendar plug-in	105
37. Install jsGantt chart plug-in	108
38. Install code highlighting plug-in	108
39. Configuration of Pilot Operational Server	108
40. Configuration of Repository for R&D pilot system	109

그림 차례

Figure	page
1. PaaS-TA Configuration Details, http://pass-ta.kr	3
2. Basic Installation Structure of PaaS-TA Using BOSH2, https://github.com/PaaS-TA/	4
3. APCC Climate Service Platform IaaS with BOSH	5
4. PaaS-TA Basic App Development Environment List	6
5. APCC App Template Configuration Screen	7
6. PaaS-TA-RabbitMQ Service Architecture, https://github.com/PaaS-TA/	12
7. RabbitMQ service pack installed through BOSH	12
8. The screen that accessed RabbitMQ in PaaS-TA through the Web-UI	13
9. Routing model for RabbitMQ, http://tmondev.blog.me/	13
10. RabbitMQ : Admin - Users Menu	14
11. RabbitMQ : Admin - Virtual Host Menu	15
12. RabbitMQ : Queues - Add a new queue Menu	16
13. RabbitMQ : Exchanges Menu	17
14. RabbitMQ : Connections Menu	18
15. DBMS - MySQL System Configuration Diagram within Climate Service Platform PaaS-TA, http://paas-ta.kr	19
16. List of installed MongoDB VMs within the climate service platform	19
17. Diagram of the DBMS - MongoDB System in the Climate Service Platform PaaS-TA, http://paas-ta.kr	20
18. MongoDB Replica Set, https://www.mongodb.com/	20
19. DBMS - Redis System Configuration Diagram within Climate Service Platform PaaS-TA, http://paas-ta.kr	21
20. The screen of mongodb-test service created through the CF command.	24
21. The screen of Hello-spring-mongodb Application Deployment	24
22. cf bind-service and application restart	25
23. MongoDB load data that can be found in deployed applications	25
24. MongoDB Instance Environment Variables Available on the PaaS-TA User Portal	26
25. Subversion Schema, http://svnbook.red-bean.com/	27
26. PaaS-TA Configuration Management Service Pack Structure Diagram,	28
27. List of SourceControl service pack packages installed on climate service platforms	29
28. Configuration Management Service screen on the PaaS-TA User Portal	29
29. Source Control Management Page Initial Screen (List of Source Control Management Projects)	30
30. Viewing the details of the repertoire within the Configuration Management page	30
31. Source Control Management - New Create Menu	31
32. Source Control Management - New Creation Detail Settings Menu	31
33. Source Control Management - List of newly created repositories	31
34. Source Control Management - Copy New Created Storage Addresses	32
35. Source Control Management - SVN Checkout Using TortoiseSVN (1)	32
36. Source Control Management - SVN Checkout Using TortoiseSVN (2)	33

37. Source Control Management - SVN Checkout Using TortoiseSVN (3)	33
38. Source Control Management - SVN Add Using TortoiseSVN (1)	33
39. Source Control Management - SVN Add Using TortoiseSVN (2)	34
40. Source Control Management - SVN Add Using TortoiseSVN (3)	34
41. Source Control Management - SVN Commit Using TortoiseSVN (1)	35
42. Source Control Management - SVN Commit Using TortoiseSVN (2)	35
43. Source Control Management - SVN Commit Using TortoiseSVN (3)	35
44. Source Control Management - Updated Repository Content - Files	36
45. Source Control Management - Updated Repository Content - Source Code	36
46. Source Control Management - Updated Repository Content - Commit Log	37
47. Use case diagram	39
48. Activity diagram to request prediction and verification data	40
49. Activity diagram to produce forecast MME	41
50. Activity diagram to produce hindcast verification	42
51. System architecture of Prediction and Verification	43
52. Activity diagram to produce	44
53. NCL code analysis of SCM using MindMap	44
54. NCL code analysis of GAUS using MindMap	45
55. Output folder structure	47
56. Code differences of CLIKS ncl code vs AFS ncl code	47
57. CVS codes	48
58. Activity diagram of CVS	48
59. Python code to start running	49
60. Modifications of CVS codes (HSS)	49
61. Activity diagram to produce SR	51
62. Additional code for Success Rate	51
63. Maven dependency (pom.xml)	59
64. Configurations for RabbitMQ and MySQL	60
65. Class to get messages from RabbitMQ	61
66. Mybatis configuration class	61
67. Mapper xml for mybatis	62
68. Start application scanning mapper	62
69. Sample of mapper xml	62
70. Class to support transaction for mybatis	63
71. Starting to process job	64
72. Shell script to run code producing MME	64
73. Class to run shell scripts	66
74. Sample using Python: Prediction	68
75. Sample using Python: Verification	68
76. Web interface and API Source code	70
77. Prediction web interface	71
78. Verification web interface	72
79. Sample using python client (Prediction)	73
80. Sample using python client (Verification)	73

81 . User web interface for Seasonal forecast data extraction.	75
82 . User web interface for Seasonal forecast data and observation data composite	76
83. Provides condition input function for region and variable extraction.	77
84. This is a visualized result by extracting precipitation data from East Asia.	78
85. This is a visualized result by extracting Sea surface temperature data from Australia.	79
86. Development the data management for seasonal forecast and observation data	80
87. The results of calculating Anomaly and Mean for seasonal prediction data of 6-month multi-model ensemble.	81
88. Visualized results Sea surface temperature from Australia and Pacific islands.	82
89. Visualized results precipitation from global.	83
90. Visualized results Sea level pressure from East Asia.	83
91. It provides a cart function for the composite of seasonal forecast data	84
92. Composite visualization result of Seasonal forecast precipitation data	85
93. Mean composite visualization result of observation sea surface temperature data	86
94. Anomaly composite visualization result of observation sea surface temperature data	86
95. Target data and file formats input function.	87
96. Data structure in Excel file format by coordinates , level, lead-time.	87
97. R&D Hub (Climate Prediction System Pilot operational server)	109
98 Layout schematic for storage rack	110

1. 서론

기후예측정보 활용 기술개발을 통해 축적된 기술과 산출물을 기후서비스에 효과적으로 활용하기 위해서 APEC기후센터에서는 목적에 맞는 개별 기후정보서비스를 통해 기후예측자료의 생산, 가공, 제공 서비스를 실시하고 있다. 하지만, 다양한 사용자의 요구를 반영하기 위한 기후서비스를 개발하기 위해서는 해당 정보서비스들의 통합적인 활용이 필요하며 이를 반영한 체계가 필요하다. 즉, 다중모델 기반의 기후예측자료 제공, 해당 자료의 표출 및 가공을 포함하는 일련의 서비스가 연계되어 구성된 기후서비스 플랫폼이 체계적으로 구축되어야 한다.

해당 과제에서는 기후서비스 플랫폼을 구축하기 위해 플랫폼서비스(Platform as a Service, PaaS)로 선정된 파스-타(한국정보화진흥원) 체계를 APCC 기후서비스에 적합하도록 설정하고, 기존의 서비스들과 기후자료를 공유할 수 있는 체계를 구축하였다. 또한 기존 APCC의 통합인증 방법과 연동하여 인증할 수 있는 체계를 구축하고 해당 인증 기능 및 공개 응용프로그래밍 인터페이스(Open API, Open Application Programming Interface)에 대한 인증방안도 개발 완료하였다. 해당 기본 기능들을 활용하여 구축된 기후자료서비스 플랫폼은 일반 사용자에게는 웹 인터페이스를 사용하여 기후자료를 손쉽게 다운로드 받을 수 있도록 구성하고 기상/기후 전문가 집단에서는 API를 사용하여 자신의 프로그램에서 손쉽게 자료를 다운받을 수 있도록 하였다. 기후자료 가공 및 다운로드 요청 등 작업의 순차적인 처리를 위해 필요한 메시지 큐(Message Queue)는 RabbitMQ를 사용하였으며, 기후자료의 등재를 위한 데이터베이스 관리시스템(DBMS, DataBase Management System)은 NoSQL을 기본으로 하는 MongoDB를 활용할 수 있도록 구성하였다. 기 구축된 기후자료서비스 플랫폼에 더하여 당해연도 과제를 통해서 APCC 다중모델앙상블 결정론적/확률론적 계절예측 및 검증 중심의 한 온라인 사용자 맞춤형 계절예측 및 검증 시스템을 웹 서비스를 통해 사용자에게 서비스를 시작하였으며, 기후자료의 지역추출 기능 및 자료표출 기능을 개발하고 서비스를 실시하였다. 또한 기후예측시스템 파일럿 현업화 서버를 활용한 연구개발허브를 기상과학원과 공동으로 구축하고 해당 구축을 위한 기술지원을 실시하였다.

본 보고서의 구성은 다음과 같다. 먼저 2장에서 기후자료서비스 플랫폼 구축을 위한 관련 기술을 재차 살펴보고 해당 기술을 기후 과학자나 서비스 개발자들이 활용할 수 있도록 어떤 맞춤 설정을 수행하고, 기반이 되는 서비스는 무엇이 있는지 서술하였다. 3장에서는 향후 계절예측 및 검증 서비스의 플랫폼화와 기후처리를 위한 자료주출 및 합성기능을 추가한 플랫폼 서비스에 대하여 기술하였다. 4장에서는 기후예측시스템 연구개발 공동구축 및 기술지원에 대하여 상술하고, 마지막 4장에서 결론과 향후과제에 대하여 논의한다.

2. 참여형 기후서비스 플랫폼 구축 개요

가. 플랫폼 서비스 구성 및 관리체계

해당 항에서는 참여형 기후서비스 플랫폼 구축 과제에서 항상 먼저 설명이 되어야 후속

개발과정에 대한 이해를 높일 수 있는 플랫폼 서비스의 구성과 해당 구성을 위한 요구사항을 정리하고, 플랫폼 운용에 필요한 어플리케이션의 배포, 메시지 큐, 관리데이터베이스시스템, 형상관리와 활용에 대하여 정리하고자 한다.

(1) 클라우드 컴퓨팅

클라우드 컴퓨팅이 IT 분야의 새로운 패러다임으로 등장하면서 구글, 아마존, 애플 등 글로벌 클라우드 기업과 KT, LG CNS, 네이버 등 국내에서도 클라우드 서비스가 급증하고 있으며, 공공부문에서 역시도 정부통합전산센터, 서울특별시 등으로 클라우드 컴퓨팅 기술이 빠르게 확산되고 있다. 하지만 국내 클라우드 서비스는 대부분 인프라 서비스(IaaS)로 클라우드 응용 프로그램을 개발·운영하는 환경(PaaS)가 없어 클라우드 컴퓨팅의 응용 단계 생태계 구축이라는 커다란 숙제는 현재에도 진행형이다.

클라우드 플랫폼은 개발자가 어플리케이션을 개발, 서비스하기 위해 사용 가능한 서비스와 기능을 제공한다. 즉, 서비스를 개발할 수 있는 안정적인 환경과 그 환경을 이용하는 응용 프로그램을 개발할 수 있는 API 까지 제공하는 형태를 말한다. 특정 소프트웨어를 사용할 수 있게 클라우드를 활용하는 것이 아닌, 표준화된 플랫폼을 서비스 형태로 제공하는 서비스이다.

본 “기후자료서비스 플랫폼 구축”은 APEC 기후센터 내 기후 과학자나 기후 관련 소프트웨어 개발자들이 기후자료를 중심으로 응용 어플리케이션들을 개방형 플랫폼에 개발·배포·운영·관리할 수 있는 기반을 마련하고, 각각의 어플리케이션 서버들을 별도의 큰 비용 없이 이식 가능하게 함으로써 개별적인 서비스를 클라우드 플랫폼에서 운영하고자 하는 전략을 제시하고자 한다.

(2) 개방형 클라우드 플랫폼 파스-타

개방형 클라우드 플랫폼이란 클라우드 플랫폼을 오픈 소스 기반의 클라우드 플랫폼으로 제공하는 것을 뜻한다. 오픈 소스의 가장 큰 특징 중 하나인 그 유연성은 오픈 소스 기반의 클라우드 플랫폼은 여러 클라우드 인프라(IaaS)에서 운영이 가능하다.

대표적인 개방형 클라우드 플랫폼은 Pivotal의 CloudFoundry와 RedHat의 OpenShift가 있다. CloudFoundry(이하 CF)는 VMWare가 인수한 SpringSource의 Java 플랫폼인 SpringSource CloudFoundry를 기반으로 시작하였고, 시간이 지남에 따라 언어와 서비스, 프레임워크 등이 추가되어 발전되어 가고 있다. Pivotal이 CloudFoundry를 인수함에 따라 역시 Pivotal에서 진행중인 Spring Framework, RabbitMQ 등과 그 시너지 효과를 내고 있다고 할 수 있다. OpenShift는 대표적인 RedHat이 어플리케이션 제작, 테스트, 운용 및 관리 등에 필요한 다양한 플랫폼, 개발 언어, 프레임워크 및 라이브러리 환경에 포괄적이고 광범위한 클라우드 사업자를 지원하는 인프라를 제공하려는 전략으로 탄생된 대표적인 개방형 클라우드 플랫폼 중 하나이다.

APEC 기후센터에서 진행 중인 참여형 기후서비스 플랫폼(이하 플랫폼)은 한국정보화진흥

원(NIA)에서 진행 중인 “개방형 PaaS 플랫폼 고도화 및 개발자 지원환경 개발” 과제로 개발된 PaaS-TA를 기반으로 구성되어 있다.

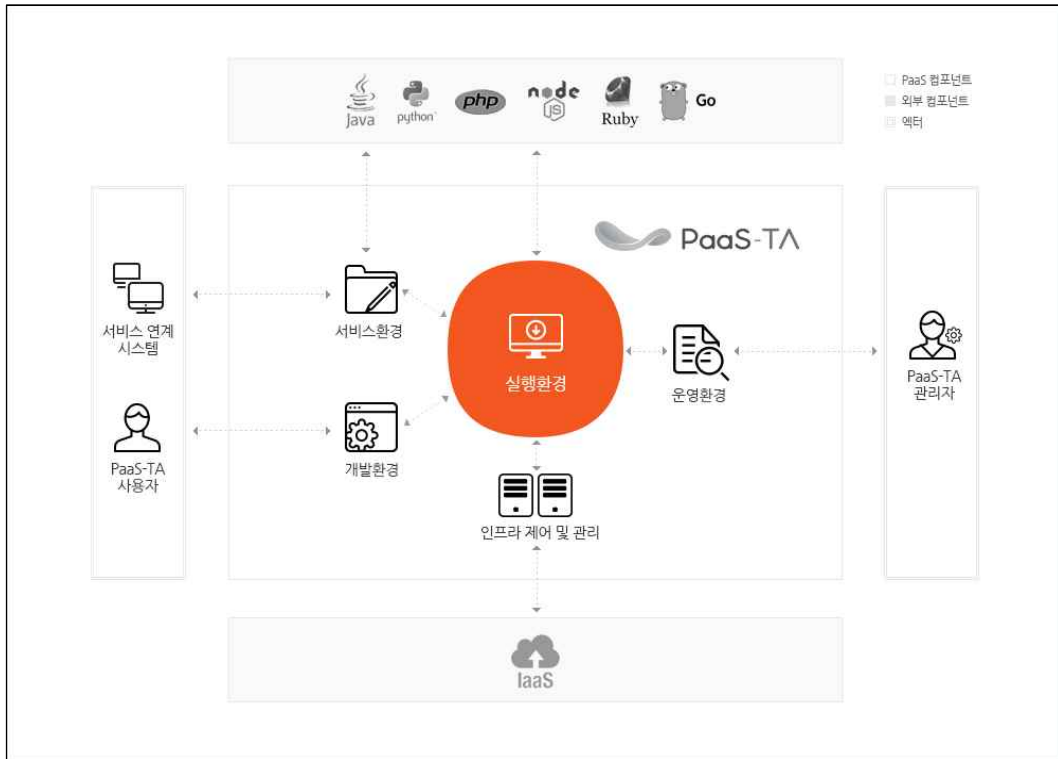


Figure 1. PaaS-TA Configuration Details, <http://pass-ta.kr>

PaaS-TA는 CF 기반으로 개발되었으며, 2017년 2월에 그 소스코드가 최초로 공개 되었다. PaaS-TA는 VMWare, OpenStack 등 7종 이상의 다양한 인프라(IaaS) 지원하며, 8종의 개발언어와 10종의 프레임워크를 스타터 카탈로그로 제공하여 풍부한 개발·운영환경을 제공하고 있다. 이와 더불어 IaaS/VM/컨테이너 등을 통합하여 모니터링할 수 있는 기능을 포함하고 있으며, 이를 통한 자동 확장 기능(Auto-Scaling)을 제공하고 있다. (출처 : 개방형 클라우드 플랫폼 파스타 발표자료, 한국정보화진흥원 윤희근 선임)

(3) PaaS-TA를 적용한 APCC 기후서비스 플랫폼

기후서비스 플랫폼은 PaaS-TA를 활용하여 구성되어 있으며, PaaS-TA는 BOSH¹⁾를 통해서 설치할 수 있다. BOSH는 초기에 Cloud Foundry PaaS를 위해 개발되었지만, 현재는 Jenkins, Hadoop 등 Yaml 파일 형식으로 소프트웨어를 쉽게 배포할 수 있으며, 수 백 가지의 VM을 설치할 수 있고, 각각의 VM에 대해 모니터링, 장애 복구 등 라이프 사이클을 관리할 수 있는 통합 프로젝트이다.

BOSH가 지원하는 IaaS는 VMware vSphere, Google Cloud Platform, AWS, OpenStack, MS Azure, VMware vCloud, RackHD, SoftLayer가 있다. PaaS-TA는 VMware vSphere, Google

1) 클라우드 환경에 서비스 시스템을 배포할 수 있는 BOSH는 릴리즈 엔지니어링, 개발, 소프트웨어 라이프사이클 관리를 통한 오픈소스 프로젝트

Cloud Platform, AWS, OpenStack, MS Azure 등의 IaaS를 지원한다. (출처 : 파스타 GitHub 페이지, <https://github.com/PaaS-TA>)

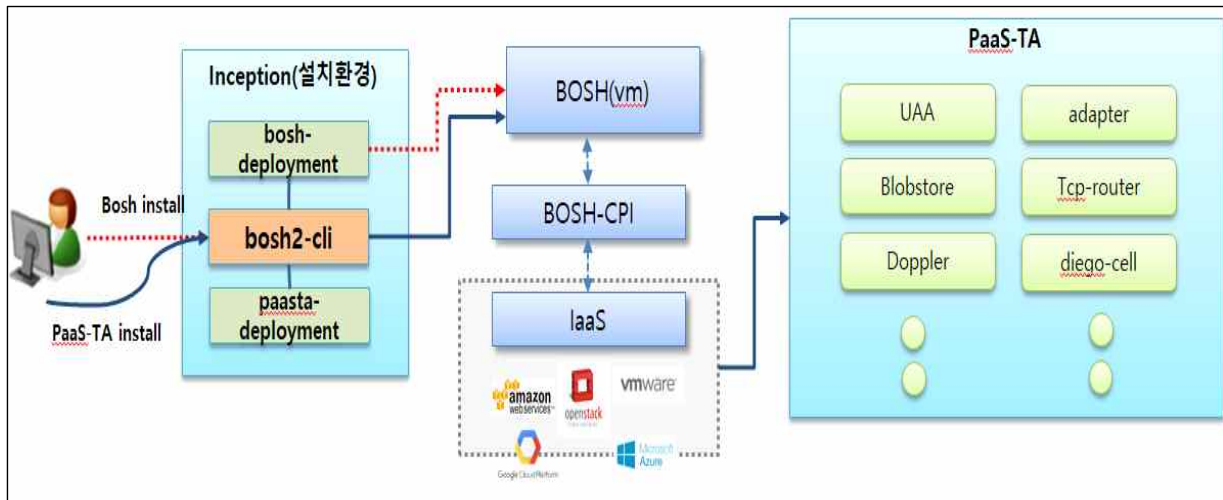


Figure 2. Basic Installation Structure of PaaS-TA Using BOSH2, <https://github.com/PaaS-TA/>

BOSH2는 위 Figure 2와 같이 BOSH2 CLI를 통하여 BOSH와 PaaS-TA를 생성한다. 그 이후, bosh-deployment를 이용하여 BOSH를 생성한 후, paasta-deployment로 PaaS-TA를 생성하는 순으로 설치가 진행된다. PaaS-TA를 설치하기 위해 IaaS 관련 network, storage, vm 관련 설정들을 정의한다. BOSH는 이러한 복잡한 각각의 설정을 Cloud-Config로 저장하여 이 설정에 따라 PaaS-TA VM(IaaS 레벨)을 생성하고 설치한다.

BOSH의 설치가 완료 되고 나면, BOSH를 통하여 PaaS-TA Core를 배포한다. paasta-deployment.yml 파일은 PaaS-TA를 배포하는 핵심 Manifest 파일이며, PaaS-TA VM에 대한 설치 정의를 하게 된다. APEC 기후센터에서 사용하고 있는 IaaS는 vSphere로 위 Figure 3과 같이 BOSH를 통하여 IaaS 내 BOSH 서버를 구성하였다.

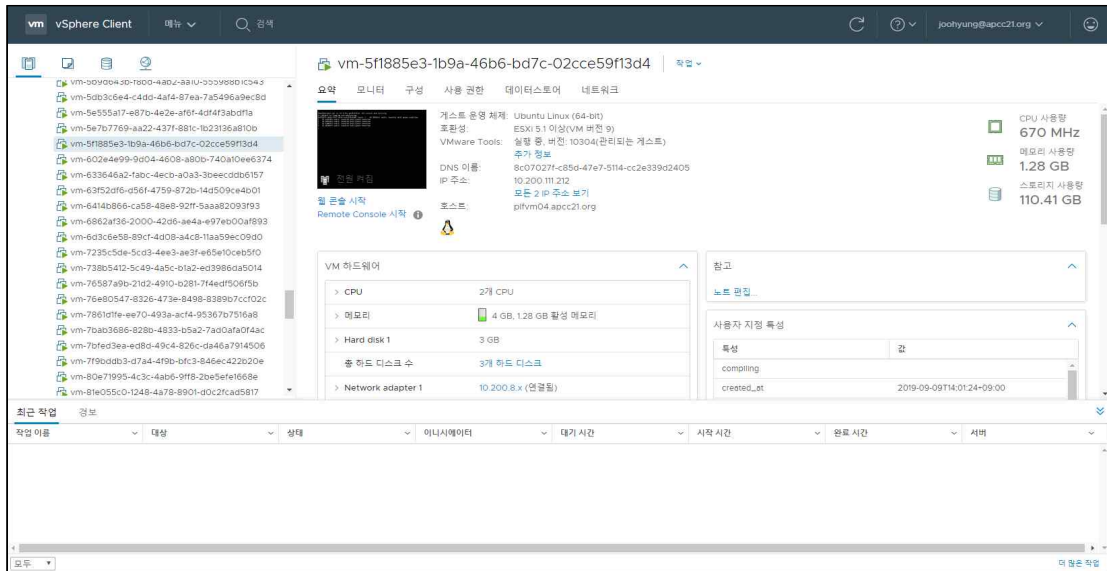


Figure 3. APCC Climate Service Platform IaaS with BOSH

PaaS-TA Core가 설치되면, 세부적인 설정을 위하여 CF CLI를 설치한다. CF는 CloudFoundry의 핵심 어플리케이션으로 각각의 PaaS 영역을 생성/삭제 및 관리한다. CF CLI를 통하여 이후 CF 명령어를 통하여 PaaS-TA 설치시 지정해주었던 CF system domain으로 접속하여 PaaS-TA의 세부 설정 및 각 서비스를 설치하였고, 그 서비스 목록은 아래 Table 1과 같다.

Table 1. List of PaaS-TA add-on services installed on climate service platforms

서비스 명	내용	참조 파일
PaaS-TA Portal	파스타 사용자/관리자 포탈	06_paasta-portal-bosh2.0.yml 07_deploy-portal-bosh2.0-vsphere.sh
paasta-container-service-2.0	파스타 Container 서비스	08_paasta-container-service-vsphere-vars.yml 09_deploy-vsphere.sh
paasta-delivery-pipeline-release-1.0	파스타 배포 파이프라인	10_paasta_delivery_pipeline_bosh2.0.yml 11_deploy-delivery_pipeline-bosh2.0.sh
paasta-glusterfs-2.0	GlusterFS 분산처리파일시스템	12_paasta_swift_object_bosh2.0.yml 13_deploy-swift-object-bosh2.0.sh
paasta-logging-service-2.0	로그 서비스	14_paasta_logging_service.yml 15_logging-service-deploy.sh
paasta-mongodb-shard-2.0	DBMS mongodb	16_paasta_mongodb_shard_bosh2.0.yml 17_deploy-mongodb-shard-bosh2.0.sh
paasta-monitoring	파스타 모니터링 서비스	18_paasta-monitoring.yml 19_monit-deploy.sh
paasta-mysql-2.0	DBMS MySQL	20_paasta_mysql_bosh2.0.yml 21_deploy-mysql-bosh2.0.sh
paasta-rabbitmq-2.0	Message Queue 서비스	22_paasta_rabbitmq_bosh2.0.yml 23_deploy-rabbitmq-bosh2.0.sh
paasta-redis-2.0	DBMS Redis	24_paasta_redis_bosh2.0.yml 25_deploy-redis-bosh2.0.sh
paasta-sourcecontrol-release-1.0	파스타 형상관리 서비스	26_paasta_sourcecontrol_bosh2.0.yml 27_deploy-sourcecontrol-bosh2.0.sh

PaaS-TA는 범용적으로 클라우드 개발 환경을 보급하고자 만들어진 목적이기에, nginx, PHP, Python, Go, Node.js, Java, Ruby 등 아래 Figure 4, Table 2와 같이 7종의 널리 쓰이는 프로그래밍 언어와 4종의 기본 어플리케이션 개발을 위한 앱 템플릿이 구성되어 있다.

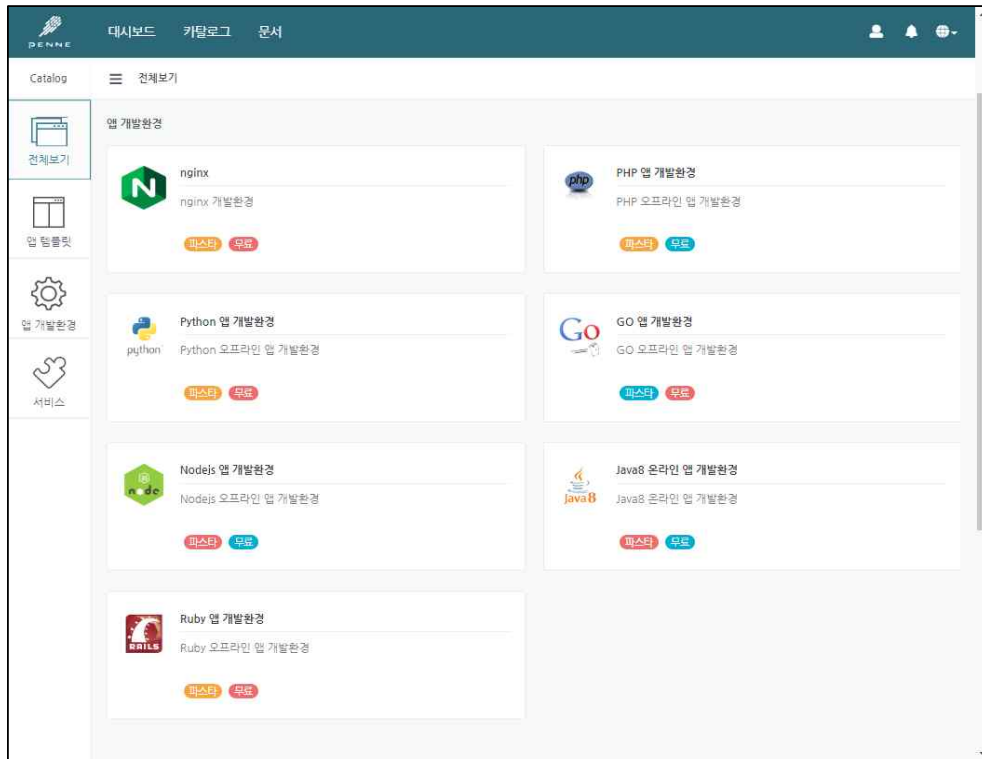


Figure 4. PaaS-TA Basic App Development Environment List

Table 2. PaaS-TA built-in app templates, app development environments

앱 템플릿	앱 개발환경
자바 기본 유형 템플릿 전자정부 프레임워크 웹 어플리케이션 템플릿 NodeJS + Mongo-DB 앱 템플릿 Java + Redis 앱 템플릿	nginx PHP Python GO Node.js Java Ruby

APEC 기후센터에서는 Python, Java, R, NCL 등의 기후데이터 가공을 위한 프로그래밍 언어의 사용 빈도가 높으며, 더불어 MySQL, MariaDB, MongoDB 등과도 같은 DBMS 역시도 많이 사용하고 있으므로, 이에 빠른 어플리케이션 개발 환경이 필요하였다.

아래 Figure 5와 같이 센터 내에서 운영하고자 하는 기후데이터 서비스에 적합한 어플리케이션 템플릿 구성을 통한 빠른 앱 개발 환경 구성하였고, PaaS-TA의 포털에서 클릭만으로 이러한 프로그래밍 언어와 DBMS의 조합을 사용할 수 있게 되었다. 결과적으로 아래 그림과 같이 Java-MySQL, Python-MongoDB 등의 조합 가능한 어플리케이션 템플릿 제공을 통한 빠른 개발환경을 제공할 수 있다.

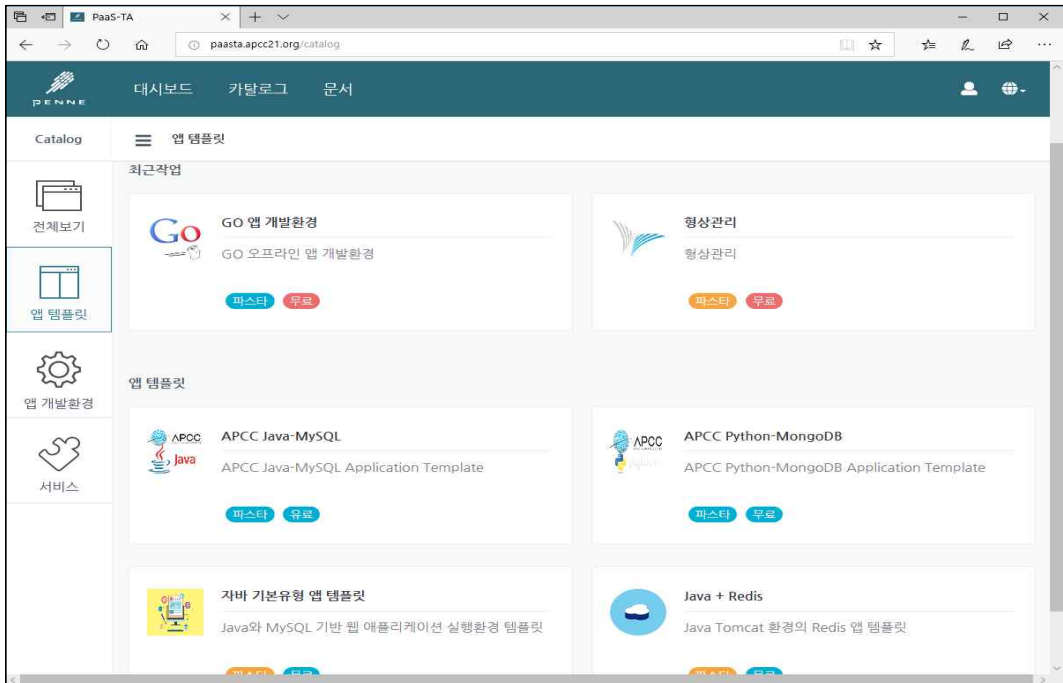


Figure 5. APCC App Template Configuration Screen

(4) 기후서비스 플랫폼 요구사항 및 Use Case 분석

본 절에서는 Open PaaS 플랫폼 PaaS-TA의 요구사항, 기술 구조 및 동작 매커니즘을 분석한다. 유즈 케이스 명세를 기준으로 포털이 제공하는 요구사항 및 유즈 케이스 요건은 다음 Table 3, 4와 같다.

Table 3. PaaS-TA User Portal Requirements

요구사항 ID	요구사항 내용
RQ_PF_PU_01	<p>시스템은 PaaS-TA의 연결 정보 및 인증 정보, 목표 스페이스를 등록하고 해당 정보를 통해 PaaS-TA 연결을 지원한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 어플리케이션 목록을 조회 할 수 있어야 한다. 어플리케이션 상세 정보를 조회 할 수 있어야 한다. 어플리케이션의 이벤트 정보를 확인 할 수 있어야 한다.

RQ_PF_PU_02	<p>시스템은 연결된 PaaS-TA의 목표 스페이스 내의 어플리케이션 목록을 조회하고, 특정 어플리케이션의 상세 정보 조회를 지원한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 어플리케이션 기본 모니터링 정보를 조회 할 수 있어야 한다. (Memory, CPU, Disk 사용량 등)
RQ_PF_PU_03	<p>시스템은 배포된 어플리케이션의 상태 제어 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 어플리케이션을 구동할 수 있어야 한다. 2. 어플리케이션을 삭제할 수 있어야 한다. 3. 어플리케이션을 이름 변경할 수 있어야 한다. 4. 어플리케이션을 재기동할 수 있어야 한다. 5. 어플리케이션을 Scale In/Out, Up/Down 할 수 있어야 한다.
RQ_PF_PU_04	<p>시스템은 배포된 어플리케이션의 라우트, 환경 변수, 메모리 사용량을 조회하거나, 수정을 지원한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 어플리케이션 환경 정보 목록을 조회할 수 있어야 한다. 2. 어플리케이션 환경 정보를 추가할 수 있어야 한다. 3. 어플리케이션 환경 정보를 삭제할 수 있어야 한다.
RQ_PF_PU_05	<p>시스템은 목표 스페이스에 PaaS-TA에서 사용할 수 있는 서비스팩을 조회하여 서비스팩 인스턴스 생성 및 어플리케이션과 서비스팩 인스턴스 바인딩을 지원한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 연결된 PaaS-TA의 서비스팩 인스턴스를 조회할 수 있어야 한다. 1. 바인드된 서비스 목록을 조회할 수 있어야 한다. 2. 어플리케이션에 서비스를 바인드할 수 있어야 한다. 3. 어플리케이션에 바인드된 서비스를 해제할 수 있어야 한다.
RQ_PF_PU_06	<p>시스템은 배포된 어플리케이션 모니터링 및 Auto-Scaling을 지원한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. Auto-Scale 조건을 조회 / 설정할 수 있어야 한다. 2. Auto-Scaling을 위한 App 모니터링 정보 수집 할 수 있어야 한다. 3. Auto-Scaling 설정 정보와 모니터링 정보 비교 분석을 해야 한다. 4. Auto-Scale 조건에 해당되면 Scale In/Out을 할 수 있어야 한다.

Table 4. PaaS-TA Administrator Portal Requirements

요구사항 ID	요구사항 내용
RQ_PF_PA_01	<p>시스템은 사용자 및 관리자 정보 조회, 삭제 및 비밀번호 재설정 등 관리 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 사용자 목록을 조회할 수 있어야 한다. 2. 사용자 상세 정보 내역을 조회할 수 있어야 한다. 3. 사용자를 삭제할 수 있어야 한다. 4. 사용자 비밀번호를 재설정할 수 있어야 한다. 5. 관리자 권한 부여할 수 있어야 한다.
RQ_PF_PA_02	<p>시스템은 클라이언트 정보 조회, 생성, 수정, 삭제 등 관리 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 클라이언트(uaa client) 목록을 조회할 수 있어야 한다. 2. 클라이언트 상세 정보 내역을 조회할 수 있어야 한다. 3. 클라이언트를 추가할 수 있어야 한다. 4. 클라이언트를 수정할 수 있어야 한다. 5. 클라이언트를 삭제할 수 있어야 한다.
RQ_PF_PA_03	<p>시스템은 조직(org) 및 스페이스(space) 정보 조회 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 조직 목록을 조회할 수 있어야 한다. 2. 조직 상세 내역을 조회할 수 있어야 한다. 3. 스페이스 목록을 조회할 수 있어야 한다. 4. 스페이스 상세 내역을 조회할 수 있어야 한다.
RQ_PF_PA_04	<p>시스템은 권한정보 조회, 사용자 권한그룹 생성, 수정, 삭제 등 관리 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 권한 그룹 목록을 조회할 수 있어야 한다. 2. 사용자 권한그룹을 생성할 수 있어야 한다. 3. 사용자 권한그룹을 삭제할 수 있어야 한다. 4. 권한 그룹에 사용자를 등록할 수 있어야 한다. 5. 권한 그룹에 사용자를 삭제할 수 있어야 한다.
RQ_PF_PA_05	<p>시스템은 서비스 카탈로그 정보 조회, 생성, 수정, 삭제 및 카탈로그에 서비스를 추가, 삭제 수정 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 서비스 카탈로그 목록을 조회할 수 있어야 한다. 2. 서비스 카탈로그 상세내역을 조회할 수 있어야 한다. 3. 서비스 카탈로그를 등록할 수 있어야 한다.

	<p>4. 서비스 카탈로그를 수정할 수 있어야 한다.</p> <p>5. 서비스 카탈로그를 삭제할 수 있어야 한다.</p> <p>6. 카탈로그에 서비스를 추가할 수 있어야 한다.</p> <p>7. 카탈로그에 서비스를 삭제할 수 있어야 한다.</p> <p>8. 카탈로그에 서비스를 수정할 수 있어야 한다.</p>
RQ_PF_PA_06	<p>시스템은 실행환경 카탈로그 정보 조회, 생성, 수정, 삭제 및 카탈로그에 빌드팩을 추가, 삭제 수정 기능을 제공한다.</p> <p><세부 요구사항></p> <p>1. 실행환경 카탈로그 목록을 조회할 수 있어야 한다.</p> <p>2. 실행환경 카탈로그 상세 내역을 조회할 수 있어야 한다.</p> <p>3. 실행환경 카탈로그를 등록할 수 있어야 한다.</p> <p>4. 실행환경 카탈로그를 수정할 수 있어야 한다.</p> <p>5. 실행환경 카탈로그를 삭제할 수 있어야 한다.</p> <p>6. 카탈로그에 빌드팩을 추가할 수 있어야 한다.</p> <p>7. 카탈로그에 빌드팩을 삭제할 수 있어야 한다.</p> <p>8. 카탈로그에 빌드팩을 수정할 수 있어야 한다.</p>
RQ_PF_PA_07	<p>시스템은 카탈로그 정보 조회, 생성, 수정, 삭제 및 서비스 및 빌드팩을 추가, 삭제 수정 기능을 제공한다.</p> <p><세부 요구사항></p> <p>1. 카탈로그 목록을 조회할 수 있어야 한다.</p> <p>2. 카탈로그 상세 내역을 조회할 수 있어야 한다.</p> <p>3. 카탈로그를 등록할 수 있어야 한다.</p> <p>4. 카탈로그를 수정할 수 있어야 한다.</p> <p>5. 카탈로그를 삭제할 수 있어야 한다.</p> <p>6. 카탈로그에 템플릿(서비스+실행환경)을 추가할 수 있어야 한다.</p> <p>7. 카탈로그에 템플릿(서비스+실행환경)을 삭제할 수 있어야 한다.</p> <p>8. 카탈로그에 템플릿(서비스+실행환경)을 수정할 수 있어야 한다.</p>
RQ_PF_PA_08	<p>시스템은 설정 정보 관리 기능을 제공한다.</p> <p><세부 요구사항></p> <p>1. 설정 정보 상세 내역을 조회할 수 있어야 한다.</p> <p>2. 설정 정보를 수정할 수 있어야 한다.</p>
RQ_PF_PA_09	<p>시스템은 코드 정보 조회, 생성, 수정, 삭제 등 관리 기능을 제공한다.</p> <p><세부 요구사항></p> <p>1. 코드 목록을 조회할 수 있어야 한다.</p> <p>2. 코드 상세 내역을 조회할 수 있어야 한다.</p> <p>3. 신규 코드를 등록할 수 있어야 한다.</p> <p>4. 코드 정보를 수정할 수 있어야 한다.</p> <p>5. 코드를 삭제할 수 있어야 한다.</p>
RQ_PF_PA_10	<p>시스템은 App 코드 정보 조회, 생성, 수정, 삭제 등 관리 기능을 제공</p>

	<p>한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 어플리케이션 생성시 제공하는 프로젝트 코드 목록을 조회할 수 있어야 한다. 2. 어플리케이션 코드 상세 내역을 조회할 수 있어야 한다. 3. 어플리케이션 코드 템플릿을 등록할 수 있어야 한다. 4. 어플리케이션 코드 템플릿을 수정할 수 있어야 한다. 5. 어플리케이션 코드 템플릿을 삭제할 수 있어야 한다.
RQ_PF_PA_11	<p>시스템은 메뉴 정보 조회, 생성, 수정, 삭제 및 메뉴에 권한 등록 등 관리 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 메뉴 목록을 조회 할 수 있어야 한다. 2. 메뉴 상세 내역을 조회 할 수 있어야 한다. 3. 메뉴 등록, 수정 및 삭제를 할 수 있어야 한다.
RQ_PF_PA_12	<p>시스템은 관리자 대시보드, VM별 모니터링 및 IaaS 모니터링 기능을 제공한다.</p> <p><세부 요구사항></p> <ol style="list-style-type: none"> 1. 관리자 계정으로 로그인시 최초로 보여주는 관리자 대시 보드를 제공해야 한다. 2. 설치된 VM별 모니터링 정보를 제공한다. 3. IaaS 모니터링 정보를 제공한다.

(5) 기후서비스 플랫폼 내 메시지 큐 구축

서버를 운영함에 있어 성능에 대한 고려는 반드시 필요한 부분이고, 이를 효율적으로 설정하는 것은 시스템 엔지니어의 몫이다. 서버는 각각의 클라이언트가 요청하는 데이터 처리를 순서대로 진행하게 되면 너무 많은 작업 요청으로 인해 요청이 대기하는 일이 발생하기도 한다. 일반적인 웹 서버의 경우 사용자는 3초 이상을 힘들어 하며, 시스템이 이상이 있는 것은 아닌가 의심하기도 한다. 이러한 작업 요청이 축적되면 서버는 과도한 부하 발생으로 인해 성능이 저하되기도 하며 부하가 특정선을 넘기면 서버가 비정상 종료되는 상황까지 발생하기도 한다.

이런 비정상적인 서버의 상황을 미연에 방지하기 위해서 서버에서 로드 밸런싱, DB 튜닝 등을 선택하여 그러한 부하를 분산하기도 한다. 하지만 접속에 대한 한계 그 자체를 해결할 수는 없다. 그래서 고안해낸 방식이 비동기 메시지 처리 방식인 메시지 큐(Message Queue)이다. 서버가 클라이언트와 동기방식으로 많은 데이터 통신을 하게 되면 병목현상이 생기게 되고, 서버의 성능이 저하되는데, 이런 현상을 막고자하여 또 하나의 미들웨어에 메시지를 위임하여 순차적으로 처리를 하는 것이다.

메시지 큐는 비동기 메시지를 사용하는 프로그램들 사이에서의 데이터 송수신(MOM²⁾을

구현한 시스템으로, 클라이언트의 요청을 즉각적으로 모두 처리하는 것이 아니라 순차적으로 처리할 수 있는 대기표를 각각의 요청들에게 주어, 공정 작업을 연기할 수 있는 유연성을 제공한다. 즉, 대용량 데이터를 처리하기 위한 배치 작업이나, 채팅 서비스, 비동기 데이터를 처리할 때 사용하기 유용하다.

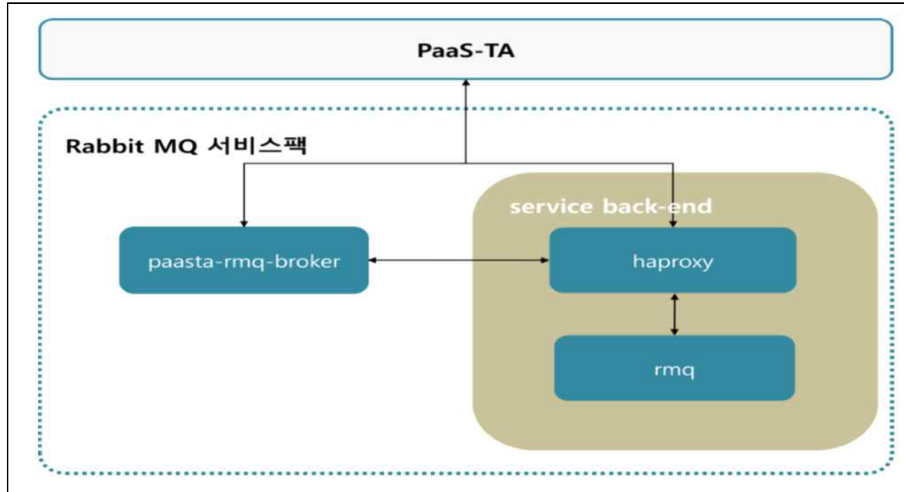


Figure 6. PaaS-TA-RabbitMQ Service Architecture,
<https://github.com/PaaS-TA/>

PaaS-TA에서도 클라이언트의 요청을 비동기 방식으로 처리하기 위하여 메시지 큐 시스템을 사용할 수 있으며, 이를 활용하기 위하여 RabbitMQ의 서비스팩을 제공한다. PaaS-TA의 RabbitMQ 서비스팩 구조는 위 Figure 6과 같이 RabbitMQ Server 1대, RabbitMQ Service Broker 1대, haproxy 1대로 구성되어 있으며, Figure 7처럼 파스타의 다른 서비스 팩들과 같이 BOSH를 통해 배포하였다.

```

joohyung@ubuntu:~/workspace/paasta-4.0/stemcells$ bosh -d paasta-rabbitmq-service vms
Using environment '10.200.8.212' as client 'admin'

Task 779819. Done

Deployment 'paasta-rabbitmq-service'

Instance                                Process State AZ  IPs                VM CID                                     VM Type Active
haproxy/686b2a43-2b92-4a86-acc5-9a9cf512b013 running      z1  10.200.8.87       vm-56256afb-f1d1-403a-9876-da9a32edbc6b  minimal true
paasta-rmq-broker/cda8da26-ac0c-406c-82df-8ae92f52653b running      z1  10.200.8.86       vm-116db943-c900-460b-a257-5f329c0c133e  minimal true
rmq/2ff0188a-41e3-45b2-b74a-8ff3406deb4f  running      z1  10.200.8.88       vm-76e80547-8326-473e-8498-8389b7ccf02c  minimal true

3 vms

Succeeded
joohyung@ubuntu:~/workspace/paasta-4.0/stemcells$

```

Figure 7. RabbitMQ service pack installed through BOSH

2) 메시지 지향 미들웨어(Message Oriented Middleware : MOM) : 분산 시스템 간 메시지를 주고 받는 기능을 지원하는 소프트웨어나 하드웨어 인프라

더불어 RabbitMQ에서 제공하는 Management를 위한 Web-UI를 설치하여, 센터 내 사용자들이 CLI 명령어를 통하여 사용하기보다 시각적으로 편리하게 활용할 수 있도록 사용자의 사용성을 제고하였다.

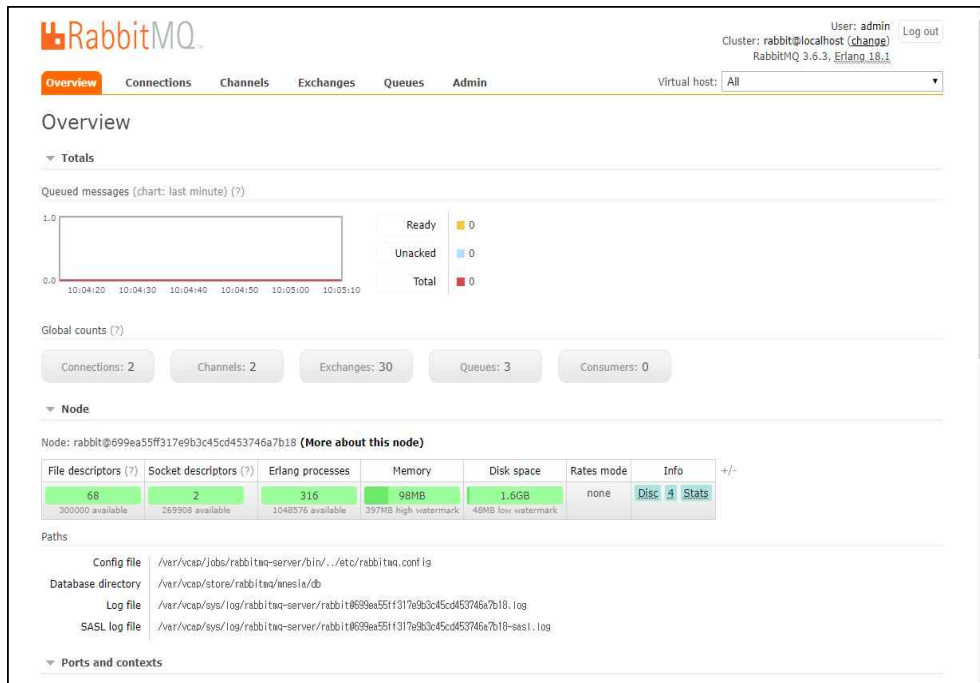


Figure 8. The screen that accessed RabbitMQ in PaaS-TA through the Web-UI

(6) 기후서비스 플랫폼 내 메시지 큐 활용

RabbitMQ를 포함한 AMQP³⁾의 라우팅 모델은 Exchange와 Queue 그리고 Binding으로 구성된다.

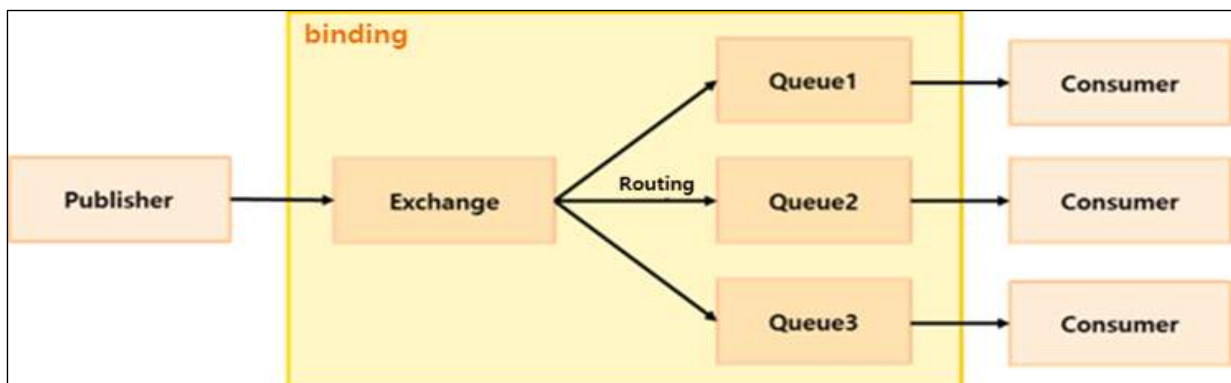


Figure 9. Routing model for RabbitMQ, <http://tmondev.blog.me/>

3) AMQP(Advanced Message Queuing Protocol, 어드밴스트 메시지 큐잉 프로토콜)는 메시지 지향 미들웨어를 위한 개방형 표준 응용 계층 프로토콜

RabbitMQ는 기본적으로 위 Figure 9와 같은 원리로 동작하며, 그 원리는 다음과 같다.

1. Publisher(혹은 Producer)가 메시지를 생산하여 송신한다.
2. Exchange는 Publisher로부터 수신한 메시지를 큐에 분배한다.
3. Queue는 메시지를 메모리나 디스크에 저장했다가 Consumer에게 전달한다.

RabbitMQ는 Management Plug-in Web-UI에서 대부분의 기능을 설정, 관리할 수 있다.

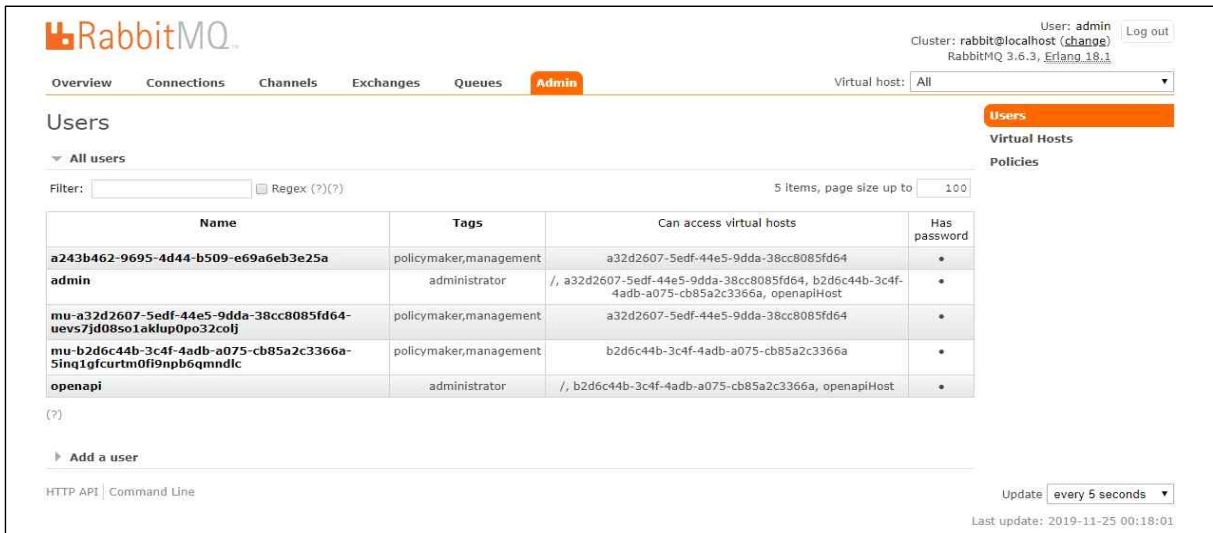


Figure 10. RabbitMQ : Admin - Users Menu

관리자(Admin)로 설정한 RabbitMQ 아이디로 Management Plug-in Web-UI로 접속하면 중앙 상단에 Admin 메뉴가 있다. Admin에는 사용자 계정을 관리하는 Users, Queue와 각각의 계정을 그룹핑할 수 있는 Virtual Host, 각각의 Virtual Host에 관한 정책을 설정하는 Policies 메뉴가 있다.

계정 생성에서 Tag는 계정의 권한을 부여하는 부분으로, Admin, Monitoring, Polycymaker, Management, None 이렇게 5가지 종류가 있으며 계정을 각각의 형태로 설정할 수 있다. Admin 권한은 RabbitMQ에 관한 모든 설정을 변경/조회가능하며, Monitoring의 경우에는 조회만 가능하다. Polycymaker는 각각 VirtualHost에 대한 정책만이 변경 가능하며, Management의 경우에는 Management Plugin을 사용할 수 있는 권한이 주어진다. None의 경우에는 Client에서만 사용할 수 있고, 그 외의 권한은 없다. 여기에 있는 모든 권한은 복수로 설정이 가능하다.

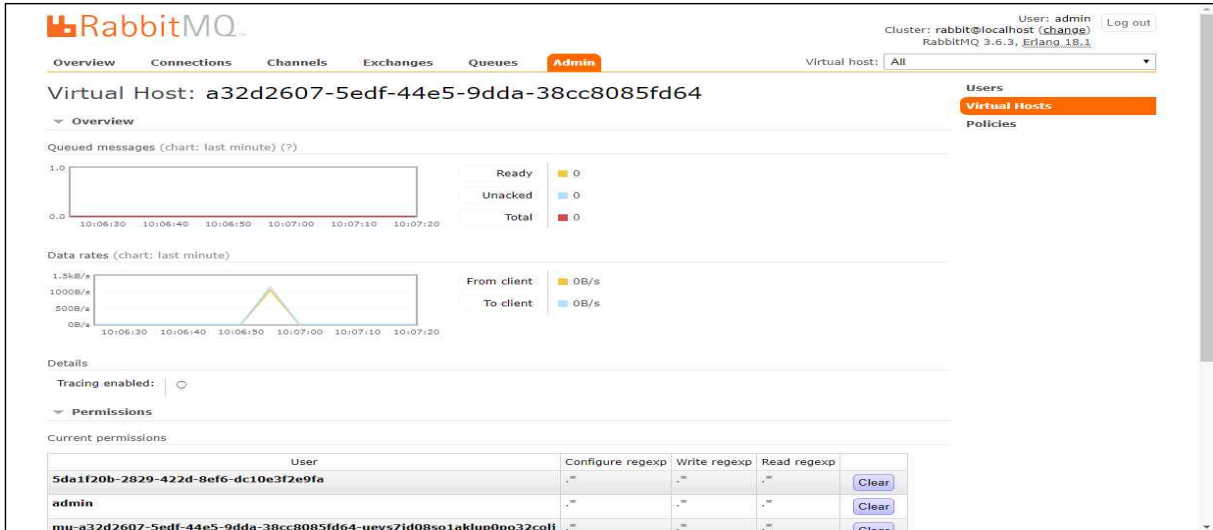


Figure 11. RabbitMQ : Admin - Virtual Host Menu

Virtual Host는 Queue와 User를 그룹핑하는 개념이다. 하나의 계정은 Virtual Host를 여러 개 할당 받을 수 있으며(1:N 매칭 가능), 자신에게 할당된 Virtual Host에 속한 Queue에만 접근이 가능한 개념이다. Management Plug-in에서 Admin - Virtual Hosts 메뉴에 들어가면 위 Figure 11와 같은 설정 화면이 나온다. Virtual Host는 서비스별로 서로 독립된 구역을 나눌 수 있고, 계정 별로 여러 Virtual Host를 설정하여 쉽게 관리할 수 있다.

Queue를 생성하기 위해서는 상단의 Queues 메뉴로 들어가서 Add a new queue 메뉴를 선택한다. Add a new queue 메뉴에는 아래와 같은 속성이 존재한다.

Virtual Host : Queue가 사용될 Virtual Host를 설정한다. Admin - Virtual Hosts에서 생성된 Virtual Host를 사용할 수 있다.

Name : Queue의 이름을 설정한다.

Durability : Durable과 Transient 둘 중 하나를 선택한다. Durable은 메시지를 디스크에 저장하는 것이며 Memory에 저장하는 것을 Transient라고 한다.

Auto-Delete : 모든 Consumer가 해당하는 Queue를 Unsubscribe 하면 해당 Queue는 자동으로 삭제된다.

RabbitMQ에서는 queue를 생성하는 것을 declare 라고 표현하며, RabbitMQ Management Plug-in Web UI에서 만들 수만 있는 것이 아니라, RabbitMQ CLI나 각각의 어플리케이션 코드에서도 쉽게 만들 수 있다. 이렇게 생성한 queue는 해당 queue가 이미 존재하고 있다면, 다시 queue를 만들지 않고, queue가 없을 경우에만 만든다.

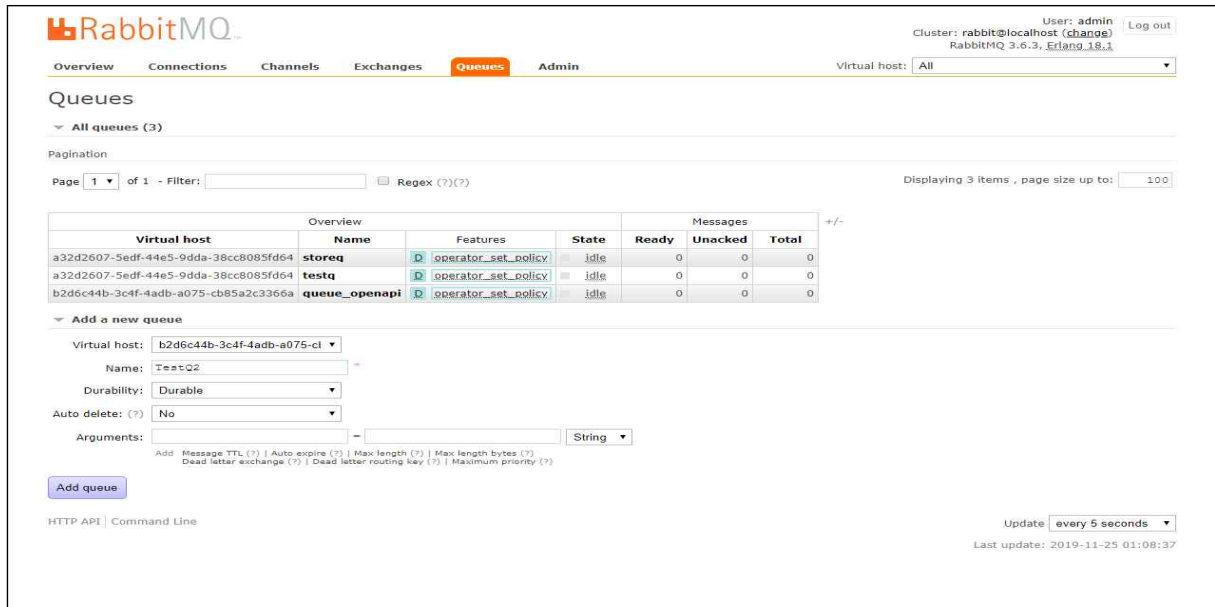


Figure 12. RabbitMQ : Queues – Add a new queue Menu

RabbitMQ의 기본 동작 원리인 Figure 9를 다시 한 번 참조하면, Exchange는 Publisher로부터 수신한 메시지를 큐에 분배하는 라우터의 역할을 하며, Queue는 다시 그 메시지를 잠시 보관하였다가 Consumer에게 전달한다. 여기에서 Binding은 Exchange와 Queue의 관계를 정의하는 방식을 말한다.

RabbitMQ에서는 여러 가지 Exchange Type을 제공하는데, 이 Exchange Type은 메시지를 어떠한 방식으로 라우팅할지를 결정하는 방법이라고 할 수 있으며, 그 Type의 종류는 아래와 Table 5과 같다.

Table 5. List of Exchange Types supported by RabbitMQ, <https://www.rabbitmq.com/>

Exchange Type	정의
Direct Exchange	Exchange에 바인딩 된 Queue 중에서 메시지의 라우팅 키와 매핑되어 있는 Queue로 메시지를 전달한다. 각 Queue는 Routing Key에 Binding이 되어 있고, Exchange에 Routing Key가 들어오면, 그 Exchange에 Binding되어 있는 Queue중에서, 그 Key와 Mapping되어 있는 Queue로 메시지를 라우팅 한다. 1:1 관계로 Unicast 방식에 적합하며, 주로 라운드 로빈 방식으로 여러 workers(Consumer)간 task를 분리에 사용된다.
Fanout Exchange	Routing Key에 상관 없이 Exchange에 Binding되어 있는 모든 Queue에 메시지를 라우팅 한다. (1:N 관계로, 모든 Queue에 메시지를 복제해서 라우팅 한다.) 즉, 메시지의 라우팅 키를 무시하고 Exchange에 바인딩 된 모든 Queue에 메시지를 전달한다. 1:N 관계로 메시지를 브로드캐스트 하는 용도로 사용된다.
Topic Exchange	Exchange에 바인딩 된 Queue 중에서 메시지의 라우팅 키가 패턴에 맞는 Queue에게 모두 메시지를 전달한다. Multicast 방식에 적합하다.
Headers Exchange	라우팅 키 대신 메시지 헤더에 여러 속성들을 더해 속성들이 매칭되는 큐에 메시지를 전달한다.

Exchange 역시도 Queue와 비슷한 방식으로 생성이 가능하다. 상단의 Exchanges 메뉴로 들어가서 Add a new exchange 메뉴를 선택한다. Add a new exchange 메뉴에는 아래와 같은 속성이 존재한다.

Virtual Host : Exchange가 사용될 Virtual Host를 설정한다.

Name : Exchange의 이름을 설정한다.

Type : Headers, Topic, Direct, Fanout 중 하나를 선택한다.

Durability : Durable(메시지를 디스크에 저장)과 Transient(메시지를 메모리에 저장) 둘 중 하나를 선택한다.

Auto-Delete : 해당하는 Exchange를 사용하지 않으면, 해당 Exchange는 자동으로 삭제

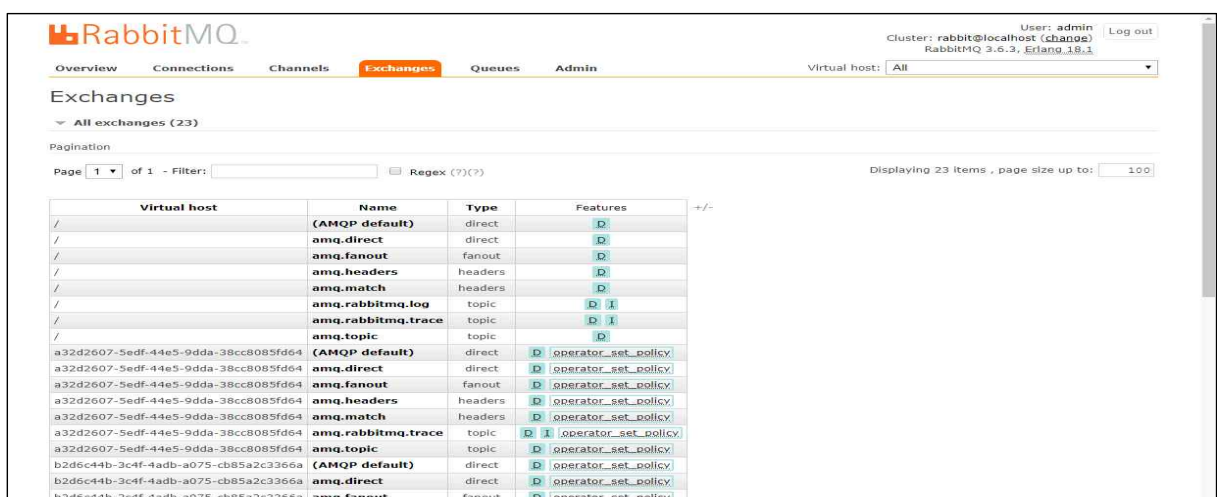


Figure 13. RabbitMQ : Exchanges Menu

Connection 메뉴에서는 해당 Connection의 세부 사항을 모니터링 하거나 강제 종료할 수 있다. 아래 Figure 14처럼 상단의 Connection 메뉴로 들어가면 현재 연결 되어 있는 Connection의 상세한 정보를 확인할 수 있으며, 필요 이상으로 Connection이 강제 종료하도록 설정이 가능하다.

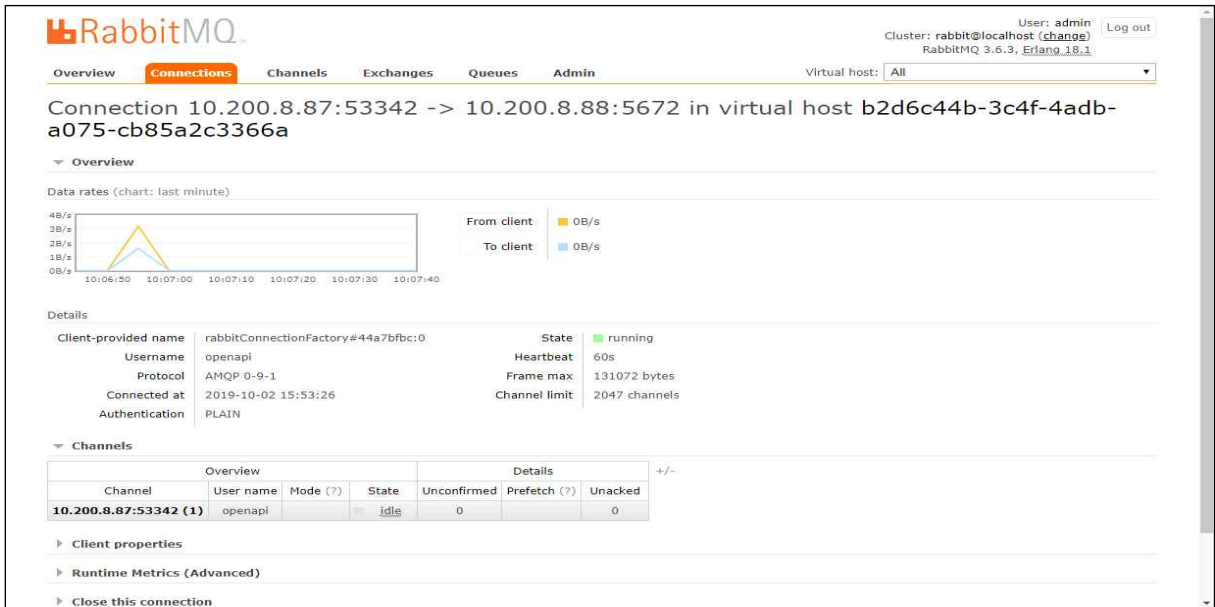


Figure 14. RabbitMQ : Connections Menu

(7) 기후서비스 플랫폼 내 DBMS(Database Management System) 구축

기후서비스 플랫폼을 이루고 있는 PaaS-TA에서는 다양한 DBMS를 설치하여 사용할 수 있다. 기후서비스 플랫폼에는 널리 사용되는 1종의 RDBMS(MySQL), 2종의 No-SQL(Redis, MongoDB) 총 3종의 DBMS를 설치하였고, 본 절에서는 이와 관련된 내용을 기술하고자 한다.

DBMS는 DataBase Management System의 약자로 사용자들이 데이터베이스 내의 데이터를 접근할 수 있도록 해주는 소프트웨어 도구를 말한다. DBMS를 사용하면 사용자나 응용 프로그램의 요구를 처리하여 데이터를 사용할 수 있도록 해준다.

NoSQL 데이터베이스는 전통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 이용하는 데이터의 저장 및 검색을 위한 매커니즘을 제공한다. NoSQL 데이터베이스는 단순 검색 및 추가 작업을 위한 매우 최적화된 키 값 저장 공간으로, 레이턴시와 스루풋과 관련하여 상당한 성능 이익을 내는 것이 목적이다. NoSQL 데이터베이스는 빅데이터와 실시간 웹 애플리케이션의 상업적 이용에 널리 쓰인다. 또, NoSQL 시스템은 SQL 계열 쿼리 언어를 사용할 수 있다는 사실을 강조한다는 면에서 “Not only SQL“로 불리기도 한다. (출처 : 위키백과, <https://ko.wikipedia.org/wiki/NoSQL>)

먼저 일반적으로 사용되는 관계형 데이터베이스인 MySQL의 구조를 설명하고자 한다. PaaS-TA 내의 MySQL 서비스 팩은 다음 Figure 15와 같은 구조를 지니고 있다.

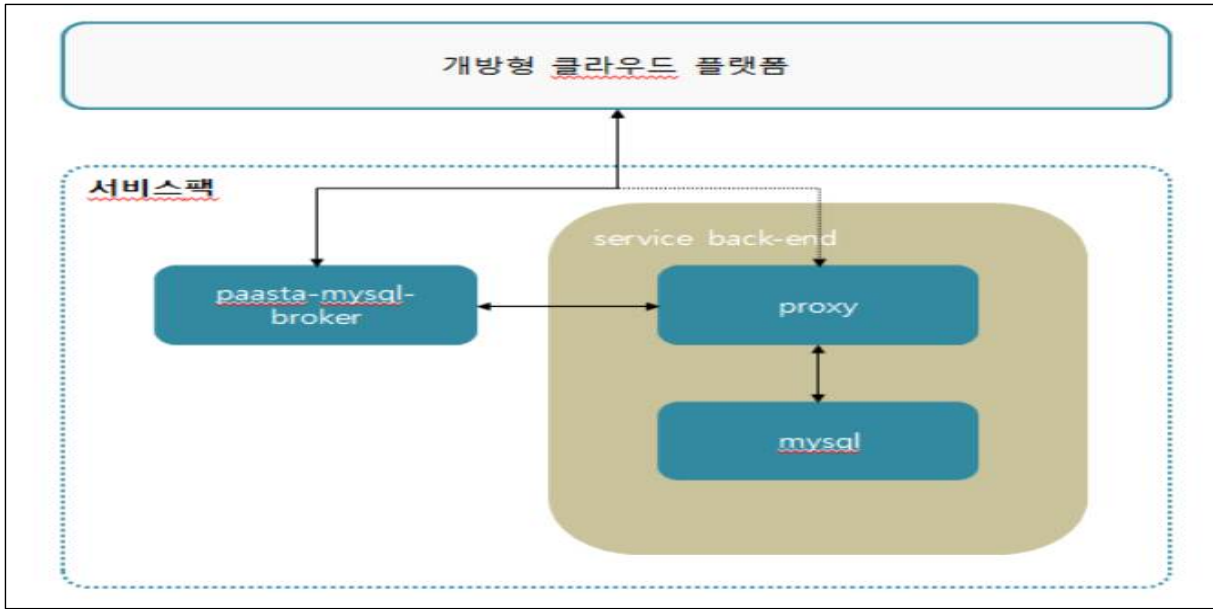


Figure 15. DBMS - MySQL System Configuration Diagram within Climate Service Platform PaaS-TA, <http://paas-ta.kr>

MySQL 자체 서버와 그 Proxy서버가 서비스 백엔드 부분을 이루고 있으며, 이 서비스 백엔드와 PaaS-TA 사이의 통신은 Service Broker가 주관하는 형식으로 구성되어 있다.

MySQL 서비스팩은 BOSH를 이용하여 설치하였으며, manifest 파일과 셸 스크립트로 그 설치가 이루어졌다.

다음은 NoSQL의 하나인 MongoDB이다. 단순히 Proxy와 DB서버로 이루어진 MySQL과 달리 비정형 데이터베이스로 아래 Figure 16과 같이 14개의 서버로 이루어져 있다.

```

joohyung@ubuntu:~/workspace/paasta-4.0/stemcells$ bosh -d 'paasta-mongodb-shard-service' vms
Using environment '10.200.8.212' as client 'admin'

Task 795645. Done

Deployment 'paasta-mongodb-shard-service'

Instance                                Process State AZ  IPs                                VM CID                                VM Type Active
mongodb_broker/368b1d5c-abad-442d-8bf8-36e2deb1ff2f  running      z1  10.200.8.114                       vm-ee29ba3b-54b6-450e-95f3-8d46099071dc minimal true
mongodb_config/4eab5fa9-34ab-4b07-8340-7301a12a6f7f  running      z1  10.200.8.110                       vm-49a9b11e-c9fb-4541-b661-3ec0a5fc6452 minimal true
mongodb_config/9b036f75-9f43-4df1-8eaf-ae80d90a7126  running      z1  10.200.8.111                       vm-ccbff020c-4139-47d4-b354-42692394c24a minimal true
mongodb_config/d28d2ad2-d076-4cd8-a9ed-28e3e961c6ed  running      z1  10.200.8.112                       vm-7f9bddd3-d7a4-4f9b-bfc3-846ec422b20e minimal true
mongodb_master1/08948544-0fe7-46e0-b74b-3ef1c25483a0  running      z1  10.200.8.101                       vm-151fd920-e9a8-4a30-a07a-4f18dc51c681 minimal true
mongodb_master2/b2dd774e-f43b-4763-a981-7520ebab602c  running      z1  10.200.8.104                       vm-a64255e9-65de-48f8-983c-350e87a2c274 minimal true
mongodb_master3/4f852b86-8608-477d-9eee-fe91e0514f58  running      z1  10.200.8.107                       vm-01dc4355-079c-4ff1-9279-6afc30b813d4 minimal true
mongodb_shard/a9f15e53-8549-41a0-992c-07b07b42e878  running      z1  10.200.8.113                       vm-f987f849-14ae-4181-bb43-d9e52d82dac2 minimal true
mongodb_slave1/296bf11c-274c-4c67-964b-be13125f5e11  running      z1  10.200.8.103                       vm-b0ff5c2c-1629-4236-a17f-68a3e91007f8 minimal true
mongodb_slave2/5168d56d-e48c-420b-9707-d0757f7babea  running      z1  10.200.8.102                       vm-633646a2-fabc-4ecb-a0a3-3beecddb6157 minimal true
mongodb_slave2/2d6ccd7c-4aa3-459e-95d5-5932b364424e  running      z1  10.200.8.106                       vm-e91bee63-e32b-4586-9703-df0bc7236775 minimal true
mongodb_slave2/9f408741-6d76-4a50-8797-152afa3b365a  running      z1  10.200.8.105                       vm-a6cf5d17-383c-4643-932b-da1259665sec5 minimal true
mongodb_slave3/3988619f-c172-4925-b266-4feab13cbc4c  running      z1  10.200.8.109                       vm-c2d38e6d-f10c-461d-8221-fa8739dde186 minimal true
mongodb_slave3/b8ee5271-d322-44a4-9060-6e4300d8542e  running      z1  10.200.8.108                       vm-29396a58-d5c6-4443-9010-ecb49ae903fa minimal true

14 vms

Succeeded
joohyung@ubuntu:~/workspace/paasta-4.0/stemcells$

```

Figure 16. List of installed MongoDB VMs within the climate service platform

기후서비스 플랫폼에 설치된 MongoDB의 세부구성은 1개의 Service Broker, 1개의 MongoDB Shard 서버, 3개의 MongoDB Config Server, 1개의 Replica 당 3개의 서버(1개의 Master, 2개의 Slave)가 총 3개로 이루어져 있다.

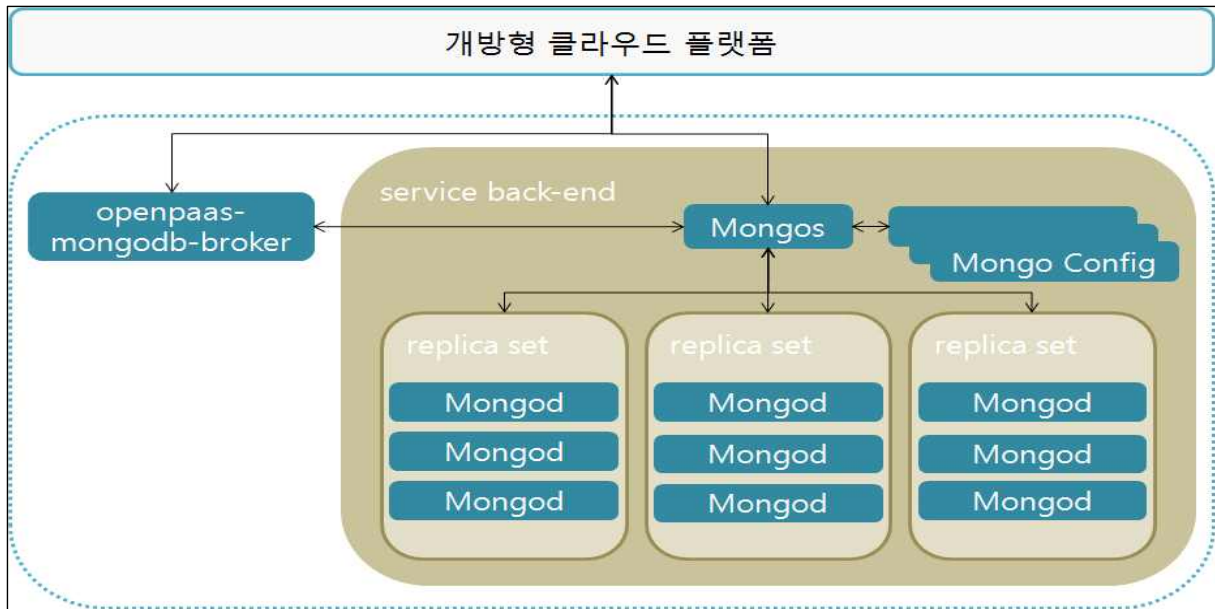


Figure 17. Diagram of the DBMS - MongoDB System in the Climate Service Platform PaaS-TA, <http://paas-ta.kr>

MongoDB 자체는 Replication을 지원한다. 이는 서비스의 지속성과 안전성을 제공하는 데이터베이스 시스템의 설비라고 할 수 있다. MongoDB는 단순히 데이터 복제를 위한 것뿐만 아니라 Master에 장애가 발생 시 이를 Slave를 Master로 자동승격 시키는 기능이 있다.

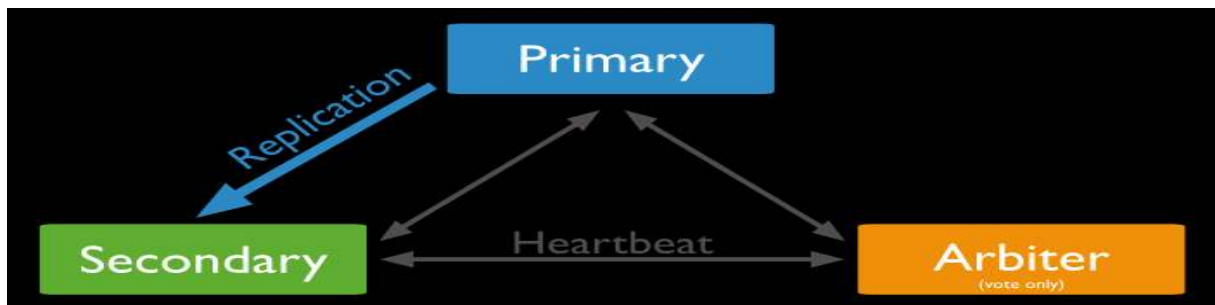


Figure 18. MongoDB Replica Set, <https://www.mongodb.com/>

MongoDB 서비스팩 역시 BOSH를 이용하여 설치 완료하였으며, manifest 파일과 셸 스크립트로 기후서비스 플랫폼 환경에 알맞게 그 설치가 이루어졌다.

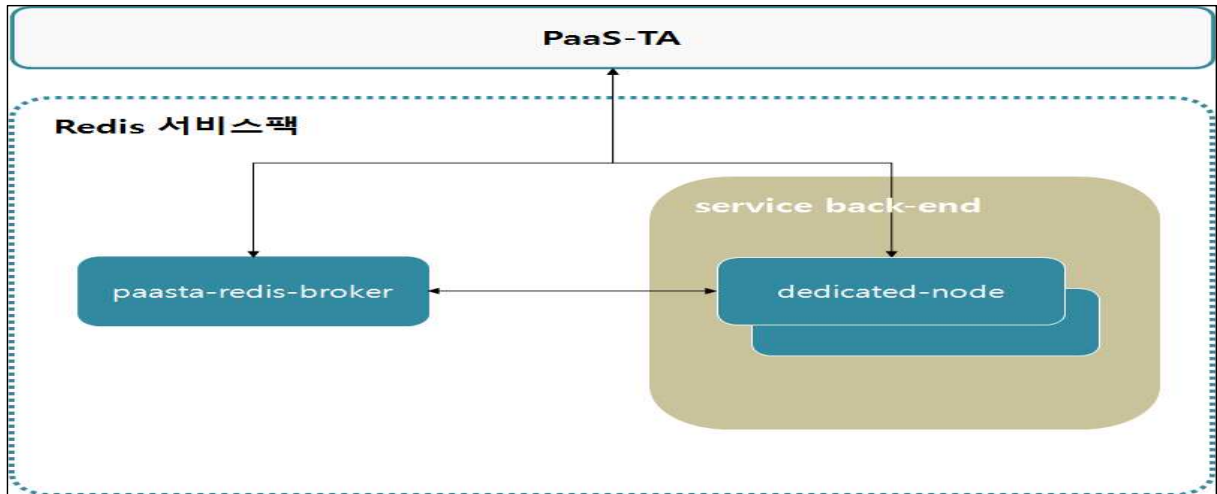


Figure 19. DBMS - Redis System Configuration Diagram within Climate Service Platform PaaS-TA, <http://paas-ta.kr>

다음은 NoSQL의 하나인 Redis이다. PaaS-TA 내 Redis 시스템 구성도는 위 Figure 19와 같다. MongoDB처럼 대량의 Replication을 위한 Master-Slave 구조나 Replication Set가 대량이 있지는 않지만 각각의 독립된 Dedicated Server를 가지고 있으며, 이 Dedicated Server는 Replication을 위하여 추가적인 관리가 가능하다.

Redis 역시 BOSH를 이용하여 설치하였으며, 1개의 Redis-Service-Broker와 1개의 Dedicated-Node(Server)로 구성하였다.

(8) PaaS-TA 내 DBMS 활용을 위한 매뉴얼

이 절에서는 MongoDB를 활용하여 일반적인 DBMS에서 사용하는 방법을 설명하고자 한다. BOSH를 사용하여 정상적으로 MongoDB 서비스팩 배포 및 설치가 완료되었으면, 사용자가 직접 DBMS에 접근하거나 어플리케이션 단계에서 DBMS와 통신하기 위하여 먼저 MongoDB 서비스 브로커를 등록해주어야 한다.

먼저 `cf service-brokers` 명령을 통하여 서비스 브로커의 목록을 확인한다.

Table 6. cf service-brokers command queries the list of service brokers

```

joohyung@ubuntu:~/workspace$ cf service-brokers
Getting service brokers as admin...

name                url
container-service-broker  http://10.200.8.7:8888
delivery-pipeline      http://10.200.8.124:8080
glusterfs-service      http://10.200.8.142:8080

joohyung@ubuntu:~/workspace$

```

현재 CF 상에 등록된 Service Broker가 없으므로 MongoDB 서비스 브로커를 등록하고 등록된 서비스 브로커를 확인한다.

Table 7. cf create-service brokers and query with the create-service-brokers command

```

joohyung@ubuntu:~/workspace$ cf create-service-broker mongodb-shard-service-broker [ID]
[PASSWD] http://10.200.8.114:8080
Creating service broker mongodb-shard-service-broker as admin...
OK
joohyung@ubuntu:~/workspace$ cf service-brokers
Getting service brokers as admin...

name                url
container-service-broker  http://10.200.8.7:8888
delivery-pipeline      http://10.200.8.124:8080
glusterfs-service      http://10.200.8.142:8080
mongodb-shard-service-broker  http://10.200.8.114:8080
rabbitmq-service-broker  http://10.200.8.86:4567

joohyung@ubuntu:~/workspace$

```

이후 접근 다른 어플리케이션이나 사용자가 해당 서비스 브로커에 접근 가능하도록 서비스 access를 enable시키고 다시 한 번 cf service-access 명령을 통해 접근 권한을 확인한다.

Table 8. cf enable-service-access Mongo-DB command after cf service-access

```

joohyung@ubuntu:~/workspace$ cf enable-service-access Mongo-DB
Enabling access to all plans of service Mongo-DB for all orgs as admin...
OK

joohyung@ubuntu:~/workspace$ cf service-access
Getting service access as admin...

broker: container-service-broker
  service      plan      access  orgs
container-service  Advanced  all
container-service  Micro    all
container-service  Small    all

broker: delivery-pipeline
  service      plan      access  orgs
delivery-pipeline-v2  delivery-pipeline-dedicated  all
delivery-pipeline-v2  delivery-pipeline-shared     all

broker: glusterfs-service
  service  plan      access  orgs
glusterfs  glusterfs-1000Mb  all
glusterfs  glusterfs-100Mb   all
glusterfs  glusterfs-5Mb     all

broker: mongodb-shard-service-broker
  service  plan      access  orgs
Mongo-DB  default-plan  all

broker: rabbitmq-service-broker
  service  plan      access  orgs
p-rabbitmq  standard  all

joohyung@ubuntu:~/workspace$

```

위 Table 8과 같이 정상적으로 Mongo-DB 서비스가 Access Enable되는 과정을 확인할 수 있다. 이제 Mongo-DB에 접속할 수 있는 Sample Application을 활용하여 다른 어플리케이션에서 해당 Mongo-DB 서비스로 접속하고자 한다.

Sample Web App은 PaaS-TA 공식 GitHub에서 다운로드 받을 수 있는 어플리케이션으로 PaaS-TA에 어플리케이션(컨테이너)으로 배포된다. 어플리케이션을 배포하여 구동시 Bind된 MongoDB 서비스 연결 정보로 접속하여 초기 데이터를 생성하며, 배포 완료 후 정상적으로 App이 구동되면 브라우저를 통하여 해당 어플리케이션에 접속하여 MongoDB 환경 정보(서비스

연결 정보)와 초기 적재된 데이터를 확인할 수 있다. 더불어 PaaS-TA 사용자 포탈에서도 해당 어플리케이션의 환경 정보를 확인할 수 있다.

```
PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb> cf marketplace
Getting services from marketplace in org system / space apcc as admin...
OK

service      plans      description
Mongo-DB     default-plan  A simple mongo implementation
container-service  Micro, Small, Advanced  For Container Service Plans, You can choose plan about CPU, Memory, disk.
delivery-pipeline-v2  delivery-pipeline-shared+, delivery-pipeline-dedicated+  A paasta source control service for application development.provision parameters : parameters
glusterfs    glusterfs-5Mb, glusterfs-100Mb+, glusterfs-1000Mb+  A simple glusterfs implementation
p-rabbitmq   standard     RabbitMQ is a robust and scalable high-performance multi-protocol messaging broker.

* These service plans have an associated cost. Creating a service instance will incur this cost.

TIP: Use 'cf marketplace -s SERVICE' to view descriptions of individual plans of a given service.
PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb> cf create-service Mongo-DB default-plan mongodb-test
Creating service instance mongodb-test in org system / space apcc as admin...
OK
PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb> cf services
Getting services in org system / space apcc as admin...

name          service      plan          bound apps      last operation
glusterfs-service-instance  glusterfs    glusterfs-1000Mb  mongodb-instance  create succeeded
mongodb-service-instance    Mongo-DB     default-plan      mongodb-instance  create succeeded
mongodb-test                 Mongo-DB     default-plan      mongodb-instance  create succeeded
my_rabbitmq_service         p-rabbitmq   standard          rabbitmq-instance  create succeeded
request                 p-rabbitmq   standard          rabbitmq-instance  create succeeded
PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb>
```

Figure 20. The screen of mongodb-test service created through the CF command.

Figure 20과 같이 CF create-service 명령을 통해 mongodb-test라는 서비스를 생성하였다. 이 다음 PaaS-TA GitHub에서 다운로드 받은 Sample 어플리케이션(hello-spring-mongodb)를 배포(push)한다. 단, 여기서 주의할 점은 어플리케이션 배포는 하되 시작하기 전에 서비스를 바인딩하기 위하여 Figure 21과 같이 cf push --no-start 옵션으로 배포해야 한다.

```
PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb> cf push --no-start
Pushing from manifest to org system / space apcc as admin...
Using manifest file C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb\manifest.yml
Getting app info...
Creating app with these attributes...
+ name: hello-spring-mongodb
+ path: C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb\build\libs\hello-spring-mongo
db.war
+ instances: 1
+ memory: 1G
+ routes:
+ hello-spring-mongodb.cf.apcc21.org

Creating app hello-spring-mongodb...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
 491.41 KiB / 491.41 KiB [=====] 100.00% 1s

Waiting for API to complete processing files...

name: hello-spring-mongodb
requested state: stopped
routes: hello-spring-mongodb.cf.apcc21.org
last uploaded:
stack:
buildpacks:

type: web
instances: 0/1
memory usage: 1024M
#0 state since cpu memory disk details
#0 down 2019-11-24T20:49:47Z 0.0% 0 of 0 0 of 0
```

Figure 21. The screen of Hello-spring-mongodb Application Deployment

```

PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb> cf bind-service hello-spring-mongodb mon
godb-test
Binding service mongodb-test to app hello-spring-mongodb in org system / space apcc as admin...
OK
TIP: Use 'cf.exe restage hello-spring-mongodb' to ensure your env variable changes take effect
PS C:\00_Data_JH\00_CF\00_Sample_Application\hello-spring-mongodb> cf restart hello-spring-mongodb
Restarting app hello-spring-mongodb in org system / space apcc as admin...

Staging app and tracing logs...
  Downloading binary_buildpack...
  Downloading dotnet_core_buildpack...
  Downloading java_buildpack...
  Downloading staticfile_buildpack...
  Downloading go_buildpack...
  Downloaded binary_buildpack
  Downloading nodejs_buildpack...
  Downloaded dotnet_core_buildpack
  Downloading ruby_buildpack...
  Downloaded java_buildpack
  Downloading python_buildpack...
  Downloaded staticfile_buildpack
  Downloading php_buildpack...
  Downloaded go_buildpack
  Downloaded ruby_buildpack
  Downloaded php_buildpack
  Downloaded nodejs_buildpack
  Downloaded python_buildpack
Cell aed0041d-28b9-4b85-9195-9fed23101d7c creating container for instance 5a33ad4b-3583-4262-aaae-4af715
f1a8ec
Cell aed0041d-28b9-4b85-9195-9fed23101d7c successfully created container for instance 5a33ad4b-3583-4262
-aaae-4af715f1a8ec
  Downloading app package...
  Downloaded app package (17.5M)

```

Figure 22. cf bind-service and application restart

이후 배포된 Sample App을 확인하고 CF 명령(bind-service)을 통해 Figure 22와 같이 Sample Web App에서 생성한 서비스 인스턴스 바인드 신청을 하고 해당 어플리케이션을 재시작한다.

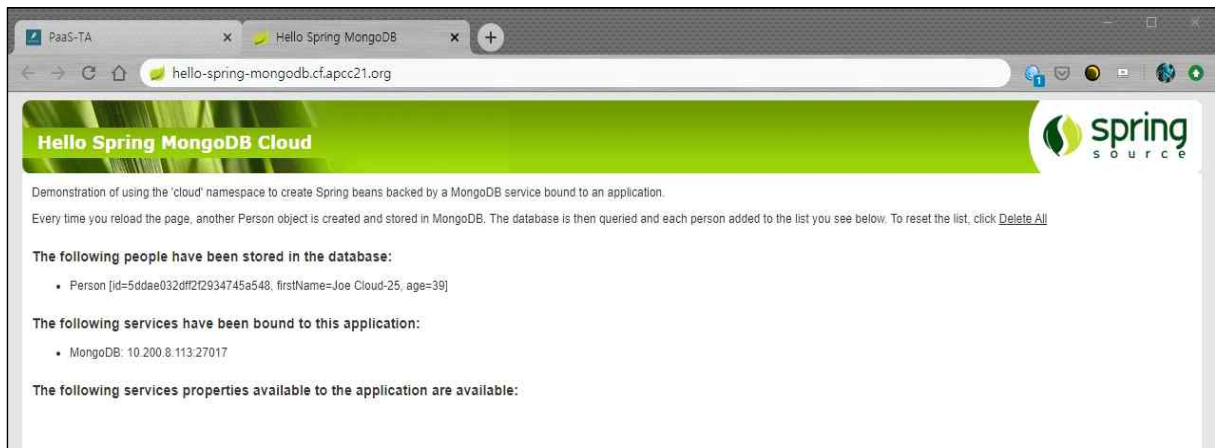


Figure 23. MongoDB load data that can be found in deployed applications

위 Figure 23과 같이 어플리케이션이 정상적으로 배포되었으며 MongoDB에 적재된 Sample 데이터 역시 확인할 수 있다. 더불어 아래 Figure 24와 같이 PaaS-TA 사용자 포털에서 해당 어플리케이션의 환경 변수에서도 해당 인스턴스로 접속할 수 있는 정보를 확인할 수 있으며, 이 정보는 사용자가 직접 접근하거나, 다른 어플리케이션에서 참조할 수 있다.

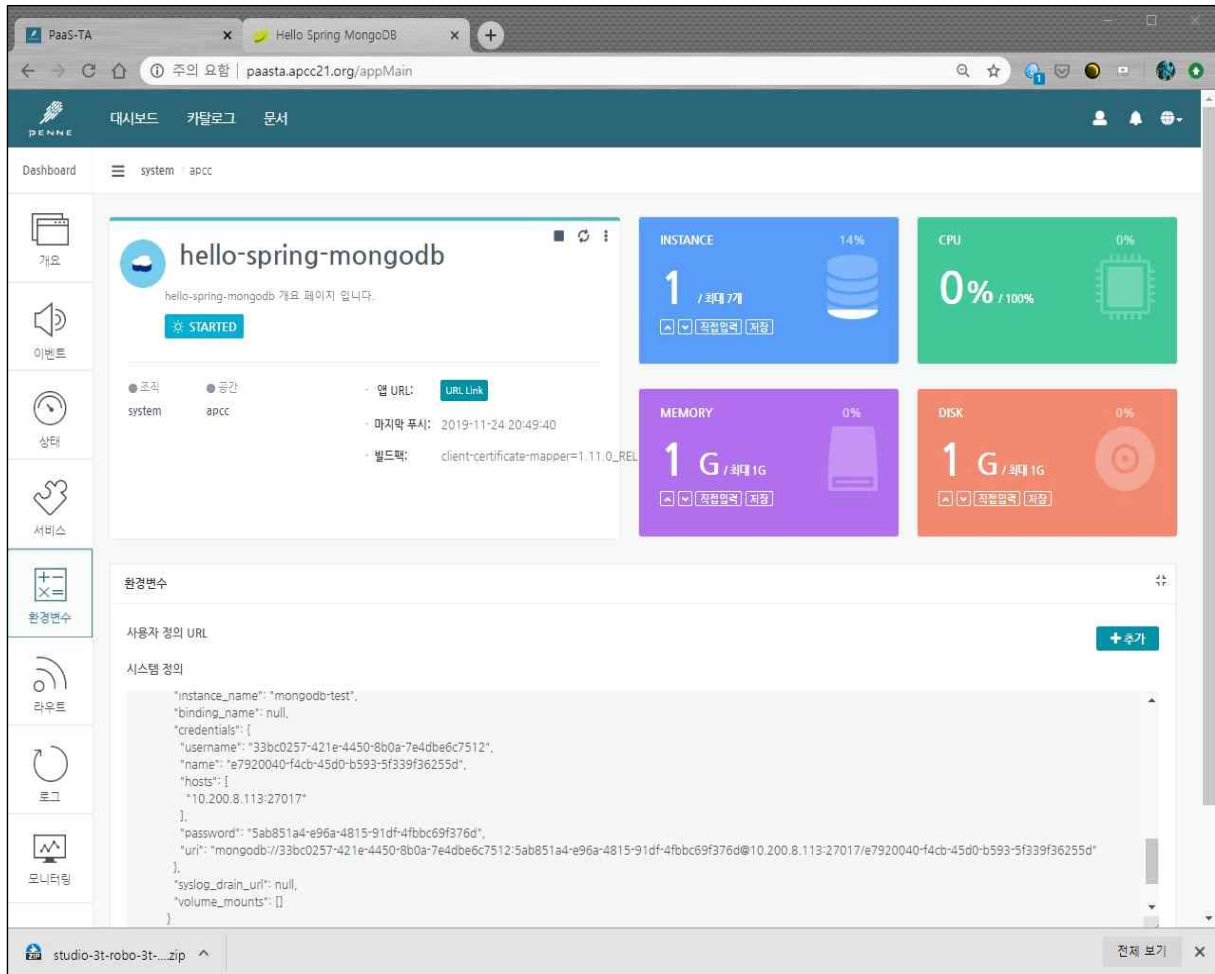


Figure 24. MongoDB Instance Environment Variables Available on the PaaS-TA User Portal

(9) 기후서비스 플랫폼 내 형상 및 배포관리

소프트웨어의 핵심 기술이라고 할 수 있는 소프트웨어 소스코드는 하나의 중요한 산업 자산이 되었으며, 이러한 흐름 때문에 대부분의 소프트웨어 소스코드는 비공개로 개발되었다. 하지만, 시간이 흐를수록 공개 소프트웨어는 개발자들을 위해 다양한 방식과 형태로 전 세계적으로 배포 · 공유되고 있다. 이러한 소프트웨어 소스코드가 공유되는 대표적인 소스코드 공유 사이트로는 SourceForge와 GitHub 등이 있으며, C나 Java등의 컴파일러 언어에서부터 최근 들어 많은 개발자가 사용하고 있는 python이나 Ruby, R 등의 스크립트 언어로 구현된 소프트웨어 소스코드도 널리 공유되고 있다.

이러한 오픈소스 소프트웨어의 경우 다수에게 소스코드가 공개되고, 각 개인의 참여도 및 프로젝트별 저장소의 권한에 따라 소스코드를 다룰 수 있는 권한이 달라진다. 즉, 다수의 개발자가 참여하는 프로젝트의 경우에는 프로그램 저장소의 버전 및 권한 관리가 그 핵심적인 요소가 된다. 소스코드의 버전관리는 동시 개발 협업 체계 구축을 위한 필수적 요소이다. 파일이나 프로젝트 복구 기능이 있어야 하며, 소스 코드 수정, 추가, 삭제 등의 기능의 기록이 로그로 남아야 한다. 이외에도 소스 코드 충돌 해결이나, 버전이나 일정 단위로 따로 저장하여 백

업 기능 등의 편의를 위한 기능들이 필요하다.

(가) Subversion

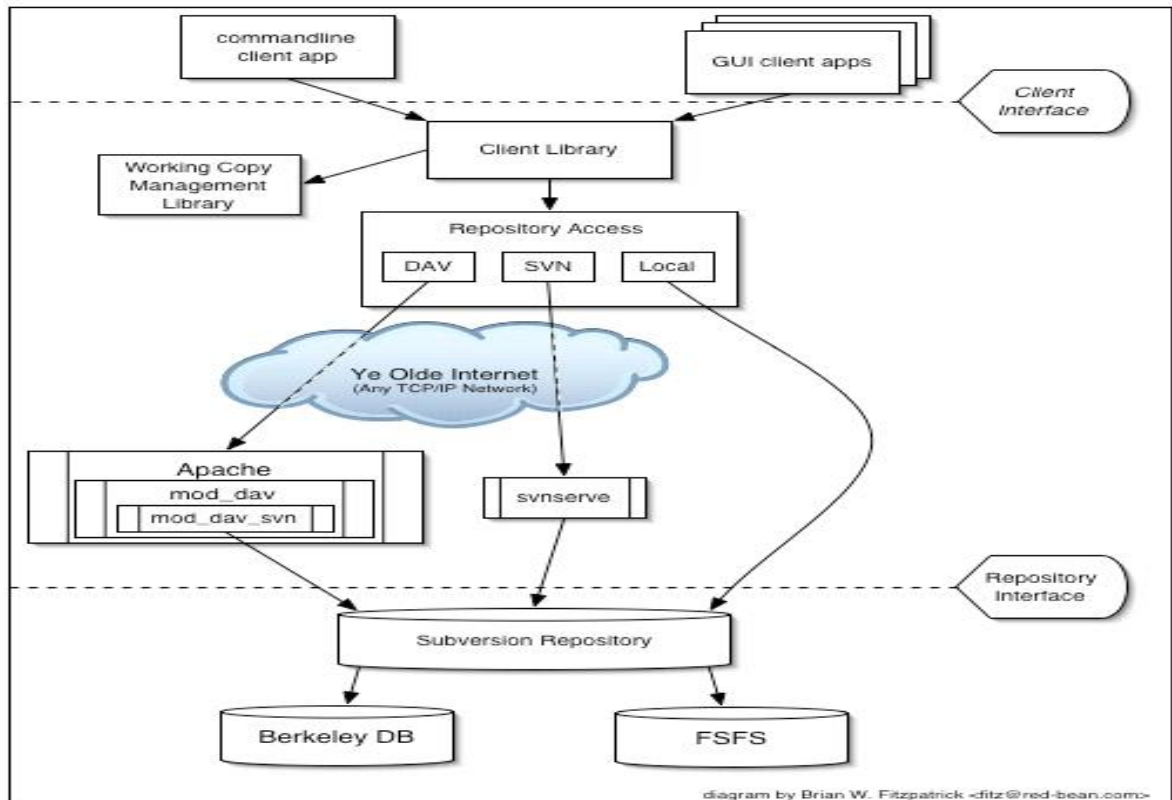


Figure 25. Subversion Schema, <http://svnbook.red-bean.com/>

Subversion은 오픈소스 소프트웨어로 개발된 버전 관리 시스템이다. Command Line에서 실행되는 명령어 svn을 그대로 본따 “SVN” 이라고 줄여서 부르기도 한다. 초기 개발 과정에서 제약이 많았던 CVS(Concurrent Version System)을 대체하기 위하여 2000년부터 CollabNet에서 개발되었으며, 현재 Apache 최상위 프로젝트로 지속적으로 개발되고 있다.

Subversion은 Server-Client 모델을 따른다. Server는 운영체제에 크게 종속되지 않으며, Server와 Client를 동시에 두고 사용이 가능하다. Subversion의 Server와 Client는 http, https, svn, svn+ssh 등의 프로토콜로 통신한다. Subversion은 현재 소개되어진 소스코드 버전 관리 도구는 가장 많이 사용되고 있으며, 다양한 운영체제와 플랫폼, IDE 등에서 활용되고 있다. 도입 비용 없이 활용할 수 있는 오픈소스 소프트웨어라는 점 역시 장점이며, 소스코드 버전을 관리하기 위하여 다양한 기능을 제공한다. Windows 운영체제에서 다양한 클라이언트용 프로그램이 개발되어 있다. 윈도우 탐색기용 플러그인으로 편리한 Client 환경을 제공하는 TortoiseSVN, 편리한 GUI(Graphic User Interface)로 구성되어 서버 관리가 용이한 VisualSVN 등이 있으며, 이외에도 Visual Studio용 플러그인인 AnkhSVN, Eclipse 플러그인으로 Subclipse 등이 있다.

(나) Git

Git은 프로그램 등의 소스코드 관리를 위한 분산형 버전 관리 시스템이다. Server-Client 구조로 된 Subversion이 널리 이용되면서 사용자들은 대용량 파일 전송에 문제점들을 직면하게 되었다. Client가 많아짐에 따라 Server의 부하로 인하여 전송속도 저하, Server 불안정 등의 문제를 발견하게 되었다. 빠른 속도와 분산형 저장소 지원을 통하여 이를 해결하고자, 리눅스의 개발자인 리누스 토발즈가 리눅스 커널 관리를 위해서 개발하였다. 또한 오픈소스 개발의 특성상 다수의 개발자가 실수하는 등의 이유로 프로젝트에 문제점이 생기기도 하는데, 이러한 특성에 맞게끔 설계가 되어 있다. 현재 마이크로소프트, 페이스북, 트위터, 모질라 재단 등에서도 Git을 활용 중에 있다.

Git의 작업 폴더는 전체 기록과 각 기록을 추적할 수 있는 정보를 포함하고 있는 저장소이다. 즉, 클라이언트에 모든 파일을 다 받아서 작업할 수 있게 된다. 작업이 끝나면 Git 원격 저장소로 다시 발행하게 되며, 원 저장소를 보호하기 위한 브랜치가 있어 가지의 개발이 완료될 시 원 저장소와 합칠 수 있으며, 또한 개발 중간중간 태그를 통해 개발을 더 수월하게 할 수 있다. 이런 강력한 Git를 호스팅하는 GitHub.com을 통하여 오픈소스 소프트웨어의 중심지로 자리매김하고 있다. 설치형 Git 서버는 리눅스 서버의 경우 기본적으로 배포본에 있는 패키지만으로 서버를 구동할 수 있는 편리성이 있다.

(10) 기후서비스 플랫폼 내 형상 및 배포 관리의 활용

기후서비스 플랫폼을 이루고 있는 PaaS-TA에서도 기본적인 형상관리를 위한 서비스 팩 패키지를 제공하고 있으며, 그 구조와 목록은 아래 Figure 26, 27과 같다.

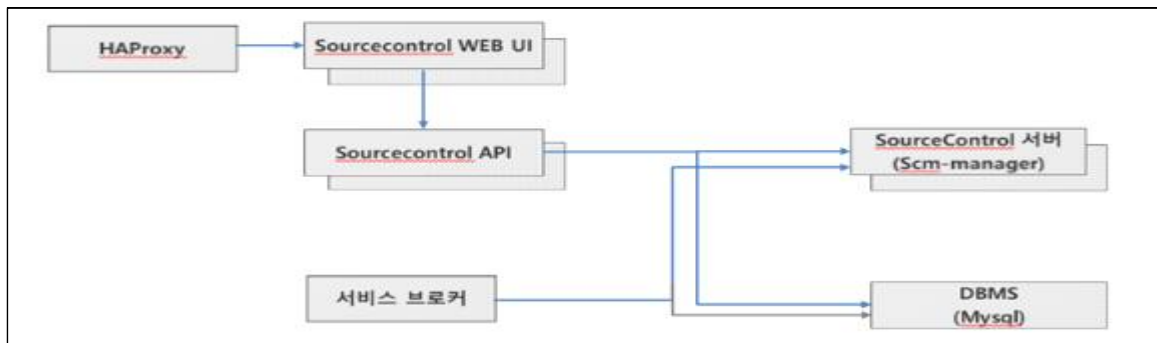


Figure 26. PaaS-TA Configuration Management Service Pack Structure Diagram, <https://github.com/PaaS-TA>

```

joohyung@ubuntu:~/workspace/paasta-4.0/stencells$ bosh -d 'paasta-sourcecontrol-service' vms
Using environment '10.200.8.212' as client 'admin'

Task 793794. Done

Deployment 'paasta-sourcecontrol-service'

Instance Process State AZ IPs VM CID VM Type Active
haproxy/b7dc2164-cd3e-4eaf-9dee-elic5271c5469 running z1 10.200.111.214 vm-1e8b5ed3-0d73-448d-9f19-63cdd7cc336 minimal true
10.200.8.95
mariadb/edd4d349-bea3-4fc8-b7db-0f47d0ee2417 running z1 10.200.8.94 vm-ec1d7fa1-a85f-4223-8f55-42c9e8f46961 minimal true
scm-server/f00f5c67-b2dd-4fae-aa1c-8a007166249f running z1 10.200.8.93 vm-6862af36-2000-42d6-ae4a-e97eb00af893 minimal true
sourcecontrol-api/1cc41ea1-8f8a-47b6-8170-da52bd2cfe29 running z1 10.200.8.97 vm-eh5335a2-e504-4f56-867c-e6f5452ac050 minimal true
sourcecontrol-broker/a2f2d2b6-242f-4299-9552-b3e53554eb6c running z1 10.200.8.98 vm-ba94a8b5-6d5b-4a5a-bfa8-7d6d773ecec8 minimal true
sourcecontrol-webui/42613713-b1f8-4403-be12-083a43a295b1 running z1 10.200.8.96 vm-d03fc475-03d5-4622-a1dd-5f702b55ac46 minimal true

6 vms

Succeeded
joohyung@ubuntu:~/workspace/paasta-4.0/stencells$

```

Figure 27. List of SourceControl service pack packages installed on climate service platforms

Service-Broker는 PaaS-TA와 SourceControl 패키지를 연동하고, HAProxy 서버가 전체적인 접속을 관리하여 SourceControl Web-UI로의 접근을 관장한다. SourceControl WEB-UI는 SourceControl API 서버에 각 기능을 호출하며, API 서버는 SourceControl 서버(SCM-Manager)에 해당 사항을 전달하고 변경된 이력은 MariaDB에 통합되어 관리된다.

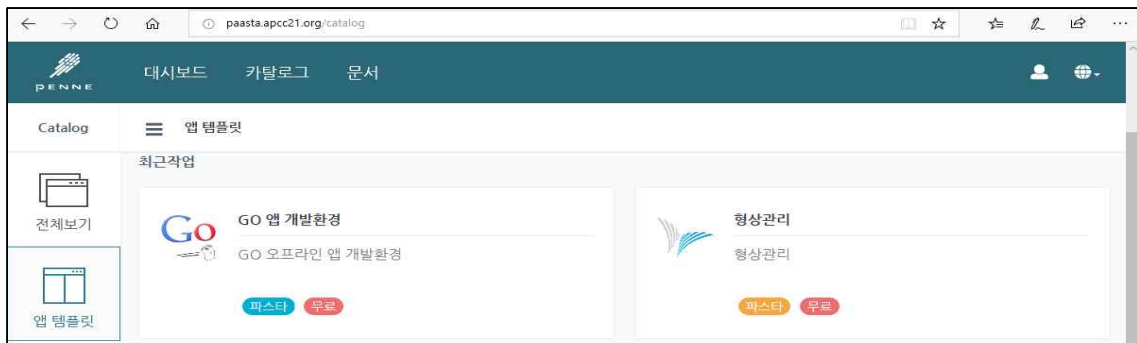


Figure 28. Configuration Management Service screen on the PaaS-TA User Portal

기후서비스 플랫폼의 형상 관리를 사용하고자 하면, 사용자 포탈에서 설정되어 있는 형상 관리 서비스 메뉴를 선택하여 진입하면, 형상관리를 위한 메인페이지로 이동하며, 현재 생성되어 있는 레파지토리(저장소) 목록이 아래 Figure 29와 같이 나타난다.

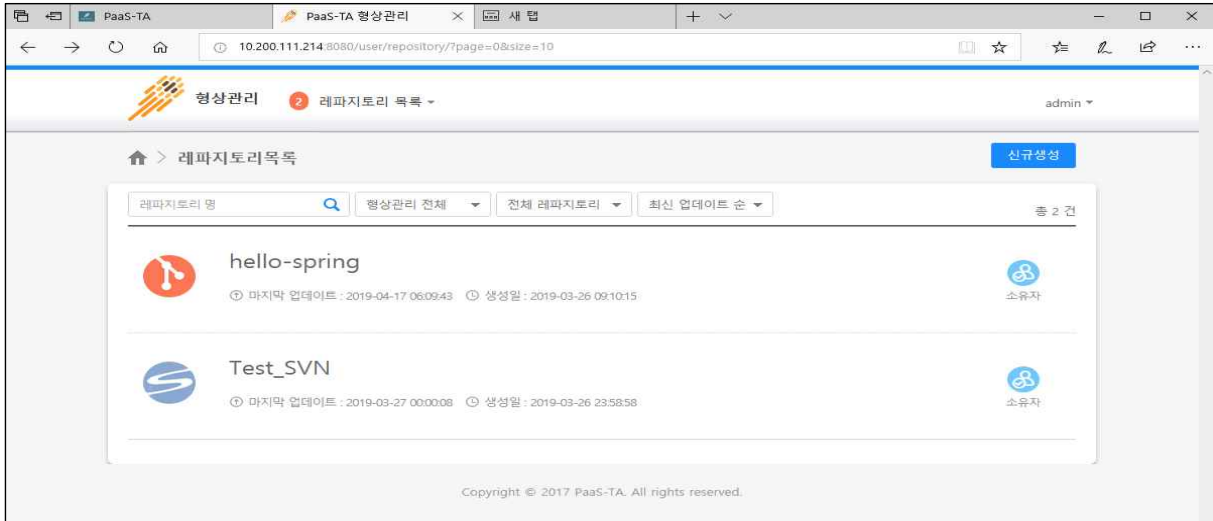


Figure 29. Source Control Management Page Initial Screen (List of Source Control Management Projects)

레파지토리를 선택하면, 아래 Figure 30과 같이 파일(File) 메뉴에서 현재 선택한 레파지토리 내 파일 목록과 최종 변경사항, 파일 크기, 마지막 업데이트 날짜를 확인할 수 있다. 커밋(Commit) 메뉴에서는 레파지토리 커밋 시 생성한 로그에 대한 설명을 확인할 수 있으며 참여자(Contributor) 메뉴에서는 해당 소스코드에 기여한 사용자의 목록을 확인할 수 있다.

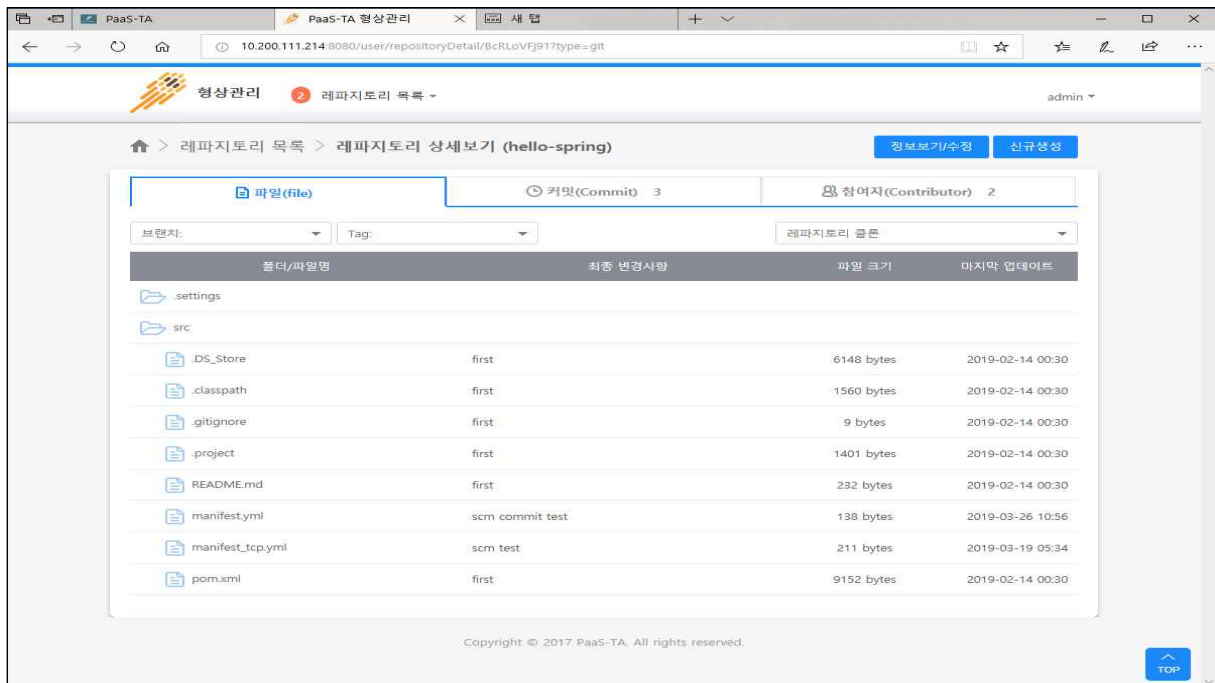


Figure 30. Viewing the details of the repertoire within the Configuration Management page

또한 형상관리 Web-UI에서는 새로운 레파지토리(저장소)를 생성, 삭제, 관리할 수 있다. 다음은 새로운 레파지토리를 생성하는 과정을 설명하고자 한다.

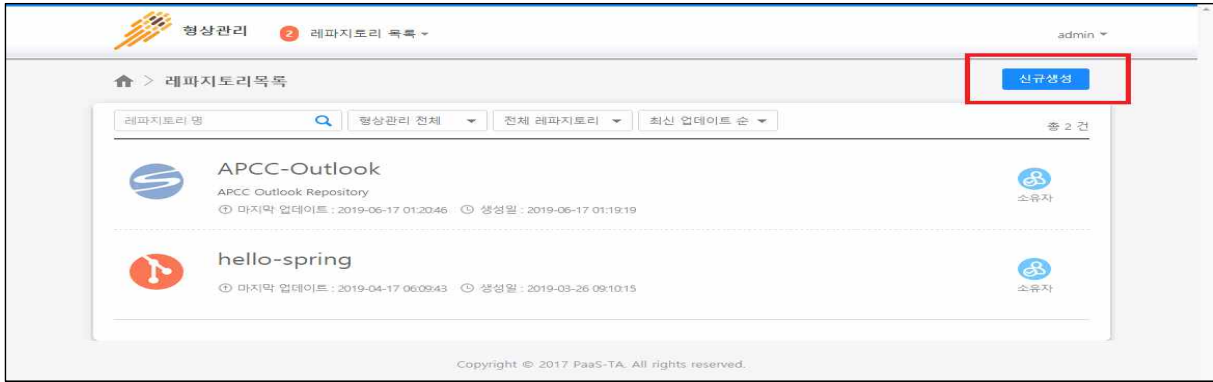


Figure 31. Source Control Management - New Create Menu

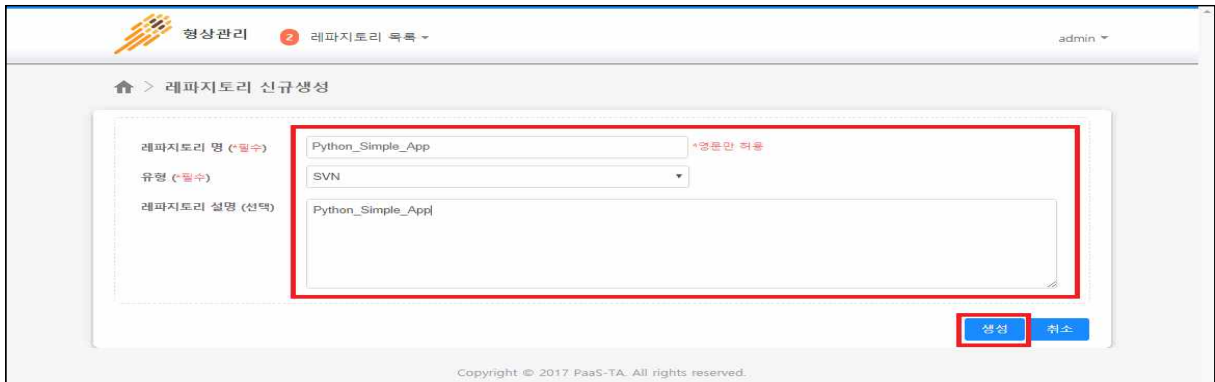


Figure 32. Source Control Management - New Creation Detail Settings Menu

먼저 형상관리 페이지에서 Figure 31과 같이 오른쪽 위의 신규 생성 버튼을 클릭하여 레파지토리 신규생성 화면으로 진입한다. Figure 32와 같이 레파지토리 세부 사항을 설정하는 화면에서 레파지토리 명, 유형, 레파지토리 설명을 기입한다. 현재 기후서비스 플랫폼의 형상관리 메뉴에서는 SVN, Git 2종류의 SCM(Source Control Management)을 지원한다.

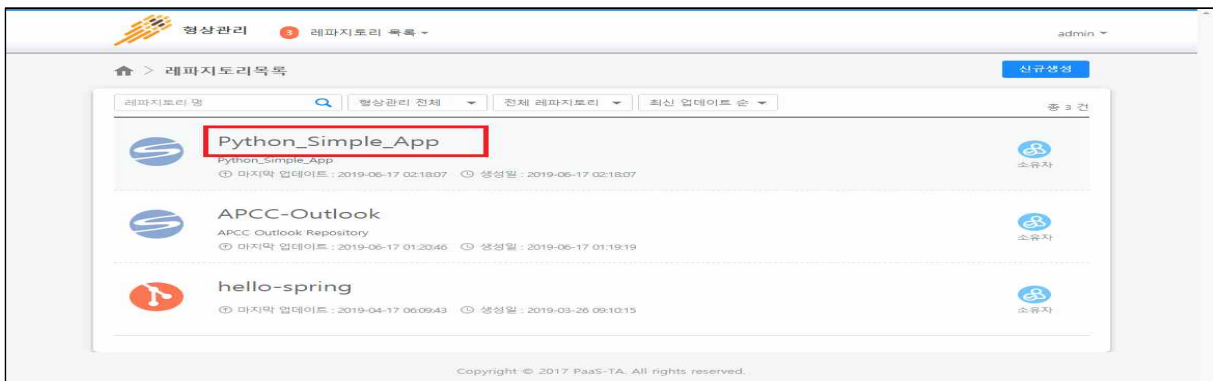


Figure 33. Source Control Management - List of newly created repositories

레파지토리 세부 설정을 마치고 생성 버튼을 클릭하면 Figure 33과 같이 레파지토리 목록에 신규로 생성된 레파지토리 목록을 확인할 수 있다.

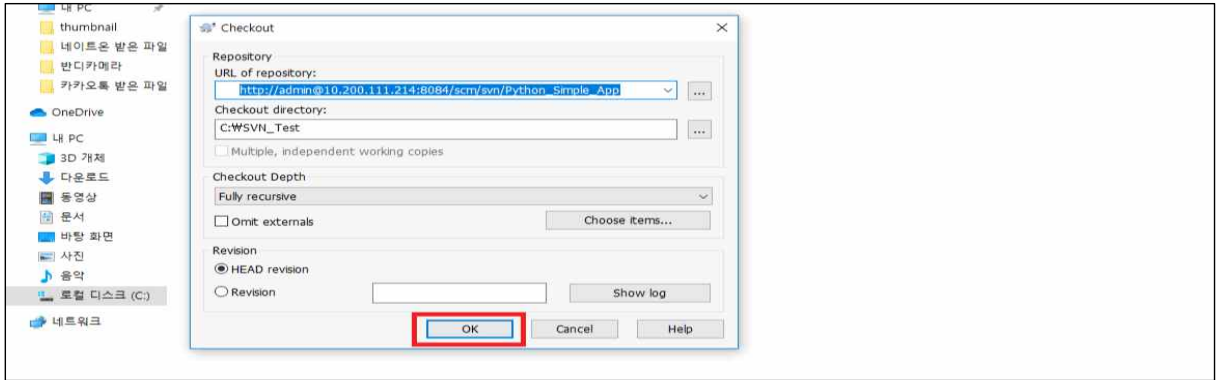


Figure 36. Source Control Management - SVN Checkout Using TortoiseSVN (2)

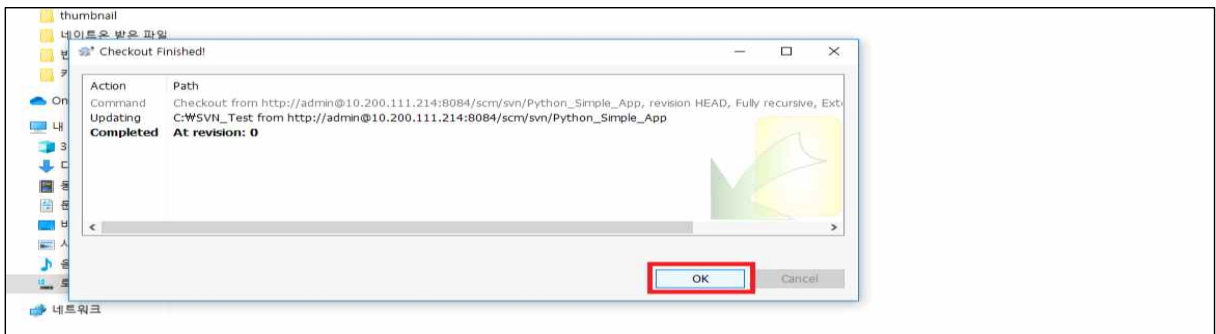


Figure 37. Source Control Management - SVN Checkout Using TortoiseSVN (3)

Figure 36에서 복사하였던 SVN 레파지토리 주소를 나타내는 TortoiseSVN URL에 입력한 후 OK 버튼을 눌러 다음을 진행하면 Figure 37과 같이 Checkout에 관련된 내용이 나타나며 Checkout이 완료된다.

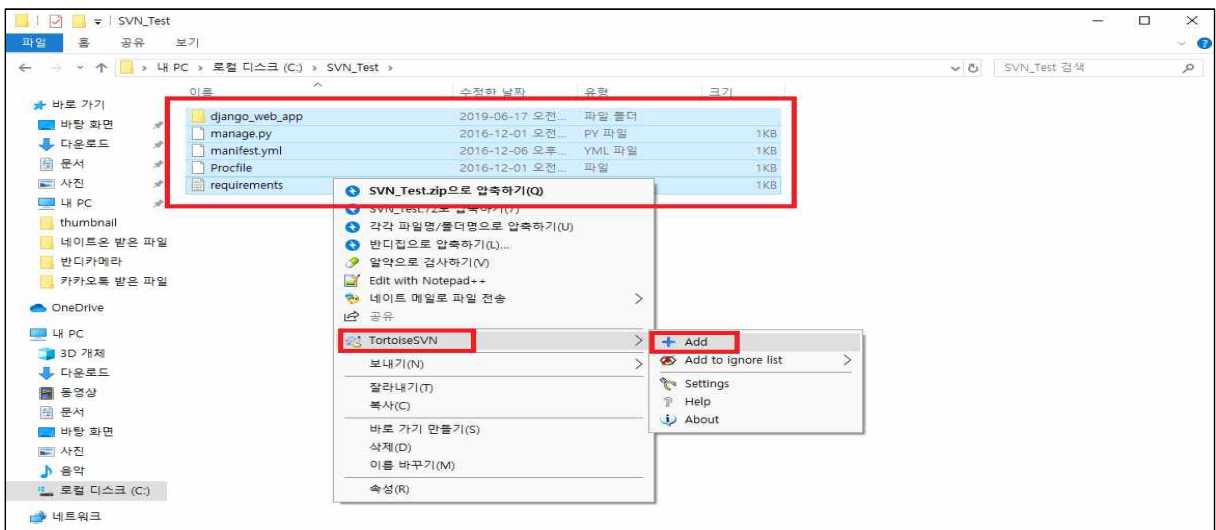


Figure 38. Source Control Management - SVN Add Using TortoiseSVN (1)

현재 상태로는 비어 있는 Repository가 로컬 컴퓨터에 Checkout 되었으므로, 정상적으로

SVN에 소스코드를 Figure 38과 같이 Add 한다.

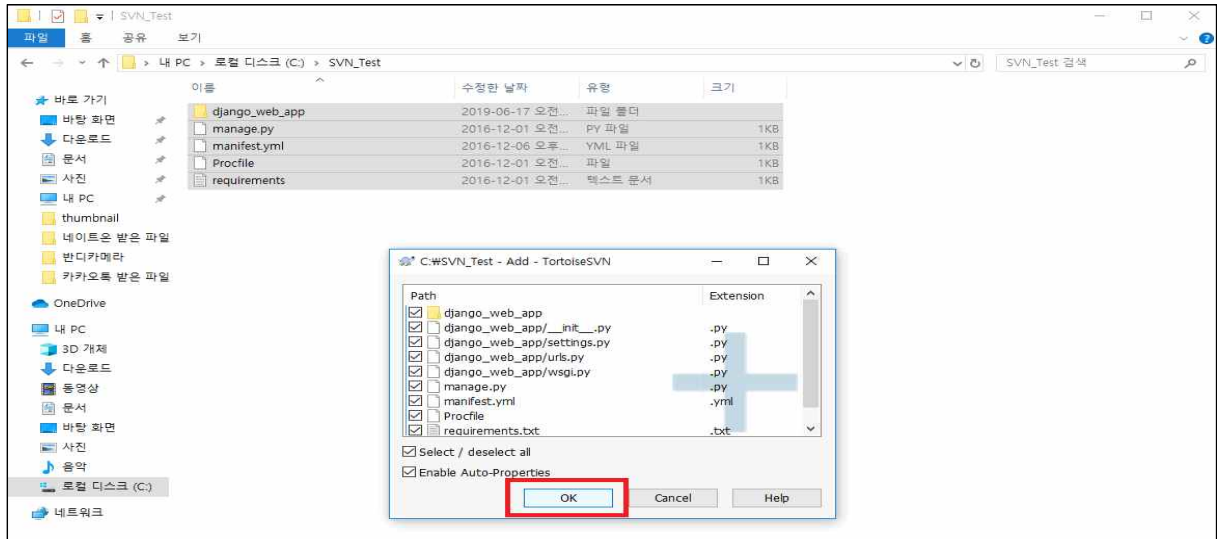


Figure 39. Source Control Management - SVN Add Using TortoiseSVN (2)

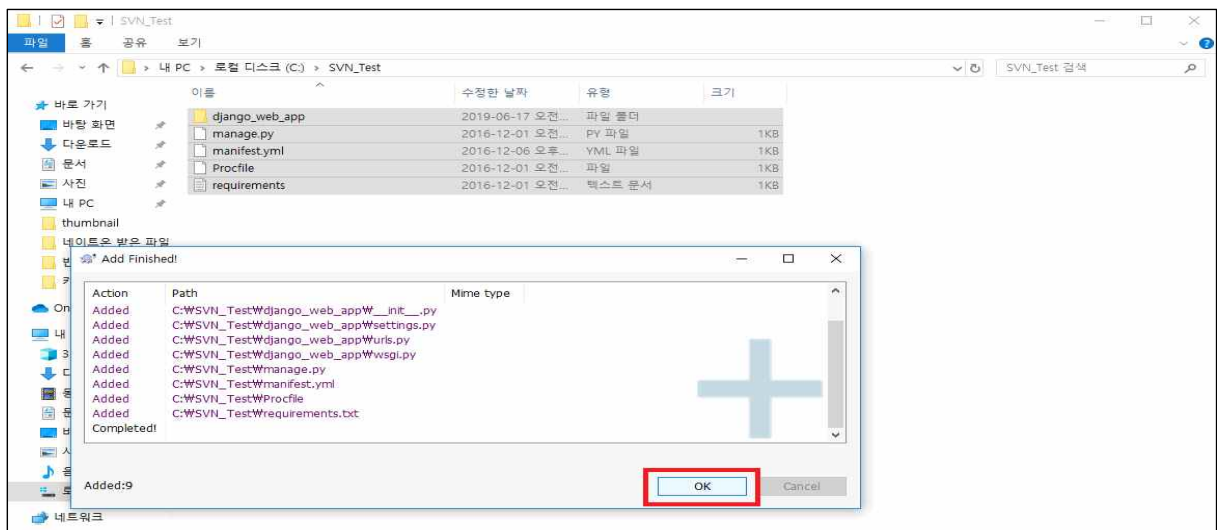


Figure 40. Source Control Management - SVN Add Using TortoiseSVN (3)

Figure 39, 40과 같이 레파지토리에 추가할 파일 목록을 체크한 후 OK 버튼을 누르면 Add 명령이 수행되며 위와 같은 결과를 확인할 수 있다.



Figure 41. Source Control Management - SVN Commit Using TortoiseSVN (1)

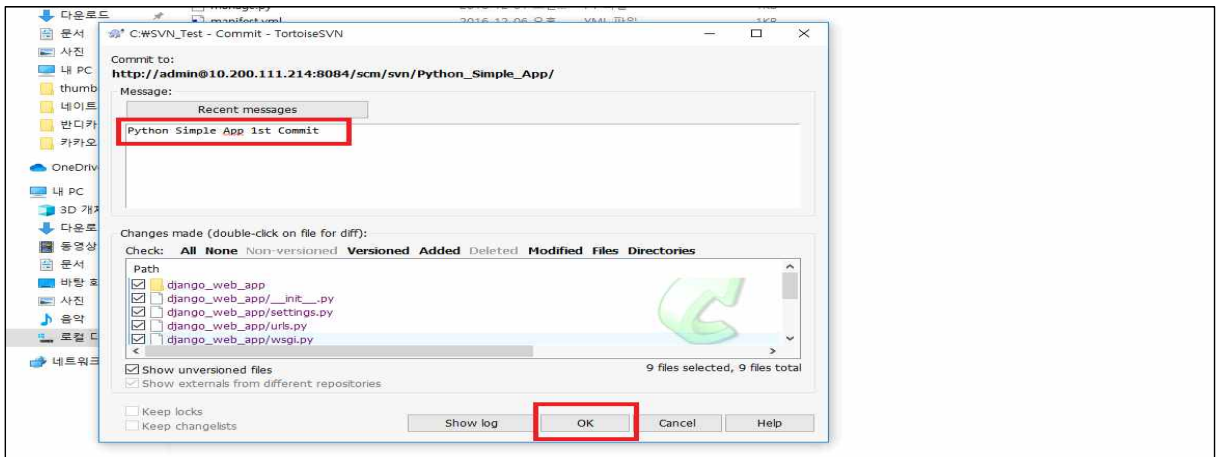


Figure 42. Source Control Management - SVN Commit Using TortoiseSVN (2)

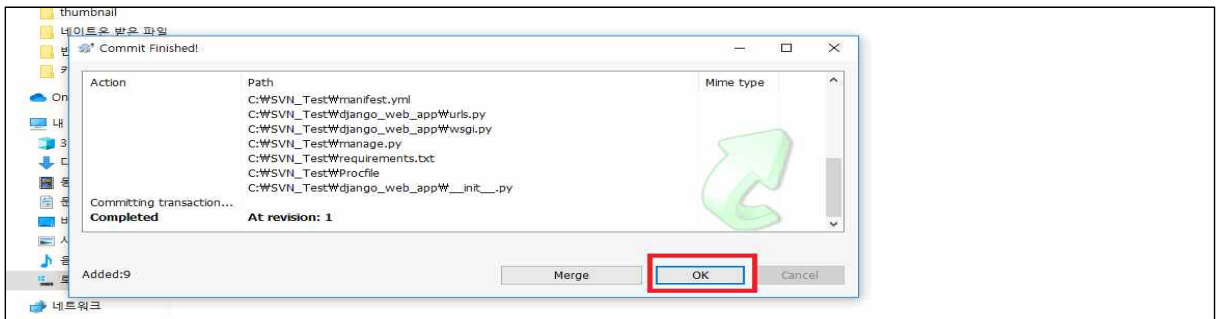


Figure 43. Source Control Management - SVN Commit Using TortoiseSVN (3)

Add가 완료되고 나면 Figure 41, 42, 43과 같이 SVN Commit을 순차적으로 수행한다. Figure 42에서처럼 SVN은 Commit시 로그를 남길 수가 있는데, 본 문서에서는 “Python Simple App 1st Commit” 이라고 입력하였다.



Figure 44. Source Control Management – Updated Repository Content – Files

로컬 컴퓨터에서 Commit이 완료된 후 Figure 44와 같이 다시 형상관리 Web-UI로 이동하면 새로이 등록된 File의 목록을 확인할 수 있다.

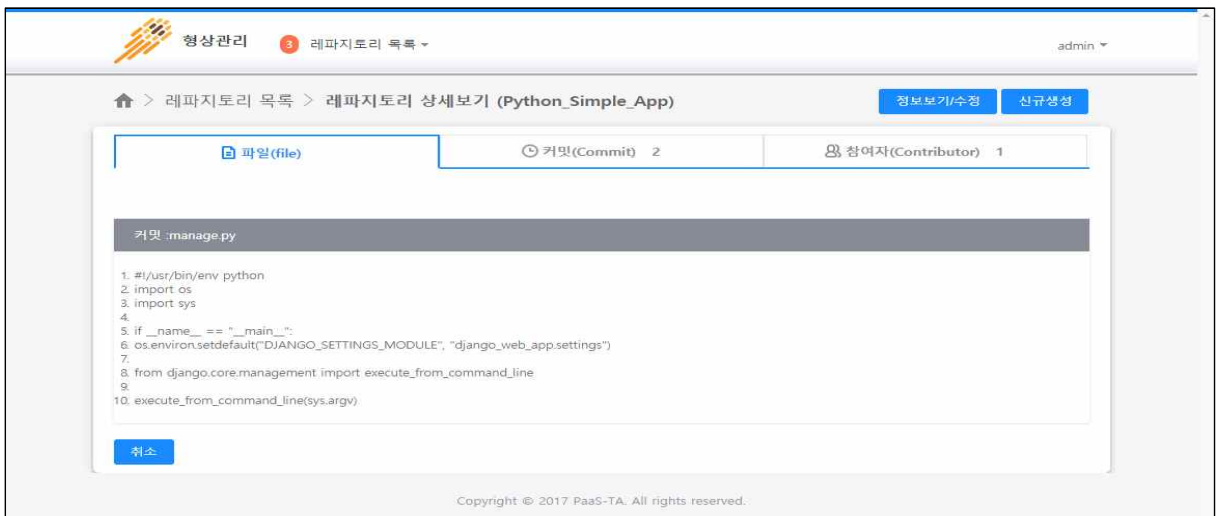


Figure 45. Source Control Management – Updated Repository Content – Source Code

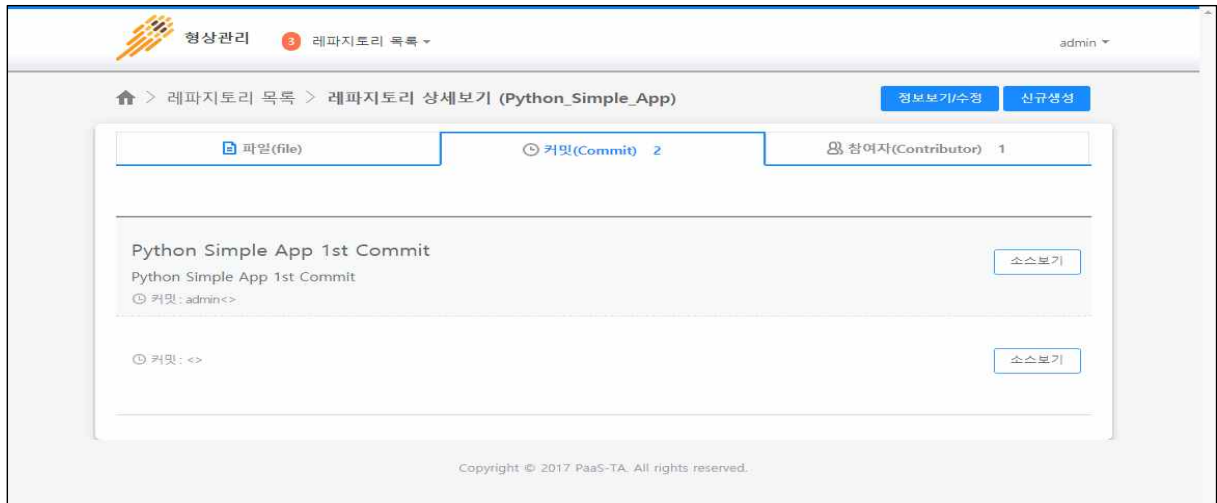


Figure 46. Source Control Management – Updated Repository Content – Commit Log

또한 Figure 45, 46과 같이 소스코드의 상세 내용 및 Commit 시 입력한 로그 역시도 형상 관리 Web-UI에서 확인할 수 있다.

3. APCC 계절예측 및 기후자료 처리서비스 플랫폼 구축

가. 플랫폼 기반 온라인 다중모델 앙상블 계절예측/검증시스템 개발

(1) 시스템 개발 필요성

APCC는 온라인 기후정보분석도구(clik.apcc21.org)에서 사용자 맞춤형 서비스를 제공하고 있다. 예측정보 활용성 증대 및 효율적인 현업 운영을 위해 MME 계절예측 통합시스템이 신규 구축되어 2020년 6월부터 운영되고 있으나 온라인 기후정보분석도구는 이러한 신규 현업시스템 자료를 반영하지 못하고 있다.

APCC 예측시스템 및 기후서비스 환경의 변화에 따라 온라인 기후정보분석도구를 대체할 수 있는 플랫폼 기반의 다중모델앙상블 계절예측 및 검증 시스템을 구축하여 신규 현업 시스템을 적용하는 동시에 플랫폼으로 통합할 수 있다. 또한 기존에 제공하던 검증정보에 계절예측에서 주로 활용되고 있는 추가의 검증기법을 추가함으로써 기후정보서비스 활용성 향상을 도모하고자 한다.

(2) 시스템 설계

(가) 요구사항

플랫폼 기반 온라인 다중모델앙상블 계절예측 및 검증 시스템에 대한 사용자 및 기능에 대한 요구사항을 정의하였다. 해당 요구사항은 각각을 구분할 수 있는 ID와 그에 대한 정의로 이루어져 있다.

Table 9. Requirements of MME prediction and verification.

요구사항 ID	요구사항 정의
RQ_PV_PV_001	사용자는 APCC 기후자료서비스 플랫폼에서 다중모델앙상블(MME, Multi-Model Ensemble) 자료를 요청할 수 있다.
RQ_PV_PV_002	사용자는 년도, 월(시즌), MME 방식, 변수, 참여 모델 등을 확인하고 선택할 수 있다.
RQ_PV_PV_003	계절예측 시스템은 년도, 시즌에 맞게 참여모델이 출력되어 사용자가 정확히 선택할 수 있도록 해야 한다.
RQ_PV_PV_004	계절예측 시스템은 결과를 그림으로 제공하여 사용자가 다운로드 받을 수 있도록 해야 한다.
RQ_PV_PV_005	사용자는 계절예측 결과 그림을 GUI에서 확인할 수 있다.

RQ_PF_PV_006	검증 시스템은 Hindcast 자료에 대한 SR(Success Rate), HSS(Heidke Skill Score), ACC(Anomaly Correlation Coefficient) 등의 검증 기법을 제공해야 한다.
RQ_PF_PV_007	검증 시스템은 검증 기법에 따라 그래프 혹은 그리드 형태의 그림 자료를 제공할 수 있다.
RQ_PF_PV_008	그리드 형태의 검증 결과는 계절예측 결과와 마찬가지로 GUI에서 확인할 수 있어야 한다.
RQ_PF_PV_09	계절예측 및 검증 시스템은 원활한 자료 제공을 위해 필요한 경우 전처리를 수행하여 자료를 미리 생산할 수 있다.
RQ_PF_PV_010	시스템이 처리하지 않은 자료를 사용자가 제공할 경우 Job 의 형태로 요청을 처리해야 한다.
RQ_PF_PV_011	사용자는 입력한 Job 리스트를 확인하고 진행상황을 확인할 수 있어야 한다.
RQ_PF_PV_012	사용자는 계절예측 및 검증 결과를 Netcdf 파일 형태로 다운로드 할 수 있다.
RQ_PF_PV_013	사용자는 계절예측 및 검증 결과 그림을 png 파일 형태로 다운로드 할 수 있다.

(나) 요구사항 분석

① Use case



Figure 47. Use case diagram

계절예측 및 검증 시스템에 대한 기능 이용사례는 Figure 47에 잘 나타나 있다. 사용자는 기후자료서비스에서 계절예측 및 검증 자료를 조회하고 다운로드할 수 있다. 또한 자신이 사용하는 컴퓨팅 시스템(PC, 서버 등)에서 Open API를 이용하여 자료를 요청할 수 있다.

② 계절예측 및 검증 요청

사용자는 자료 서비스 홈페이지에서 예측 및 검증 서비스를 요청한다. 자료 서비스가 해당 웹 페이지를 출력하면 사용자는 원하는 조건을 선택하여 자료를 요청할 수 있다. 예측 및 검증 서비스는 사용자가 요청한 조건에 해당하는 자료를 조회하여 이미 생산된 자료가 있으면 즉시 plot 이미지를 출력한다. 사용자는 이미지를 확인하고 이미지 및 원본 netcdf 자료를 다운로드 할 수 있다.

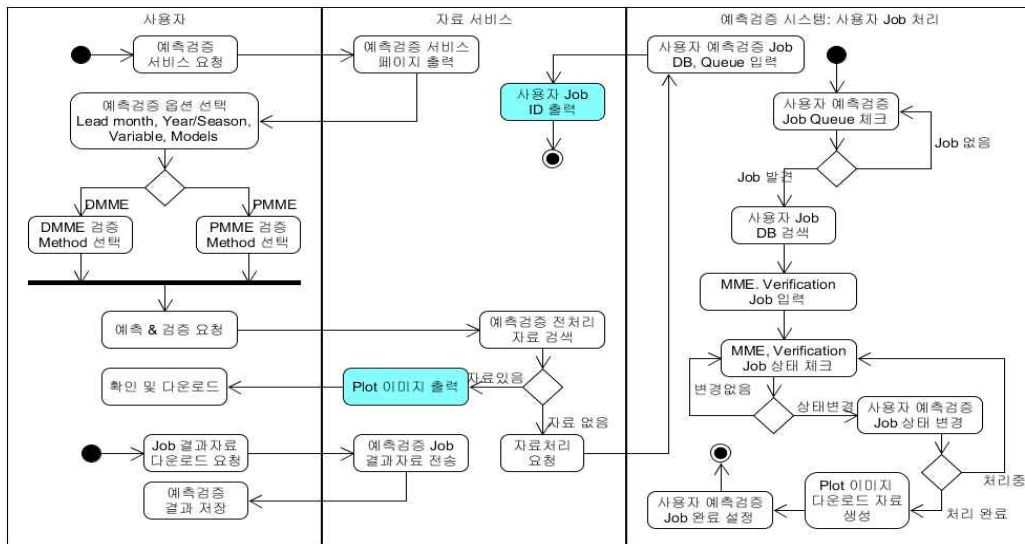


Figure 48. Activity diagram to request prediction and verification data

생산되어 있는 자료가 없을 경우 예측 및 검증 서비스는 job을 생성하여 처리를 요청한다. job을 처리하여 자료를 생산하는 모듈은 새로운 job이 입력되면 조건에 맞는 자료를 생산한다. job 처리가 완료되면 job의 처리 상태를 사용자가 다운로드할 수 있는 상태로 변경한다.

③ 계절예측 자료 생산

계절예측 자료 생산 모듈은 job queue를 사용하여 신규 job을 확인한다. 입력된 job의 조건에 따라 계절예측 자료 생성 코드를 호출하고 결과 파일을 생산한다. 마지막으로 job 처리 결과를 업데이트한다.

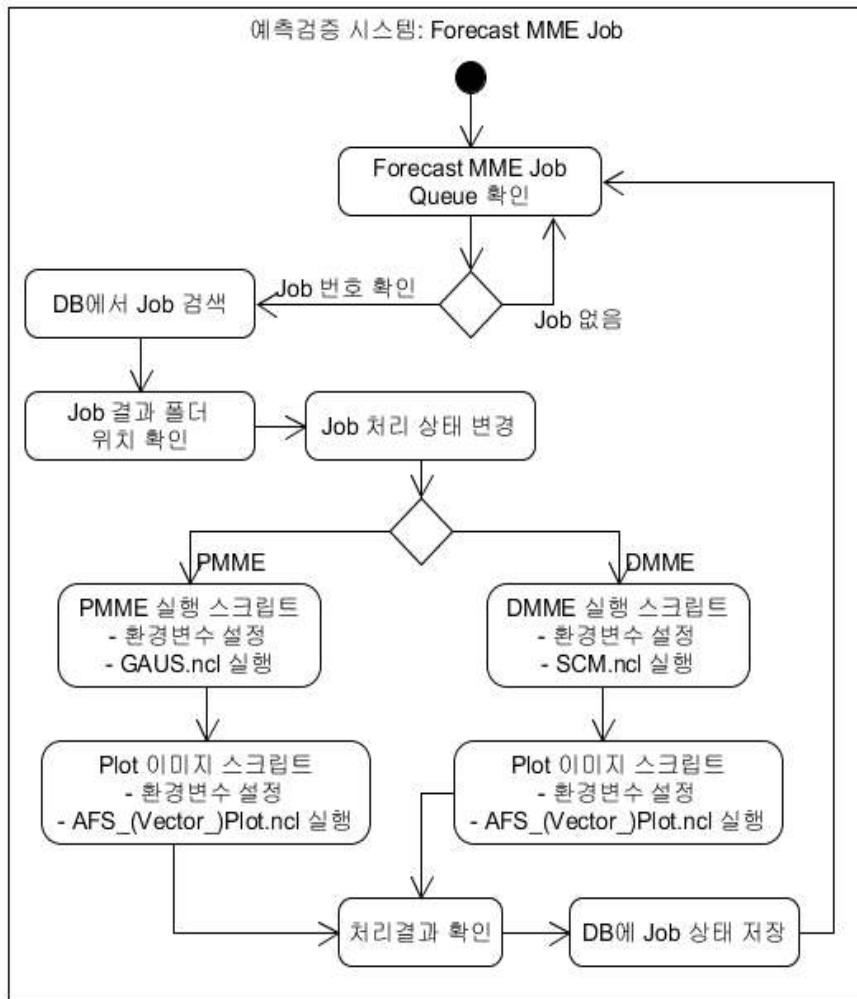


Figure 49. Activity diagram to produce forecast MME

④ 검증 자료 생산

검증 자료 생산 모듈은 계절예측 생산 모듈과 마찬가지로 job queue를 사용한다. 입력된 job의 조건에 따라 검증 자료 생성 코드를 호출하고 결과 파일을 생산한다. 마지막으로 job 처리 결과를 업데이트한다.

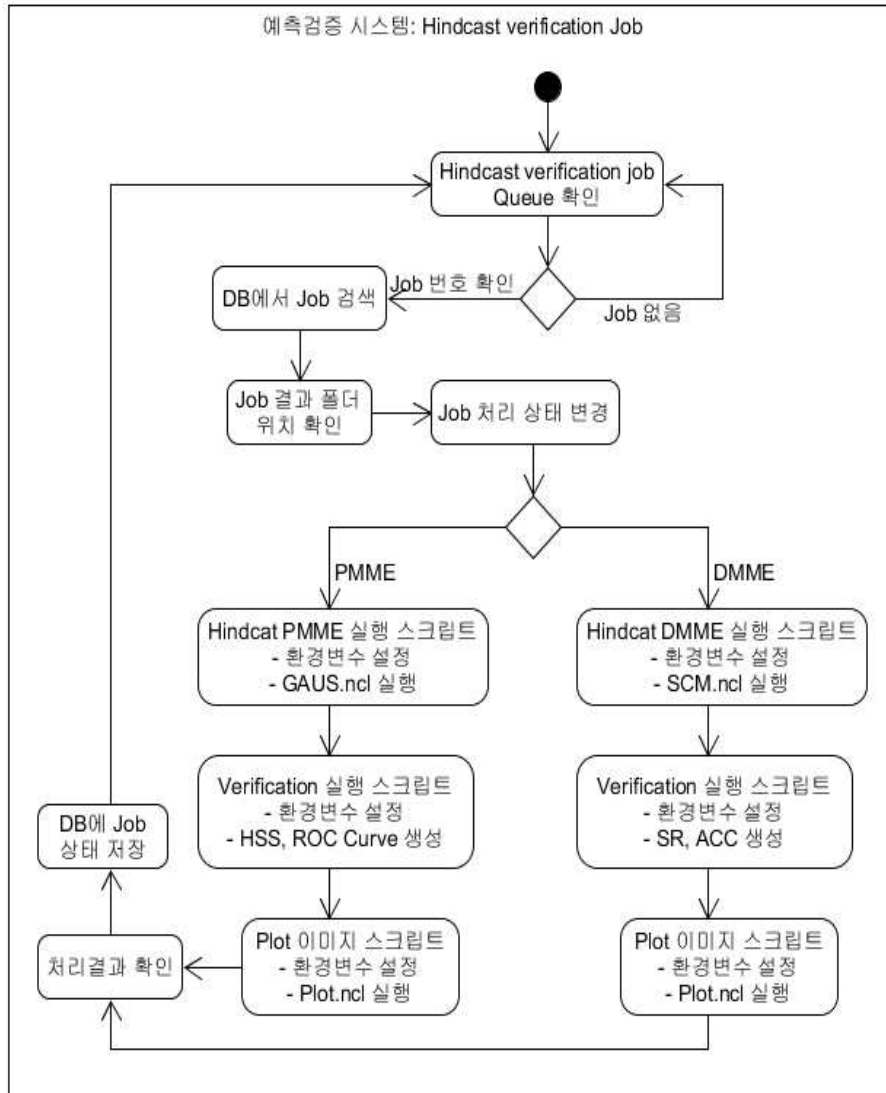


Figure 50. Activity diagram to produce hindcast verification

(3) 계절예측 및 검증 시스템 구성 및 운영 환경

계절예측 및 검증 시스템은 플랫폼에 배포되어 운영된다. APCC에서 운영 중인 플랫폼은 한국정보화진흥원(NIA)에서 개발한 개방형 클라우드 플랫폼 파스-타(PaaS-TA)를 기반으로 해서 구축되었다. Figure 51에서 보는 바와 같이 파스-타는 CF (Cloud Foundry) 기반의 PaaS (Platform-as-a-Service)와 Kubernetes 기반의 CaaS (Containers-as-a-Service)로 구성이 되어 있다. PaaS 영역에는 계절예측 및 검증 시스템의 웹 인터페이스와 Open API가 동작하고 있고, 실제 계산 작업을 하는 Docker 기반의 검증 및 계절예측 컨테이너는 CaaS 영역에서 동작하고 있다. 그리고 데이터를 관리하는 데이터베이스와 데이터를 저장하는 스토리지는 CaaS와 PaaS 영역에서 공유할 수 있도록 구성이 되어 있다.

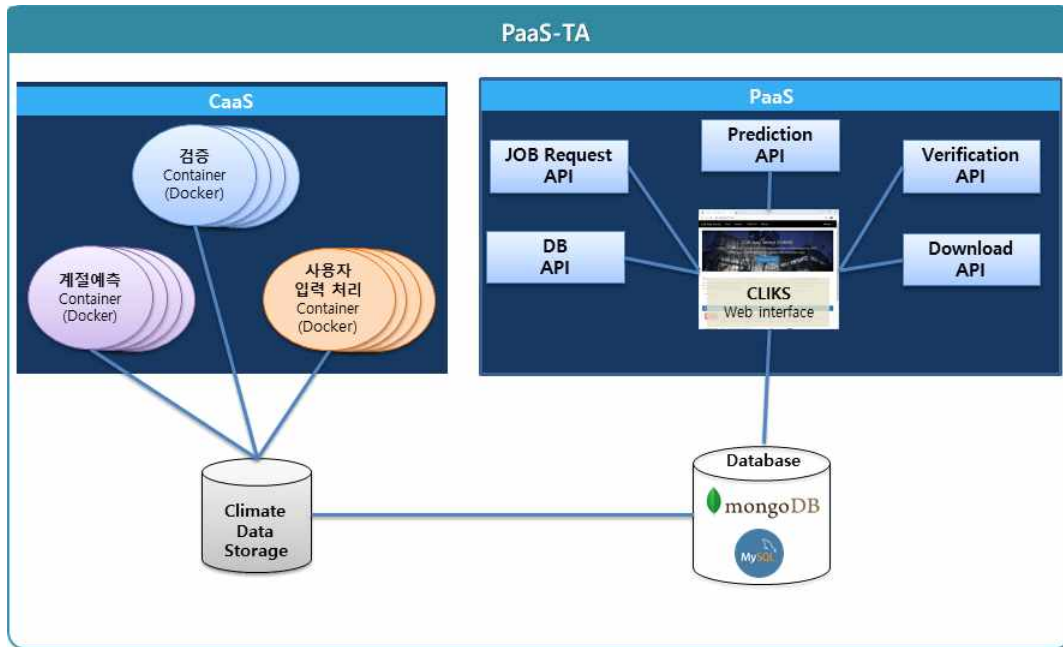


Figure 51. System architecture of Prediction and Verification

(4) 현업 시스템 모듈화

계절예측 및 검증 시스템은 APCC의 현업 시스템과 기존의 CLIK 기능을 기반으로 하고 있다. 현업 시스템 및 CLIK 기능을 분석하여 플랫폼 운영환경에 따라 이식할 수 있도록 모듈화를 수행하였다.

① 계절예측 현업 시스템 모듈화

계절예측 현업 시스템은 NCL을 기반으로 개발된 시스템이다. 사용자가 직접 입력 값을 설정하여 실행하도록 구성이 되어 있다. 이것은 자동화가 아닌 현업 담당자가 직접 실행하여 현업 결과를 생성하는 구조이다. 그러나 시스템 모듈화를 위해서는 사용자의 개입이 최소화되어야 하고 웹 인터페이스를 통해 사용자가 입력한 값을 통해서 시스템이 자동으로 실행하도록 구성이 되어야 한다.

우선 계절예측 현업 시스템에 대한 분석을 실시하였다. Figure 52은 현재 현업에서 운영되고 있는 방식을 기반으로 하여 분석한 결과이다. 이에 각 기법별로 상세 분석을 실시하였다.

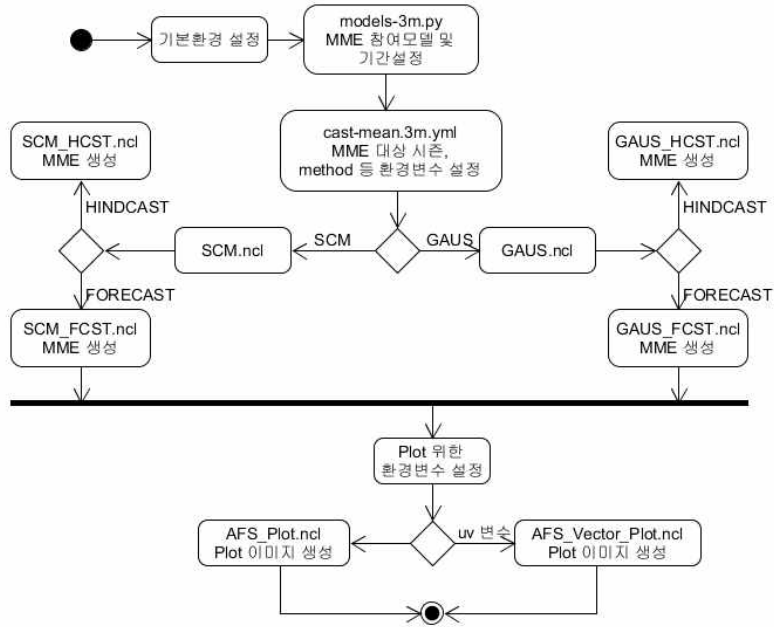


Figure 52. Activity diagram to produce

Figure 53은 SCM (Deterministic) 기법의 NCL 코드를 분석한 결과이고, Figure 54은 GAUS (Probabilistic) 기법의 NCL 코드를 분석한 결과이다.

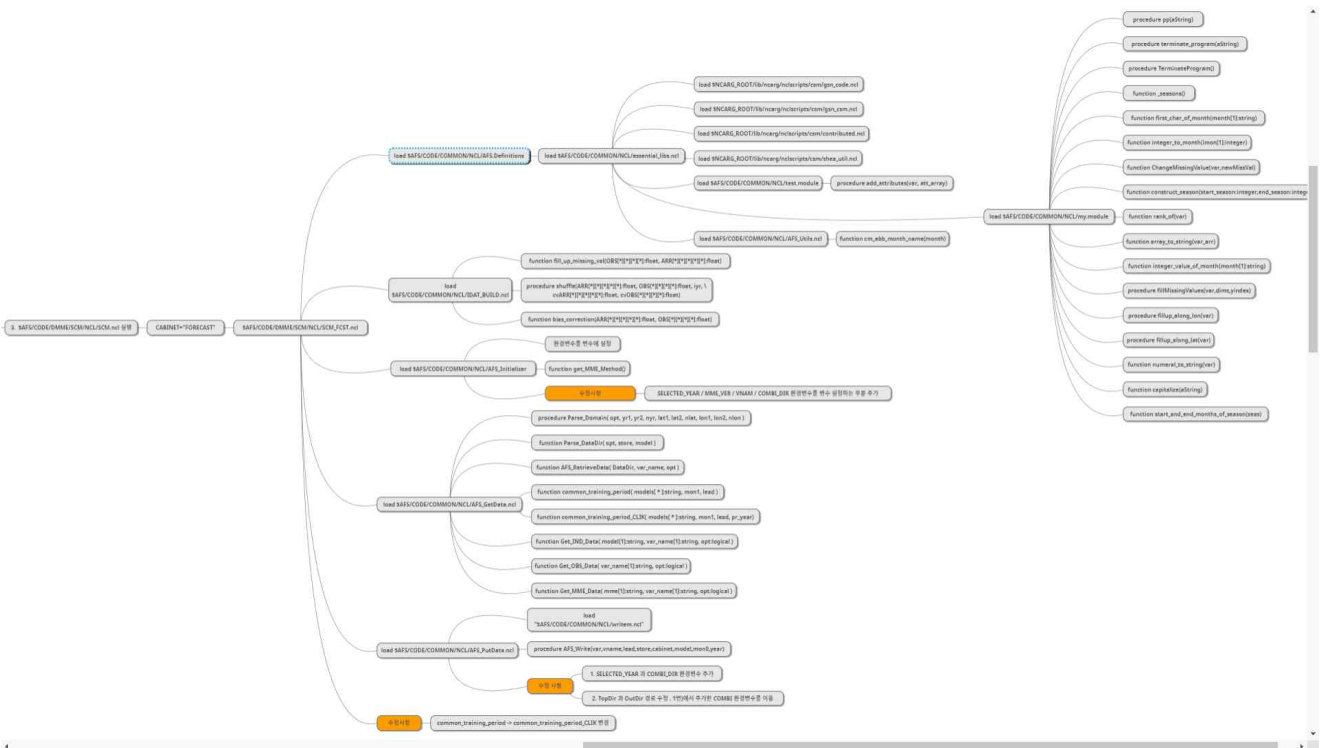


Figure 53. NCL code analysis of SCM using MindMap

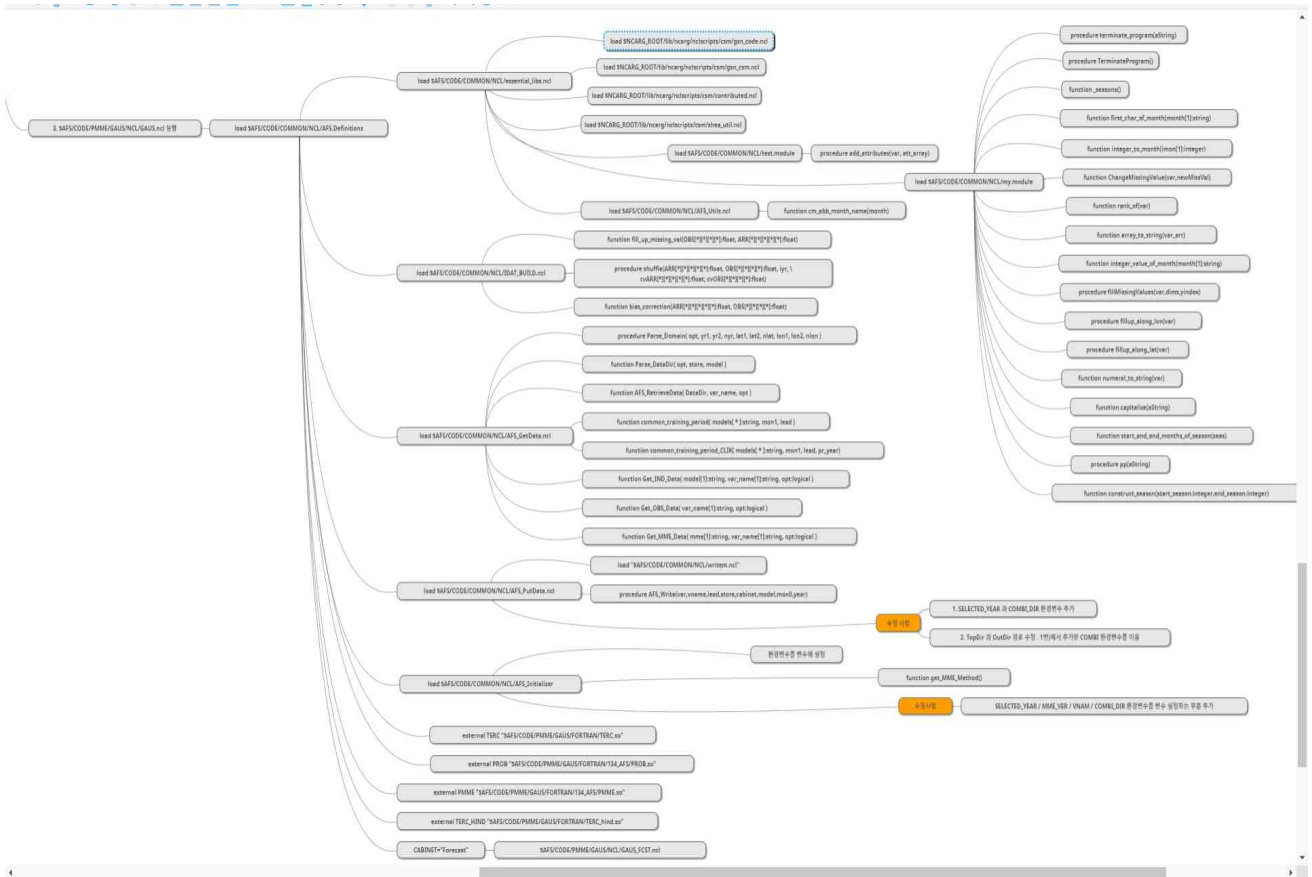


Figure 54. NCL code analysis of GAUS using MindMap

위의 분석한 결과를 토대로 NCL 코드의 입출력 부분을 계절예측 및 검증시스템에 맞게 재구성을 실시하였다. 입력부분은 사용자가 입력한 조건들을 환경변수로 설정하여 현업코드가 실행되도록 개선하였고, 결과 폴더는 데이터베이스와 연동하여 관리하기 위해 폴더 구조 설계를 진행하였다. Table 10에서 보는 바와 같이 기존의 환경 변수 설정 부분에서 사용자 입력을 반영하기 위한 환경 변수들을 추가하였다. 해당 환경변수들은 사용자의 입력 값을 받아들여 계절예측 코드를 구동할 때 기존에 직접 입력하였던 부분을 대체하게 된다.

Table 10. Add environment variables for web interface

<code>_FillValue</code>	<code>= AFS_FillValue()</code>
<code>_StartMon</code>	<code>= integer_value_of_month(getenv("START_MON"))</code>
<code>_LeadTime</code>	<code>= stringtoint(getenv("LEAD_TIME"))</code>
<code>_EndMon</code>	<code>= _StartMon+_LeadTime-1</code>
<code>_ForecastYear</code>	<code>= stringtoint(getenv("START_YEAR"))</code>
<code>_HindcastYear</code>	<code>= _ForecastYear</code>
<code>_SelectedYear</code>	<code>= stringtoint(getenv("SELECTED_YEAR"))</code>

_HindcastMon	= _StartMon
_ForecastMon	= _StartMon
_PlotYear	= _ForecastYear
_Store	= getenv("STORE")
_Cabinet	= getenv("CABINET")
_AFSDataType	= _Cabinet
_Model	= getenv("MODEL")
_MME	= _Model
_MMEClass	= getenv("MME_CLASS")
_PMMECat	= stringtoint(getenv("PMME_CAT"))
_VName	= getenv("VNAME")
_PMedia	= getenv("PLOT_MEDIA")
_MMEModels	= getenv("MME_MODELS")
_InstituteName	= getenv("INSTITUTE")
_IndModelName	= getenv("IND_MODEL")
_ModelName	= getenv("SUPP_MODEL")
_SeaLand	= getenv("SEA_LAND")
_Score	= getenv("SCORE")
_AFS_Temp	= getenv("TMP_DIR")
_Version	= getenv("MME_VER")
_VName	= getenv("VNAME")
_HindcastYears	= getenv("HIND_YEARS")
_CombiDir	= getenv("COMBI_DIR")
_SeasonName	= getenv("SEASON")
_Issue	= getenv("ISSUE")

결과 폴더에 대해서도 현업 코드를 기반으로 구조를 변경하였다. 우선 설계를 진행하였는데, Figure 55는 설계 진행 결과이다. 해당 결과 폴더는 데이터베이스와 연동이 되어 있어 사용자끼리 결과를 공유도 할 수 있도록 구성이 되어 있다.

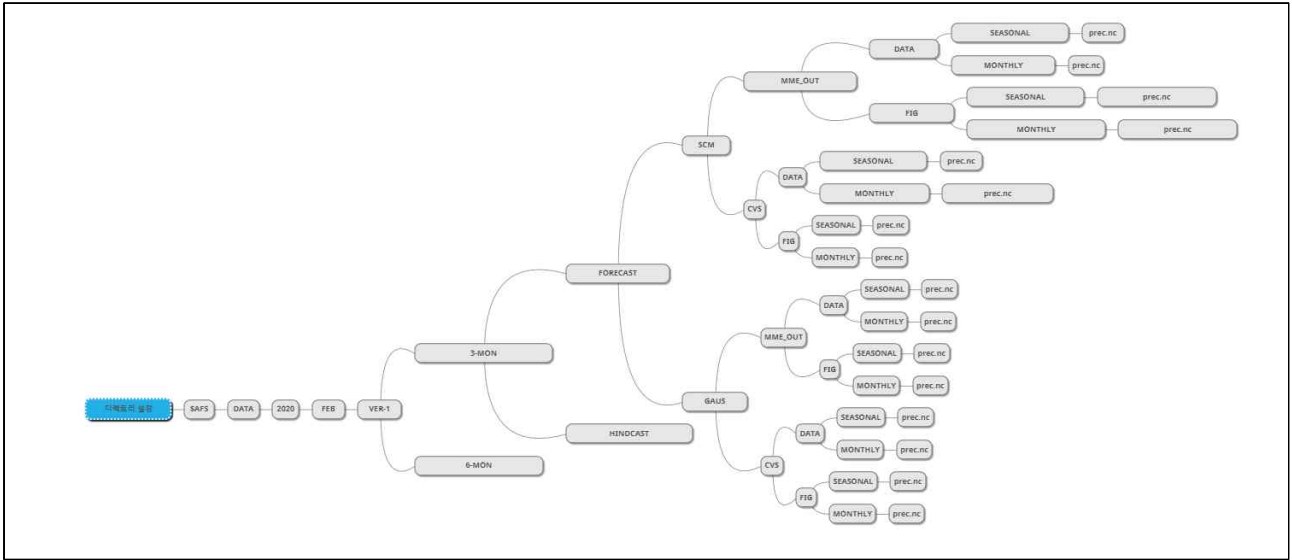


Figure 55. Output folder structure

설계된 결과 폴더를 기반으로 해서 기존의 현업 코드를 수정하였다. 출력과 관련된 수정은 Figure 56에서 보는바와 같이 NCL 코드를 수정하여 출력 폴더를 수정하였다.

Figure 56. Code differences of CLIKS ncl code vs AFS ncl code

② 검증 현업 시스템 모듈화

구축한 검증 서비스는 Hindcast verification을 제공하도록 설계되었다. 현업 시스템에서 제공하는 검증 기법과 기존 CLIK의 SR 기법을 제공한다(Table 11). 추후 개선을 통하여 검증 기법을 추가하거나 Forecast verification 서비스도 제공할 수 있다.

Table 11. Verification skills for hindcast

Skills	MME Class	Long name
SR	Deterministic	Success Rate
ACC	Deterministic	Anomaly Correlation Coefficient
HSS	Probabilistic	Heidke Skill Score
ROC Curve	Probabilistic	Relative Operating Characteristics Curve

현업 검증 시스템인 CVS (Climate Verification System)는 Figure 57과 같이 MME 종류, Lead time, 검증 기법에 따라 코드(Python) 파일이 나뉘어져 있다. 각 코드는 기본적으로 Figure 58와 같은 구조로 구성되어 있다.

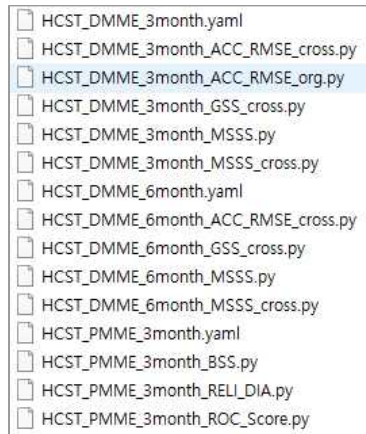


Figure 57. CVS codes

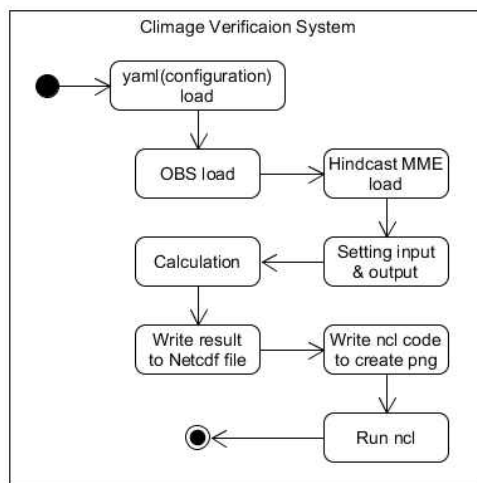


Figure 58. Activity diagram of CVS

현업 시스템을 플랫폼에 이식할 때 가장 중요한 부분은 configuration을 포함한 초기 코드 실행과 입출력 자료 설정이다. 초기 코드 실행을 위해 추가로 작성한 코드는 다음과 같다.

```
hindcast_month = os.environ['START_MON']
hindcast_year = os.environ['SELECTED_YEAR']
ncl_dir = os.environ['COMBI_DIR'] + "/RUN/"
run_dir = os.environ['CVS_DIR'] + "/"
lead_time = os.environ['LEAD_TIME']
var_name = os.environ['VNAM']
skill = os.environ['SKILL']
jobid = os.environ['JOBID']

CONF = """
DataList:
  - Hindcast_month : {0}
    Current_year   : {1}
    Variable_name  : {2}
"""
.format(hindcast_month, hindcast_year, var_name)
if lead_time == "3":
    with open(ncl_dir + "HCST_PMME_3month_{0}.yaml".format(jobid), "w") as file:
        file.writelines(CONF)
    run_command_clik.run_python(run_dir + "HCST_PMME_3month_{0}.py".format(skill))
else:
    with open(ncl_dir + "HCST_PMME_6month_{0}.yaml".format(jobid), "w") as file:
        file.writelines(CONF)
    run_command_clik.run_python(run_dir + "HCST_PMME_6month_{0}.py".format(skill))
```

Figure 59. Python code to start running

Figure 59의 코드에서 시스템 환경설정 값을 받아와서 현업 검증코드에 입력할 yaml 파일을 작성한다. yaml 파일을 읽어서 실제 검증 자료를 생산하는 현업 코드의 주요 수정내용은 다음과 같다.

```
# model combination folder
combi_dir = os.environ['COMBI_DIR']

# folder for output netcdf and png
```

```

nc_dir = combi_dir + "/CVS/DATA/ "
fig_dir = combi_dir + "/CVS/FIG/"

# folder for ncl code
ncl_dir = combi_dir + "/RUN/ "

# folder for observation data
obs_path = "/data02/clik/CVS/OBS/ "

# folder hindcast mme data
fcst_mme_path = combi_dir + "/MME_OUT/DATA/ "

# load yaml (configuration) file
data = yaml.load(open(ncl_dir + "HCST_PMME_3month_{0}.yaml".format(jobid),'r'))

# set output path
ncout_season_path = nc_dir + "HSS/SEASONAL/"
nclout_season_path = ncl_dir + "HSS/SEASONAL/"
figout_season_path = fig_dir + "HSS/SEASONAL/ "

# run ncl
run_command_clik.run_ncl(ncl_fname.format(nclout_month_path))

```

Figure 60. Modifications of CVS codes (HSS)

기존 CLIK (clik.apcc21.org)에서 제공하는 검증 기법 중 SR (Success rate)을 플랫폼으로 이식하였다. CLIK의 SR 생성 과정은 다음과 같다.

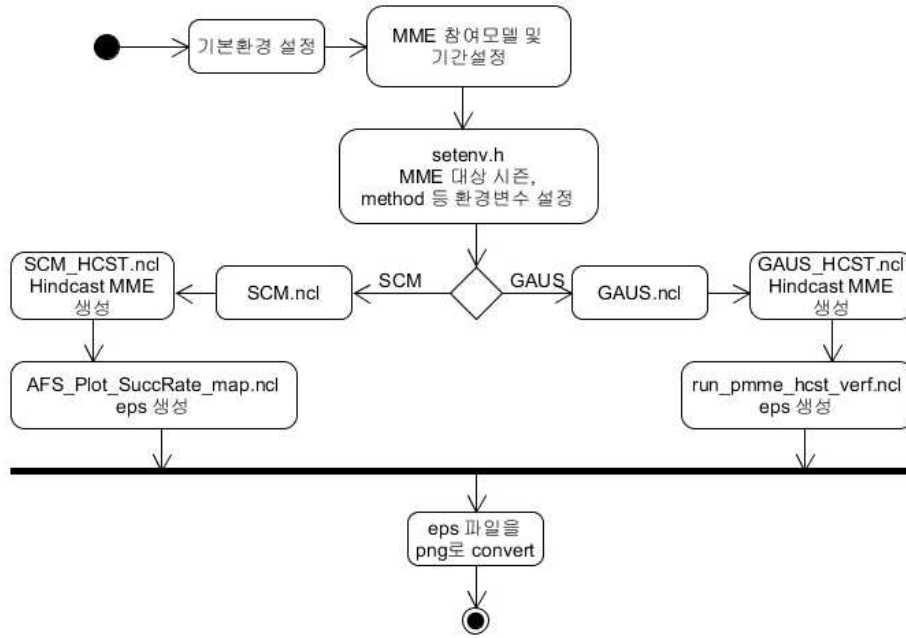


Figure 61. Activity diagram to produce SR

기존 CLIK의 검증 실행 결과는 plot image에 지도 및 추가 정보의 decoration이 포함되지 않아 플랫폼에 이식하기 위해서 다음과 같은 추가의 수정이 필요하였다.

```

title_season = construct_season(_SeasStart+_StartMon, _SeasEnd+_StartMon)
title = "Success Rate : " + str_upper(VName) + ", " + title_season + " (" + hind_start + "-" +
hind_end + ")"
p_type = plttyp

wks = gsn_open_wks(p_type, pltfil)

res          = True
res@gsnDraw  = False
res@gsnFrame = False

res@tmXTOOn = False
res@tmYROOn = False

res@cnFillOn          = True
res@cnLinesOn         = False
res@cnLineLabelsOn   = False

res@mpOutlineOn          = True
res@mpOutlineBoundarySets = "Geographical"
res@mpOutlineBoundarySets = "National"
res@mpGeophysicalLineColor = "Black"

```

```

res@mpGeophysicalLineThicknessF = 1.2
res@mpNationalLineColor          = "Black"
res@mpNationalLineThicknessF    = 1.2
res@mpFillOn                     = False
res@mpCenterLonF                 = 180

res@pmLabelBarDisplayMode = "Always"           ; turns on label bar
res@pmLabelBarSide         = "Bottom"         ; default is right
res@pmLabelBarWidthF      = 0.6              ; default is shorter
res@pmLabelBarHeightF     = 0.1              ; default is taller

res@mpMinLatF = -90
res@mpMaxLatF = 90

res@lbAutoManage          = False             ; we control label bar
res@lbOrientation         = "Horizontal"     ; ncl default is vertical
res@lbTitleString = " "
res@lbTitleFontHeightF = 0.001
res@lbLabelFontHeightF   = .015             ; default is HUGE
res@lbPerimOn            = False             ; default has box

res@gsnLeftString = " "

res@cnLevelSelectionMode = "ExplicitLevels"
res@cnLevels = (/0.11,0.22,0.33,0.44,0.55,0.66,0.77,0.88,0.99/)

colors = (/ (255,255,255/), (/0,0,0/), \
(255,255,255/), \
(255,255,255/), \
(255,255,255/), \
(/187,248,255/), \
(/117,210,255/), \
(/060,134,255/), \
(/255,213,153/), \
(/255,120,085/), \
(/255,044,000/), \
(/215,020,046/) /) *1.0
colors = colors/255.           ; normalize (required by NCL)

gsn_define_colormap(wks, colors) ; generate new color map

res@lbLabelBarOn          = False
res@cnSmoothingOn        = True
res@cnFillOn              = True
res@cnLinesOn            = False
res@cnLineLabelsOn       = False           ; turn off line labels
res@cnLineColor          = "black"
res@cnInfoLabelOn        = False
res@gsnContourNegLineDashPattern = 1 ;
res@gsnSpreadColors       = True

```

```

; PLOT
plot = gsn_csm_contour_map(wks,scr,res)

; GET INFO 4 COLOR BAR
getvalues plot
    "vpXF"      : vpx
    "vpYF"      : vpy
    "vpHeightF" : vph
    "vpWidthF"  : vpw
end getvalues

pcolors0 = gsn_retrieve_colormap(wks)
pcolors  = pcolors0(2,:);
ncolors  = dimsizes(pcolors)
nboxes   = ncolors(0)

getvalues plot@contour
    "cnLevels" : levels
end getvalues

num_colors = nboxes
nstp       = 1
labels     = levels
lbres      = True
lbcols=ispan(1,num_colors-0,nstp)

lab2              = labels * 1.0
neg_labs          = labels.le.0.33
lab2(ind(neg_labs)) = labels(ind(neg_labs))
zer_labs          = labels.gt.0.33.and. labels.le.0.66
lab2(ind(zer_labs)) = labels(ind(zer_labs))
pos_labs          = labels.gt.0.66
lab2(ind(pos_labs)) = labels(ind(pos_labs))

lbres@lbMonoFillPattern      = True
lbres@lbOrientation="horizontal"
lbres@lbAutoManage           = False      ; we control label bar
lbres@lbLabelOffsetF = 0.05
lbres@lbBoxMinorExtentF = 0.2
lbres@lbBoxMajorExtentF = 0.9
lbres@lbTitleOn = True
lbres@lbTitleJust = "TopCenter"
lbres@lbTitleOffsetF = -0.55
lbres@lbTitleFontHeightF = 0.013
lbres@lbPerimOn              = False      ; Turn off labelbar perimeter.

; BELOW NORMAL LABEL
lblabels=lab2(ind(neg_labs))
snboxes=dimsizes(lblabels)
fill_colors=lbcols(ind(neg_labs)) + 1
lbres@vpXF = 0.22

```

```

lbres@vpYF = 0.24
lbres@vpWidthF = 0.18
lbres@vpHeightF = 0.15
lbres@lbLabelFontHeightF = 0.01
lbres@lbLabelJust = "CenterLeft"
lbres@lbFillColor = fill_colors ; use nice strided colors
lbres@lbTitleString = "Poor"
gsn_labelbar_ndc(wks,snboxes,lblabels,lbres@vpXF,lbres@vpYF,lbres)

; NEAR NORMAL LABEL
delete(lblabels)
delete(fill_colors)
delete(lbres@lbFillColor)
lblabels=lab2(ind(zer_labs))
snboxes=dimsizes(lblabels)
fill_colors=lbcols(ind(zer_labs)) + 1
lbres@vpXF = 0.40
lbres@vpYF = 0.24
lbres@vpWidthF = 0.18
lbres@vpHeightF = 0.15
lbres@lbLabelFontHeightF = 0.01
lbres@lbLabelJust = "CenterLeft"
lbres@lbFillColor = fill_colors ; use nice strided colors
lbres@lbTitleString = "Reasonable"
gsn_labelbar_ndc(wks,snboxes,lblabels,lbres@vpXF,lbres@vpYF,lbres)

; ABOVE NORMAL LABEL
delete(lblabels)
delete(fill_colors)
delete(lbres@lbFillColor)
lblabels=lab2(ind(pos_labs))
snboxes=dimsizes(lblabels)
fill_colors=lbcols(ind(pos_labs)) + 1
lbres@vpXF = 0.58
lbres@vpYF = 0.24
lbres@vpWidthF = 0.18
lbres@vpHeightF = 0.15
lbres@lbLabelFontHeightF = 0.01
lbres@lbLabelJust = "CenterLeft"
lbres@lbFillColor = fill_colors ; use nice strided colors
lbres@lbTitleString = "High"
gsn_labelbar_ndc(wks,snboxes,lblabels,lbres@vpXF,lbres@vpYF,lbres)
draw(plot)

; TITLE
Tires = True
Tires@txFontHeightF = .020
Tires@txFont = "times-roman"
gsn_text_ndc(wks, title, 0.51, 0.75, Tires)

; POLYLINE SETTING

```

```

resOutBox      = True
resOutLine     = True
sres           = True
txres          = True
dxres          = True
intxres        = True
amres          = True

; TEXT CONTENTS
bottomsimbol   = "c"
bottomtext     = "APEC Climate Center"
issuedate      = systemfunc("date +%d")
issuemon       = systemfunc("date +%b")
issueyear      = systemfunc("date +%Y")
creatdate      = "Issued: " + issuedate + " " + issuemon + ", " + issueyear

; POLYLINE SETTING
resOutBox@gsLineColor      = "Black"
resOutBox@gsLineThicknessF = 1.0
resOutLine@gsLineColor     = "Black"
resOutLine@gsLineThicknessF = 1.0

; TEXT SETTING
txres@txFont      = "times-roman"
txres@txFontHeightF = 0.0135
dxres@txFont      = "times-roman"
dxres@txFontHeightF = 0.013
sres@txFont       = "text-symbols"
sres@txFontHeightF = 0.012

; POINTS LOCATION FOR POLYLINE
bxpts = (/0.02, 0.98,0.98,0.02,0.02/)
bypts = (/0.10, 0.10,0.80,0.80,0.10/)
lxpts = (/0.04, 0.96/)
lypts = (/0.16, 0.16/)
simbolxpt=(/0.755/)
simbolypt=(/0.1397/)
textxpt=(/0.86/)
textypt=(/0.14/)
datexpt=(/0.131/)
dateypt=(/0.14/)

; OUTSIDE BOX PLOTTING
gsn_polyline_ndc(wks,bxpts,bypts,resOutBox)
gsn_polyline_ndc(wks,lxpts,lypts,resOutLine)

; BOTTOM TEXT PLOTTING
gsn_text_ndc(wks,bottomsimbol,simbolxpt,simbolypt,sres)
gsn_text_ndc(wks,bottomtext,textxpt,textypt,txres)

draw(plot)

```

frame(wks)
delete(wks)

Figure 62. Additional code for Success Rate

(5) 데이터베이스 설계 및 구축

계절예측 및 검증 관련 자료를 효율적으로 관리하기 위하여 데이터베이스를 이용할 필요가 있다. 계절예측 및 검증을 위한 입출력 자료 중 스토리지의 파일 형태로 저장하지 않고 데이터베이스에 저장할 자료를 선별하여 논리 테이블을 설계하였다. 실제 데이터베이스를 구축하기 위해서 논리 테이블을 바탕으로 물리 테이블로 설계한 결과는 다음과 같다.

Table 12. Database table: clik_provider (list of model providers)

Column name	Data type	Key	Nullable
provider_name	varchar(45)	4)PK	N
country	varchar(45)	-	Y
ETC (Constraint)			
-			

Table 13. Database table: clik_model (list of models)

Column name	Data type	Key	Nullable
provider_name	varchar(45)	PK	N
model_name	varchar(45)	PK, FK ⁵⁾	N
start_ym	varchar(7)	-	N
end_ym	varchar(7)	-	Y
ETC (Constraint)			
FOREIGN KEY (provider_name) REFERENCES clik_provider (provider_name)			

Table 14. Database table: clik_variable (list of variables)

Column name	Data type	Key	Nullable
var_name	varchar(45)	PK	N
unit	varchar(45)	-	Y
long_name	varchar(45)	-	N
ETC (Constraint)			
-			

4) PK (Primary Key): Database table에서 각 행의 정보를 식별할 수 있는 컬럼

5) FK (Foreign Key): 다른 table의 PK를 참조하는 컬럼이며, 다른 table과의 관계를 나타내기 위하여 사용함

Table 15. Database table: klik_verification_method (list of verification skills)

Column name	Data type	Key	Nullable
cabinet	varchar(10)	PK	N
mme_class	varchar(10)	PK	N
method_name	varchar(20)	PK	N
long_name	varchar(50)	-	N
rep_name	varchar(30)	-	Y
ETC (Constraint)			
-			

Table 16. Database table: klik_data (list of model combinations)

Column name	Data type	Key	Nullable
id	int(11)	PK	N
year	int(11)	-	N
month	int(11)	-	N
model_list	varchar(1024)	-	N
status	varchar(20)	-	N
ETC (Constraint)			
id: AUTO_INCREMENT			

Table 17. Database table: clik_job (list of jobs to produce prediction and verification data)

Column name	Data type	Key	Nullable
id	int(11)	PK	N
data_id	int(11)	FK	N
lead_month	int(11)	-	N
cabinet	varchar(10)	-	N
year	int(11)	-	N
month	int(11)	-	N
model_list	varchar(1024)	-	N
job_type	varchar(20)	-	N
mme_class	varchar(10)	-	N
period	varchar(10)	-	N
variable_list	varchar(128)	-	N
verification_list	varchar(512)	-	Y
input_date	timestamp	-	N
start_date	timestamp	-	N
end_date	timestamp	-	N
status	varchar(20)	-	Y
user	varchar(128)	-	Y
ETC (Constraint)			
id: AUTO_INCREMENT			
variable_list: DEFAULT 'prec,slp,sst,t2m,t850,z500'			
input_date: DEFAULT CURRENT_TIMESTAMP			
start_date: DEFAULT '0000-00-00 00:00:00'			
end_date: DEFAULT '0000-00-00 00:00:00'			
FOREIGN KEY (data_id) REFERENCES clik_data (id)			

물리적 데이터베이스 테이블 설계 후 테이블을 생성할 실제 데이터베이스는 PaaS-TA에서 제공하는 MySQL을 사용하였다. PaaS-TA는 두 가지 MySQL 서비스를 제공한다. 첫 번째는 사용자 최대 접속수를 10개 이하로 제한하는 서비스이고 두 번째는 100개 까지 가능한 서비스이다. 첫 번째 서비스는 간단한 기능을 제공하는 소규모 응용 프로그램에 적합하고, 두 번째의 경우는 병렬 및 분산 처리가 필요하거나 중급 이상의 응용 프로그램에 적합하다. 계절예측 및 검증 시스템은 웹 서비스 뿐 아니라 사용자 요청을 여러 플랫폼 컨테이너에 분산하여 처리해야 하므로 두 번째 MySQL 서비스를 이용하기로 하였다.

데이터베이스 물리 설계 내역에 따라 데이터베이스 테이블 생성 스크립트를 작성하였다. PaaS-TA의 Mysql Service Instance를 신청하여 데이터베이스를 생성한 후 스크립트를 실행하여 테이블을 생성하였다.

(6) Open API 및 자료 생성 모듈 개발

계절예측 및 검증 시스템은 다음과 같이 크게 세 영역으로 구성된다. 이 절에서는 자료 생산, Open API 개발 방법에 대하여 기술한다.

Table 18. Three areas of the system

영역	역할
자료 생산	실질적인 계절예측 및 검증 자료 생산
Open API ⁶⁾	사용자 요청 처리, 웹 서비스의 지원 요청을 처리하고 결과 반환
웹 서비스	사용자가 계절예측 및 검증 자료를 확인하고 요청할 수 있는 GUI

① 계절예측 및 검증 자료 생산

Figure 49, 50는 계절예측 및 검증 자료를 생성하는 과정을 설계한 그림이다. Figure 49, 50에서와 같이 계절예측과 검증 생산 모듈은 동일하게 Job queue를 이용한다. Queue로부터 job id를 확인하고 데이터베이스에서 job의 상세 정보를 검색하여 job의 상태를 처리 시작으로 변경한다.

Queue (RabbitMQ)와 데이터베이스(MySQL)를 사용하기 위하여 Figure 63과 같이 프로젝트의 pom.xml에 dependency를 설정하였다. MySQL을 이용하는 방법은 여러 가지가 있지만 계절예측 및 검증 시스템은 mybatis 라이브러리를 사용하였다.

```

<dependencies>
  <dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>2.1.3</version>
  </dependency>

  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.amqp</groupId>
    <artifactId>spring-rabbit</artifactId>
  </dependency>

```

Figure 63. Maven dependency (pom.xml)

또한 Spring Boot의 설정 파일인 application.properties 파일에 다음과 같이 환경을 설정하였다.

6) Open API: 웹 사이트의 기능을 이용할 수 있도록 공개된 Application Programming Interface

```
# mysql configuration
spring.datasource.url=jdbc:mysql://bb1d89ade1b610b0:[Password]@10.200.8.40:3306/op_4457dea0
_d7ca_444a_9a45_7c77347514bf?zeroDateTimeBehavior=convertToNull
spring.datasource.username=bb1d89ade1b610b0
spring.datasource.password=[Password]
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.max-pool-size=6
spring.datasource.max-active=6
spring.datasource.max-idle=4
spring.datasource.min-idle=2
spring.datasource.hikari.maximum-pool-size=6

# rabbit mq
spring.rabbitmq.host=10.200.8.36
spring.rabbitmq.port=5672
spring.rabbitmq.username=openapi
spring.rabbitmq.password=[Password]
spring.rabbitmq.virtual-host=83d53fff-69a7-4b7d-8b76-9c913e277ca5
spring.rabbitmq.template.exchange=exchange_clik_comp
spring.rabbitmq.template.routing-key=clik_comp
spring.rabbitmq.template.default-receive-queue=queue_clik_comp
```

Figure 64. Configurations for RabbitMQ and MySQL

자바 개발 프로젝트에서 RabbitMQ를 사용하여 메시지를 받기 위해 Figure 65와 같은 클래스를 추가하였다. MySQL mybatis 라이브러리를 사용하기 위해서는 Figure 66과 같은 설정이 필요하다.

```

package org.apcc.clik.precomp;

import org.springframework.amqp.rabbit.annotation.EnableRabbit;
import org.springframework.amqp.rabbit.core.RabbitTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.amqp.core.Message;

@Component
@EnableRabbit
public class JobReceiver
{
    @Autowired
    private RabbitTemplate rabbitTemplate;

    public String getJob()
    {
        Message msg = rabbitTemplate.receive();
        if (msg == null) return null;
        byte []body = msg.getBody();
        if (body == null) return null;

        String strMsg = new String(body);

        return strMsg;
    }
}

```

Figure 65. Class to get messages from RabbitMQ

```

package org.apcc.clik.precomp;

import javax.sql.DataSource;
import org.apache.ibatis.session.SqlSessionFactory;
import org.mybatis.spring.SqlSessionFactoryBean;
import org.mybatis.spring.SqlSessionTemplate;
import org.mybatis.spring.transaction.SpringManagedTransactionFactory;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.io.support.PathMatchingResourcePatternResolver;
import org.springframework.jdbc.datasource.DataSourceTransactionManager;
import org.springframework.transaction.PlatformTransactionManager;
import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration
@EnableTransactionManagement
public class MybatisConfig
{
    @Bean
    public SqlSessionFactory sqlSessionFactory (DataSource dataSource) throws Exception
    {
        SqlSessionFactoryBean sqlSessionFactory = new SqlSessionFactoryBean();
        sqlSessionFactory.setDataSource(dataSource);
        sqlSessionFactory.setTransactionFactory(new SpringManagedTransactionFactory());
        PathMatchingResourcePatternResolver resolver = new PathMatchingResourcePatternResolver();
        sqlSessionFactory.setMapperLocations(resolver.getResources("classpath:mapper/**/*.xml"));
        return sqlSessionFactory.getObject();
    }

    @Bean
    public SqlSessionTemplate sqlSession (SqlSessionFactory sqlSessionFactory)
    {
        return new SqlSessionTemplate(sqlSessionFactory);
    }

    @Bean
    public PlatformTransactionManager transactionManager(DataSource dataSource) {
        return new DataSourceTransactionManager(dataSource);
    }
}

```

Figure 66. Mybatis configuration class

mybatis는 Figure 66에서와 같이 지정된 위치에서 mapper xml 파일을 로드하여 사용한다.

mapper xml 파일을 Figure 67과 같이 프로젝트에 추가하고 Figure 68와 같이 설정하여 application이 시작할 때 xml 파일을 스캔하여 초기화 하도록 할 수 있다.

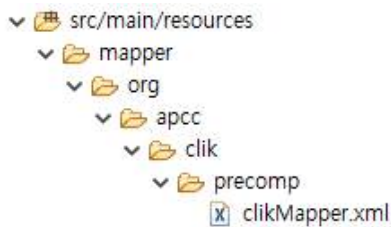


Figure 67. Mapper xml for mybatis

```
package org.apcc.klik.precomp;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.scheduling.annotation.EnableScheduling;

@EnableScheduling
@MapperScan(basePackages = "org.apcc.klik.precomp")
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Figure 68. Start application scanning mapper

mapper xml 파일은 실제 모든 MySQL query를 포함하고 있으며 Figure 69과 같이 작성한다. 모든 insert, select, update, delete query는 id를 가지고 있으며 xml 파일에서 입출력 자료 형식, query문을 정의할 수 있다.

xml	version="1.0" encoding="UTF-8"
DOCTYPE	mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"
mapper	(cache-ref cache resultMap* parameterMap* sql* insert* update* delete* select*)+
namespace	org.apcc.klik.precomp.ClikMapper
!--	Start CLIK DATA
insert	(selectKey include trim where set foreach choose if bind)*
id	insertClikData
useGeneratedKeys	true
keyProperty	id
parameterType	org.apcc.klik.dto.ClikDataDTO
	INSERT INTO klik_data (year, month, model_list, status) VALUES (#{year}, #{month}, #{model_list}, #{status})
select	(include trim where set foreach choose if bind)*
id	selectClikDataById
parameterType	int
resultType	org.apcc.klik.dto.ClikDataDTO
	SELECT * FROM klik_data WHERE ID = #{id}

Figure 69. Sample of mapper xml

또한 데이터베이스의 transaction을 사용하기 위해 Figure 70와 같은 클래스를 정의하였다.

```

package org.apcc.clik.precomp;

import org.apache.ibatis.transaction.TransactionException;

@Service
@Scope("prototype")
public class MybatisTransactionManager extends DefaultTransactionDefinition
{
    private static final long serialVersionUID = -1375151959664915520L;

    protected PlatformTransactionManager transactionManager;

    TransactionStatus status;

    public MybatisTransactionManager(PlatformTransactionManager transactionManager)
    {
        this.transactionManager = transactionManager;
    }
    public void start() throws TransactionException
    {
        status = transactionManager.getTransaction(this);
    }
    public void commit() throws TransactionException
    {
        if (!status.isCompleted())
        {
            transactionManager.commit(status);
        }
    }
    public void rollback() throws TransactionException
    {
        if (!status.isCompleted())
        {
            transactionManager.rollback(status);
        }
    }
    public void end() throws TransactionException
    {
        rollback();
    }
}

```

Figure 70. Class to support transaction for mybatis

계절예측 및 검증 시스템은 자료 생산을 위해 주기적(1초)으로 queue를 검사하여 job id를 확인한다. job id가 전송되면 job 처리를 시작하고 처리가 끝난 후 1초 후 다시 queue를 검사한다. Figure 71는 job 처리를 시작하는 코드를 보여준다. 지속적이고 주기적인 처리를 위하여 scheduler를 사용하였다.

```

package org.apcc.clik.precomp;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;

@Component
public class JobScheduler
{
    @Autowired
    ClikEnvironment env;

    // rabbitmq
    @Autowired
    JobReceiver jobReceiver;

    // mysql
    @Autowired
    private ClikMapper clikMapper;

    private final Logger logger = LoggerFactory.getLogger(this.getClass());

    @Scheduled(initialDelay = 1000, fixedDelay = 1000)
    public void loadJob()
    {
        String jobid = jobReceiver.getJob();
        logger.debug("check queue");

        if (jobid != null && jobid.length() > 0)
        {
            logger.info("Start clik job : " + jobid);
            JobProcessor processor = new JobProcessor(env, clikMapper);
            processor.run(jobid);
        }
    }
}

```

Figure 71. Starting to process job

자료 생산은 실질적으로 플랫폼에 이식된 현업 시스템이 담당하지만, 입력된 job에 따라 현업 코드 실행할 수 있도록 쉘 스크립트를 생성한다. Figure 72과 같은 기본 template 쉘 스크립트에 환경설정 내용(“!! !!”)을 기록하여 현업 코드 실행에 필요한 정보를 전달하였다.

```

#!/bin/bash

export AFS=!!AFS!!
export MyAFS=$AFS
export IndvModel=!!IndvModel!!
export AFS_NCL=!!AFS_NCL!!
export SEAS_START=!!SEAS_START!!
export SEAS_END=!!SEAS_END!!
export LEAD_TIME=!!LEAD_TIME!!
export START_MON=!!START_MON!!
export SEASON=!!SEASON!!
export SEASONAL=!!SEASONAL!!
export STORE=!!STORE!!
export CABINET=!!CABINET!!
export MME_CLASS=!!MME_CLASS!!

```

```

export MODEL=!!MODEL!!
export MME_VER=!!MME_VER!!
export SELECTED_YEAR=!!SELECTED_YEAR!!
export COMBI_DIR=!!COMBI_DIR!!
export MME_MODELS=$COMBI_DIR/MODELS

vars=( !!VARS!! )
SYEAR=!!SYEAR!!
EYEAR=!!EYEAR!!
NCL_FILE=" "

if [ "$MME_CLASS" == "PMME" ]; then
    NCL_FILE=$AFS/CODE/PMME/GAUS/NCL/GAUS.ncl
else
    NCL_FILE=$AFS/CODE/DMME/SCM/NCL/SCM.ncl
fi

if [ "$SCABINET" == "Hindcast" ]; then
    iYYYY=$SYEAR
    while (( "${iYYYY}" <= "${EYEAR}" )); do
        echo "START_YEAR $iYYYY"
        export START_YEAR=${iYYYY}
        for var_name in "${vars[@]}"; do
            export VNAME=$var_name
            echo $VNAME
            ncl $NCL_FILE < /dev/null
        done
        iYYYY='echo "${iYYYY} +1" | bc'
    done
else
    export START_YEAR=${SYEAR}
    for var_name in "${vars[@]}"; do
        export VNAME=$var_name
        ncl $NCL_FILE < /dev/null
    done
fi

```

Figure 72. Shell script to run code producing MME

셸 스크립트를 실행하는 과정에서 발생하는 현업 코드의 콘솔 출력을 상황에 따라 로그파일
 일에 기록할 수 있도록 Figure 73과 같은 클래스를 개발하였다.

```

public class RuntimeProcess
{
    public static boolean run(String command, Logger logger)
    {
        logger.debug("Start to run command : " + command);
        try
        {
            Process proc = Runtime.getRuntime().exec(command);
            BufferedReader reader = new BufferedReader(new InputStreamReader(proc.getInputStream()));
            String line = null;

            while((line = reader.readLine()) != null)
            {
                logger.debug(line);
            }
        }
        catch(Exception ex)
        {
            logger.error("Failed to run command : " + command, ex);
            return false;
        }

        logger.debug("Finished to run command : " + command);

        return true;
    }
}

```

Figure 73. Class to run shell scripts

② 웹 서비스 지원 Open API 개발

웹 서비스 지원 Open API는 기본적으로 Rest API를 기반으로 하고 있다. Rest (Representational State Transfer)는 자원을 이름(자원의 표현)으로 구분하여 해당 자원의 상태 (정보)를 주고받는 모든 것을 의미한다. REST는 기본적으로 웹의 기존 기술과 HTTP 프로토콜을 그대로 활용하기 때문에 웹의 장점을 최대한 활용할 수 있는 아키텍처 스타일이다. API(Application Programming Interface)란 데이터와 기능의 집합을 제공하여 컴퓨터 프로그램 간 상호작용을 촉진하며, 서로 정보를 교환가능 하도록 하는 것이다. REST API는 REST 기반으로 서비스 API를 구현한 것이다. 최근 Open API (누구나 사용할 수 있도록 공개된 API: 구글 맵, 공공 데이터 등), 마이크로 서비스(하나의 큰 애플리케이션을 여러 개의 작은 애플리케이션으로 쪼개어 변경과 조합이 가능하도록 만든 아키텍처) 등을 제공하는 업체 대부분은 REST API를 제공한다.

웹 서비스 지원 기능은 Rest API의 형태로 개발하기 위해 Spring Rest 라이브러리를 이용하였다. Controller 클래스를 정의하고 “@RestController” 태그를 표시한다. 자료 전송 메소드를 정의하고 HTTP POST/GET 방식을 이용한 요청을 받을 수 있도록 “@PostMapping”, “@GetMapping” 태그와 함께 URI 패스를 설정한다.

계절예측 및 검증 웹 서비스에서 사용자가 자료 조회, 다운로드 등을 요청하는 경우 웹 서비스가 직접 실행하지 않고 Open API를 요청한다. 사용자가 요청한 조건에 맞는 자료가 생산되어 있는지 조회를 요청하고, 자료가 있으면 image 출력을 요청하거나 다운로드 URL을 요청할 수 있다. 원하는 자료가 생산되지 않은 경우에는 기후자료서비스(CLIK)에 자료를 요청하는 정식 job을 요청한다. 기후자료서비스(CLIK)는 상황에 따라 내부 자료생산 job 입력 및 상태 조회를 위하여 API를 호출할 수 있다. 계절예측 및 검증 웹 서비스 및 기후자료서비스 job

processor를 지원하기 위한 API 목록은 Table 19과 같다.

Table 19. List of API to support other services

URL	HTTP Method	교환 데이터	교환 데이터 형식	기능
prediction.apcc21.org /addjob/	POST	자료생산 내용	JSON	자료생산 job 입력(job processor 요청)
prediction.apcc21.org /status/	GET	Job ID	Path string, JSON	자료생산 job의 진행상황 조회(job processor 요청)
prediction.apcc21.org /imagepath/	POST	자료조회 조건	JSON	조건에 맞는 plot 이미지 조회
prediction.apcc21.org /datapath/	POST	자료조회 조건	JSON	조건에 맞는 netcdf 자료 조회
prediction.apcc21.org /image/	GET	이미지 URI	Image (PNG)	해당 URL의 PNG 이미지 전달
prediction.apcc21.org /runningjob/	POST	자료생산 조건	JSON	조건에 해당하는 다른 job 이 이미 실행중인지 조회
prediction.apcc21.org /geturl/image/	POST	자료조회 조건	JSON	조건에 해당하는 plot image를 다운로드할 수 있는 URL 조회
prediction.apcc21.org /geturl/data/	POST	자료조회 조건	JSON	조건에 해당하는 데이터 (netcdf)를 다운로드할 수 있는 URL 조회
prediction.apcc21.org /download/	GET	다운로드 URI	BIN (octet-stream)	해당 URL의 자료 전달

③ 사용자 요청 Job 처리 Open API 개발

사용자는 웹 서비스나 Open API 클라이언트 tool을 이용하여 자료를 요청할 수 있다. 웹 서비스나 Open API 클라이언트가 자료를 요청하는 API는 다음과 같다.

Table 20. List of API to support the others

URL	HTTP Method	교환 데이터	교환 데이터 형식	기능
request.apcc21.org/apccdata/	POST	자료요청 내용	JSON	자료요청 job 입력
request.apcc21.org/status/	GET	Job ID	Path string	자료요청 job의 진행상황 조회

download.apcc21.org/	GET	Job ID	BIN (octet-stream)	해당 job의 결과 자료 다운로드 요청
----------------------	-----	--------	-----------------------	-----------------------

Table 20의 request.apcc21.org/apccdata API는 계절예측 및 검증 뿐 아니라 MME, 개별모델, Clipped CMIP5 등의 자료 다운로드 서비스도 함께 제공한다. 요청하는 자료의 상세 내역에서 어떤 자료를 요청하는지 기술하면 자료서비스(CLIK)의 processor에서 자료를 구분하여 처리한다.

사용자는 Python 클라이언트를 이용하여 Figure 74, 75와 같이 계절예측 및 검증 자료를 요청할 수 있다.

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'Prediction',
        'type': 'FORECAST',
        'method': 'SCM',
        'model': ['APCC_SCOPS', 'BOM_ACCESS-S1',
        'KMA_GLOSEA5GC2'],
        'variable': ['prec', 'slp', 'sst', 't2m', 't850', 'z500'],
        'year': '2020',
        'month': '7'
    },
    'prediction.zip'
)
```

Figure 74. Sample using Python: Prediction

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'Verification',
        'type': 'HINDCAST',
        'method': 'GAUS',
        'model': ['APCC_SCOPS', 'BOM_ACCESS-S1',
```

```
'CWB_GFST119', 'KMA_GLOSEA5GC2'],  
'variable': ['prec', 'slp'],  
'verification': 'ROC_Curve',  
'year': '2020',  
'month': '9'  
,  
'verification.zip'  
)
```

Figure 75. Sample using Python: Verification

(7) 계절예측 및 검증 웹 서비스 개발

계절예측 및 검증 웹 서비스를 개발에는 스프링 부트(Spring Boot)를 비롯한 MVC 모델의 방법론과 ajax 등의 다양한 기술들이 사용되었다.

계절예측 및 검증 웹 서비스를 개발하는데 사용된 스프링 부트에 소개를 하려면 우선 스프링 프레임워크(Spring Framework)에 대해서 알아야 한다. 스프링 프레임워크는 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크로서 간단히 스프링(Spring)이라고도 불린다. 동적인 웹 사이트를 개발하기 위한 여러 가지 서비스를 제공하고 있다. 대한민국 공공기관의 웹 서비스 개발 시 사용을 권장하고 있는 전자정부 표준프레임워크의 기반 기술로서 쓰이고 있다. 이러한 스프링 프레임워크는 기능이 많은 만큼 환경설정이 복잡한 편이다. 이에 어려움을 느끼는 사용자들을 위해 나온 것이 바로 스프링 부트이다. 스프링 부트는 스프링 프레임워크를 사용하기 위한 설정의 많은 부분을 자동화하여 사용자가 정말 편하게 스프링을 활용할 수 있도록 돕는다. 스프링 부트 starter dependency만 추가해주면 바로 API를 정의하고, 내장된 톰캣으로 웹 어플리케이션 서버를 실행할 수 있다. 심지어 스프링 홈페이지의 이니셜라이저를 사용하면 바로 실행 가능한 코드를 만들어준다. 실행환경이나 의존성 관리 등의 인프라 관련 등은 신경 쓸 필요 없이 바로 코딩을 시작하면 된다.

계절예측 및 검증 웹 서비스 개발에 MVC 모델을 사용하였는데 MVC 모델은 소프트웨어 공학에서 사용되는 소프트웨어 디자인 패턴이다. 이 패턴을 성공적으로 사용하면, 사용자 인터페이스로부터 비즈니스 로직을 분리하여 애플리케이션의 시각적 요소나 그 이면에서 실행되는 비즈니스 로직을 서로 영향 없이 쉽게 고칠 수 있는 애플리케이션을 만들 수 있다. MVC에서 모델은 애플리케이션의 정보(데이터)를 나타내며, 뷰는 텍스트, 체크박스 항목 등과 같은 사용자 인터페이스 요소를 나타내고, 컨트롤러는 데이터와 비즈니스 로직 사이의 상호동작을 관리한다. 이러한 디자인 패턴을 기반으로 해서 Spring Boot를 이용하여 계절예측 및 검증 웹 서비스를 개발하였다.

Ajax (Asynchronous JavaScript and XML)는 비동기적인 웹 애플리케이션의 제작을 위해 이용하는 웹 개발 기법이다. 표현 정보를 위한 HTML과 CSS를 사용하고 동적인 화면 출력 및 표시 정보와의 상호작용을 위한 DOM, 자바스크립트를 조합하여 이용할 수 있고, 웹 서버와 비

동기적으로 데이터를 교환하고 조작하기 위한 XML, XSLT, XMLHttpRequest를 조합해서 사용할 수 있다. 계절 예측 및 검증 웹 서비스 개발시에 사용자의 편의를 제공하기 위해 ajax를 사용하였는데 과거의 웹 페이지에서는 화면 전환시에 지속적인 페이지 이동을 하였으나 ajax를 사용함으로써 고속으로 화면을 전환할 수 있다. 예를 들어 사용자가 년도와 월을 변경하면 해당하는 년도와 월에 맞게 모델리스트를 설정하여야 하는데, 예전의 기술은 모델리스트를 변경하기 위해서는 지속적인 화면이 새로 고침이 발생하여야 하지만, ajax 기술을 사용하면 화면 새로 고침 없이 모델 리스트만 변경이 가능하도록 개발이 가능하다. 이러한 장점으로 인해 인터랙티브한 웹페이지를 구현할 수 있다.

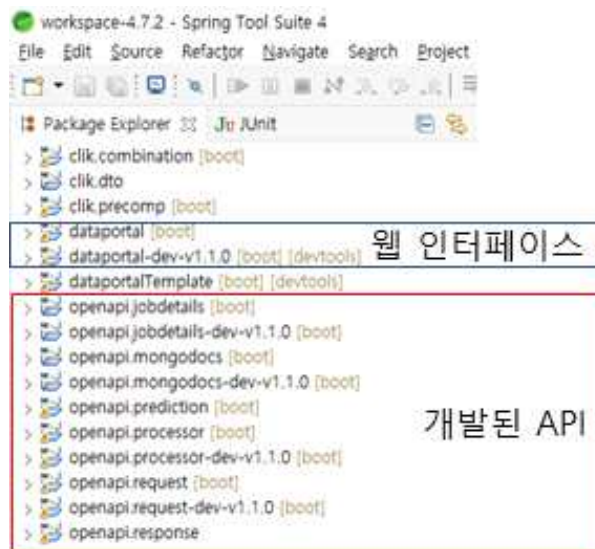


Figure 76. Web interface and API Source code

REST는 Representational State Transfer이라는 용어의 약자로서 2000년도에 로에 필딩(Roy Fielding)의 박사학위 논문에서 최초로 소개되었다. 로이 필딩은 HTTP의 주요 저자 중 한 사람으로 그 당시 웹(HTTP) 설계의 우수성에 비해 제대로 사용되어지지 못하는 모습에 안타까워하며 웹의 장점을 최대한 활용할 수 있는 아키텍처로서 REST를 발표하였다. REST는 자원(Resource - URI), 행위(Verb, HTTP METHOD) 그리고 표현(Representations)으로 구성되어 있다. 이런 REST API는 HTTP 프로토콜의 인프라를 그대로 사용하기에 별도의 인프라를 구축할 필요가 없고, HTTP 프로토콜의 표준을 최대한 활용하며, HTTP 표준 프로토콜을 따르는 모든 플랫폼에서 사용이 가능하기에 API를 REST 기반으로 해서 개발을 하였다. Figure 76은 실제 계절예측 및 검증 시스템 개발에 사용되고 있는 Spring Tool Suite의 화면이다. 그림에서 보는 바와 같이 계절예측 및 검증 시스템은 웹 인터페이스와 각각의 기능들을 담당하는 API들로 나누어져 있다. 웹 인터페이스에서 사용자 요청에 따른 필요한 기능들은 아래 API들을 이용해서 요청하고 결과를 받아서 웹 인터페이스를 통해서 사용자에게 제공을 하게 된다.

(8) 계절예측 및 검증 서비스 실행

계절예측 웹 서비스는 계절예측 결과를 제공하는 웹 페이지이다(Figure 77). 사용자는 몇 가지 조건을 설정하면 원하는 결과를 얻을 수 있다. 사용자가 설정하는 조건으로는 계절예측하기를 원하는 ① 년도와 월을 선택하고 ② Deterministic와 Probabilistic 기법 중에서 원하는 기법을 선택한다. 그리고 최종적으로 사용하고자 하는 ③ 모델 조합을 선택하고 ④ “Predict” 버튼을 누르면 사용자가 원하는 결과(⑤)를 얻을 수 있다. 만약 사용자가 설정한 결과가 없으면 자동으로 사용자 설정 조건으로 이미지를 생성하는 프로세스를 수행하여 결과를 생성하게 된다.

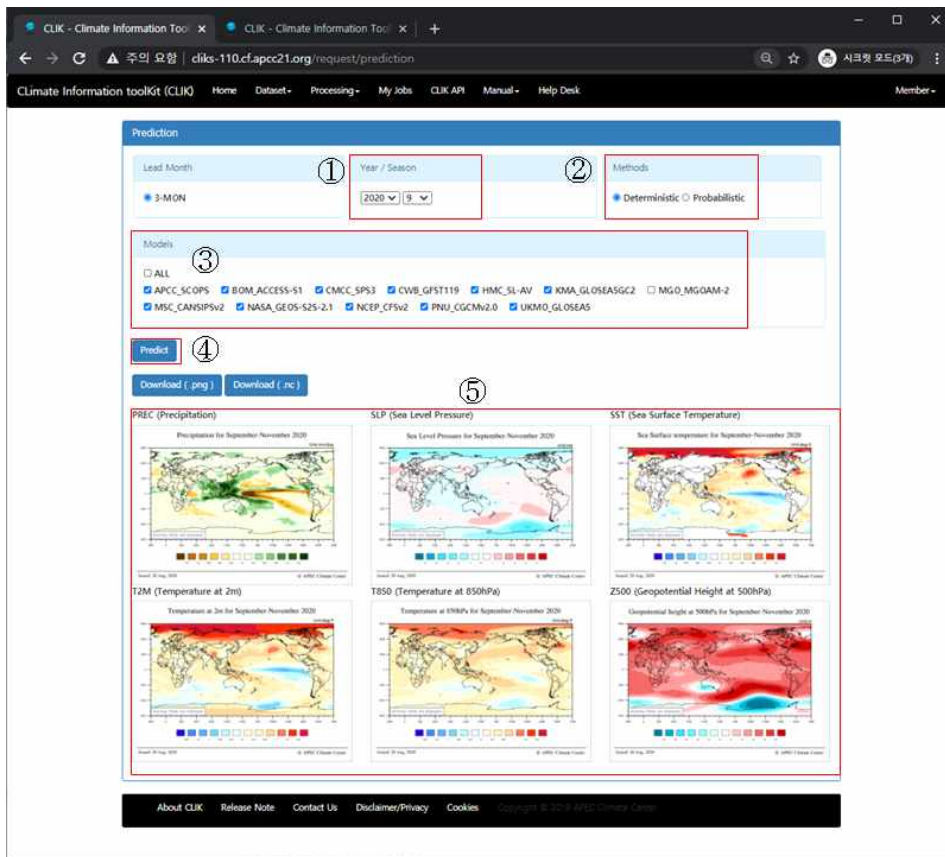


Figure 77. Prediction web interface

검증 웹 서비스는 검증 결과를 제공하는 웹 페이지이다(Figure 78). 계절예측 페이지와는 동일한 웹 인터페이스 구조로 개발이 되었다. 계절예측과 마찬가지로 사용자는 몇 가지 조건을 설정하는 원하는 검증 결과를 얻을 수 있다. 사용자가 설정하는 조건으로는 원하는 ①년도와 월을 선택하고 ② 4개의 검증 기법 중 원하는 기법을 선택한다. 그리고 ③변수를 선택하면 되는데, 변수는 총 6개중에서 원하는 변수를 선택하면 된다. 그리고 최종적으로 사용하고자 하는 ④모델 조합을 선택하고 ⑤ “Verify” 버튼을 누르면 사용자가 원하는 결과(⑥)를 얻을 수 있다. 만약 사용자가 설정한 결과가 없으면 자동으로 사용자 설정 조건으로 이미지를 생성하는 프로

세스를 수행하여 결과를 생성하게 되는데, 이 부분은 계절예측과 동일한 프로세스로 진행이 되도록 개발이 되었다.

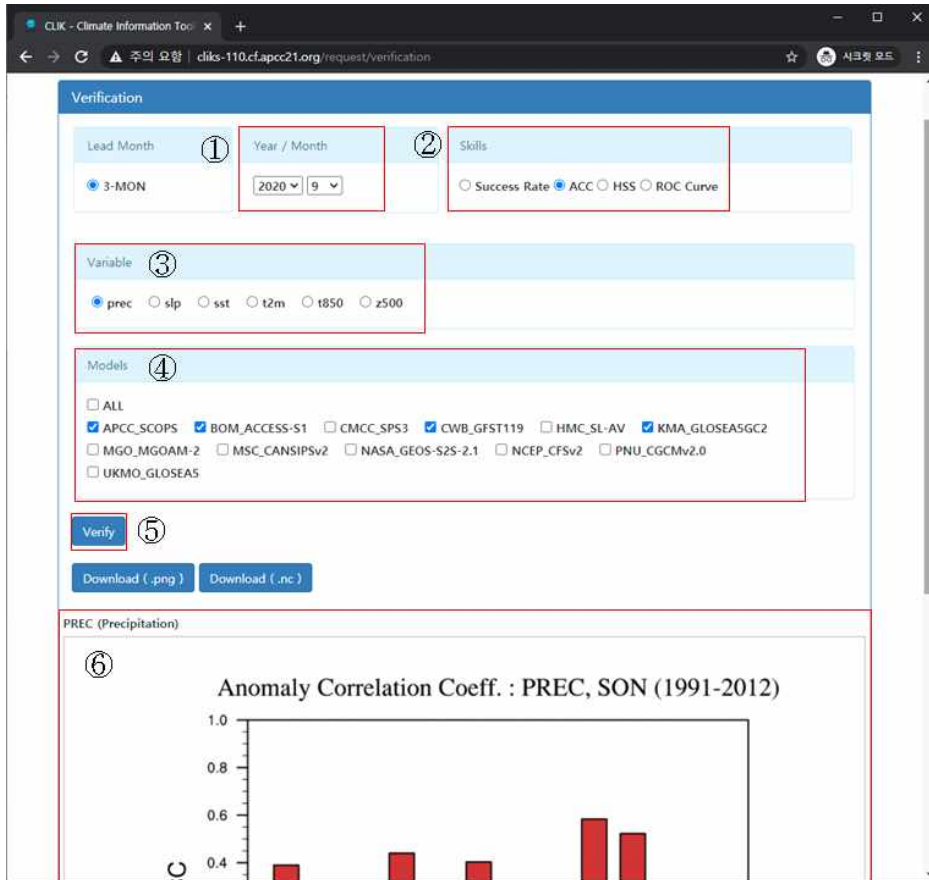


Figure 78. Verification web interface

사용자는 자신의 로컬 PC 또는 서버에서 Python 혹은 Java 언어를 이용하여 자료를 직접 요청하고 다운로드할 수 있다. 기후자료서비스 홈페이지(clicks.apcc21.org)에서 API 클라이언트를 다운로드하여 개발 코드에 추가하여 사용할 수 있다. API 클라이언트는 사용자 확인 결과, 입력한 job id, 자료 요청의 진행상황 등을 출력한다.

```

[jhshin@newAFS-dev pythontest]$ cat prediction.py
#!/usr/bin/env python
import apccapi

c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'Prediction',
        'type': 'FORECAST',
        'method': 'SCM',
        'model': ['APCC_SCOPS', 'BOM_ACCESS-S1', 'KMA_GLOSEA5GC2', 'MGO_MGOAM-2'],
        'variable': ['prec', 'slp', 't2m', 't850', 'z500'],
        'year': '2020',
        'month': '7'
    },
    'prediction.zip'
)
[jhshin@newAFS-dev pythontest]$ python prediction.py
<Response [202]>
[2020-11-05 11:35:53,919] [INFO] Hello jhshin77.
[2020-11-05 11:35:53,919] [INFO] Your job id is 5fa3652b9c46610007f45ca1
[2020-11-05 11:35:53,919] [INFO] Request is Queued
[2020-11-05 11:35:56,949] [INFO] Request is Running
[2020-11-05 11:36:03,009] [INFO] Request is Complete
[2020-11-05 11:36:03,009] [INFO] Start to save file - prediction.zip
[2020-11-05 11:36:04,919] [INFO] Done

```

Figure 79. Sample using python client (Prediction)

```

[jhshin@newAFS-dev pythontest]$ cat verification.py
#!/usr/bin/env python
import apccapi

c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'Verification',
        'type': 'HINDCAST',
        'method': 'GAUS',
        'model': ['APCC_SCOPS', 'BOM_ACCESS-S1', 'CWB_GFST119', 'KMA_GLOSEA5GC2'],
        'variable': ['prec', 'slp', 'sst', 't2m', 't850', 'z500'],
        'verification': 'ROC_Curve',
        'year': '2020',
        'month': '9'
    },
    'verification.zip'
)
[jhshin@newAFS-dev pythontest]$ python verification.py
<Response [202]>
[2020-11-05 13:31:43,592] [INFO] Hello jhshin77.
[2020-11-05 13:31:43,592] [INFO] Your job id is 5fa380519c46610007f45ca2
[2020-11-05 13:31:43,593] [INFO] Request is Queued
[2020-11-05 13:31:46,622] [INFO] Request is Running
[2020-11-05 13:32:01,768] [INFO] Request is Complete
[2020-11-05 13:32:01,768] [INFO] Start to save file - verification.zip
[2020-11-05 13:32:02,100] [INFO] Done
[jhshin@newAFS-dev pythontest]$ █

```

Figure 80. Sample using python client (Verification)

나. 플랫폼 기반 APCC 기후자료처리 시스템 개발

(1) 개요 및 필요성

APCC는 현재 기후자료를 기반으로 한 다양한 형태의 기후정보시스템 (ADSS⁷⁾, CLIPs⁸⁾, OpenWPS⁹⁾)등을 운영하고 있으며, 동시에 개별 기후정보서비스 간의 유사 기능 등을 통합하고 확대하기 위한 플랫폼 기반 기후자료처리시스템을 구축하고 있다. 2019년에는 기존의 기후자료 서비스(ADSS)를 대체할 수 있는 플랫폼 기반 기후자료 제공 서비스를 구축하였으며, 2020년에는 CLIPs의 기후자료 추출·가공 기능을 플랫폼화하기 위해 웹브라우저 기반의 새로운 기후자료처리시스템을 구축하고자 하였다. 기존 응용프로그램 기반으로 제공되었던 CLIPs를 웹기반으로 변환하고 APCC 플랫폼에 이식을 통해 확장된 현대의 자료서비스 제공하고자 하였다.

(2) 플랫폼기반 기후자료서비스 개발

(가) 계절예측자료 추출 서비스 웹 인터페이스 개발

기존의 기후자료처리시스템(CLIPs)은 사용자 컴퓨터에 직접 설치하여 사용하는 응용소프트웨어 형태로 서비스를 제공하였다. 하지만, 다양한 운영체제에서 동작하기에는 제약 사항이 많았다. 그리하여 범용으로 사용할 수 있는 웹 브라우저를 이용하여 웹 기반 자료처리 시스템은 자료추출 서비스를 개발하였다.

자료추출 서비스는 APCC에서 생산하는 계절예측자료를 이용하여 사용자가 원하는 지역, 변수, 기간, 모델, 앙상블 기법 등을 이용하여 자료를 추출할 수 있으며, 개별모델을 이용하여 평균값이나 사용자가 Hindcast기간을 자유롭게 조절하면서 Anomaly 계산한 결과를 얻을 수 있도록 구성하였다. 사용자들에게 자주 사용하는 지역을 미리 정의하여 지역을 선택하여 자료를 추출할 수도 있고 특정 좌표를 입력하여 자료를 추출할 수 있도록 기능을 제공한다.

상단에는 계절예측자료의 다중모델앙상블 자료나 협약기관에서 제공하는 개별모델 자료를 선택하고 선택된 자료의 특성에 따라 변수, 리드타임, 계산방법, 자료생산 기간을 선택할 수 있는 대상 자료를 선택하는 기능과 자주 사용하는 지역명칭이나 사용자가 직접 좌표 값을 입력하여 대상 자료에서 추출하고자 하는 지역을 선택할 수 있도록 인터페이스를 구성하였다. 개별 모델에서 자료의 평균값이나 편차 값을 계산할 수 있는 조건을 추가하여 계산된 결과를 각 리드타임별로 지도기반의 시각화 자료를 통해 사용자에게 직접 제공하도록 하였다. 이런 계산결과로 표출된 자료는 사용자가 원하는 이미지, 아스키 형태의 파일로 포맷을 변환하여 제공받을 수 있는 기능을 제공하도록 사용자 인터페이스를 구성하였다.

7) APCC Data Service System(APEC기후센터 자료서비스시스템)

8) CLimate Information Processing System(기후정보처리시스템)

9) Open-Web Processing Service(기후자료처리 인터페이스 서비스)

MME Individual Model

Variable:

Leadtime:

Method:

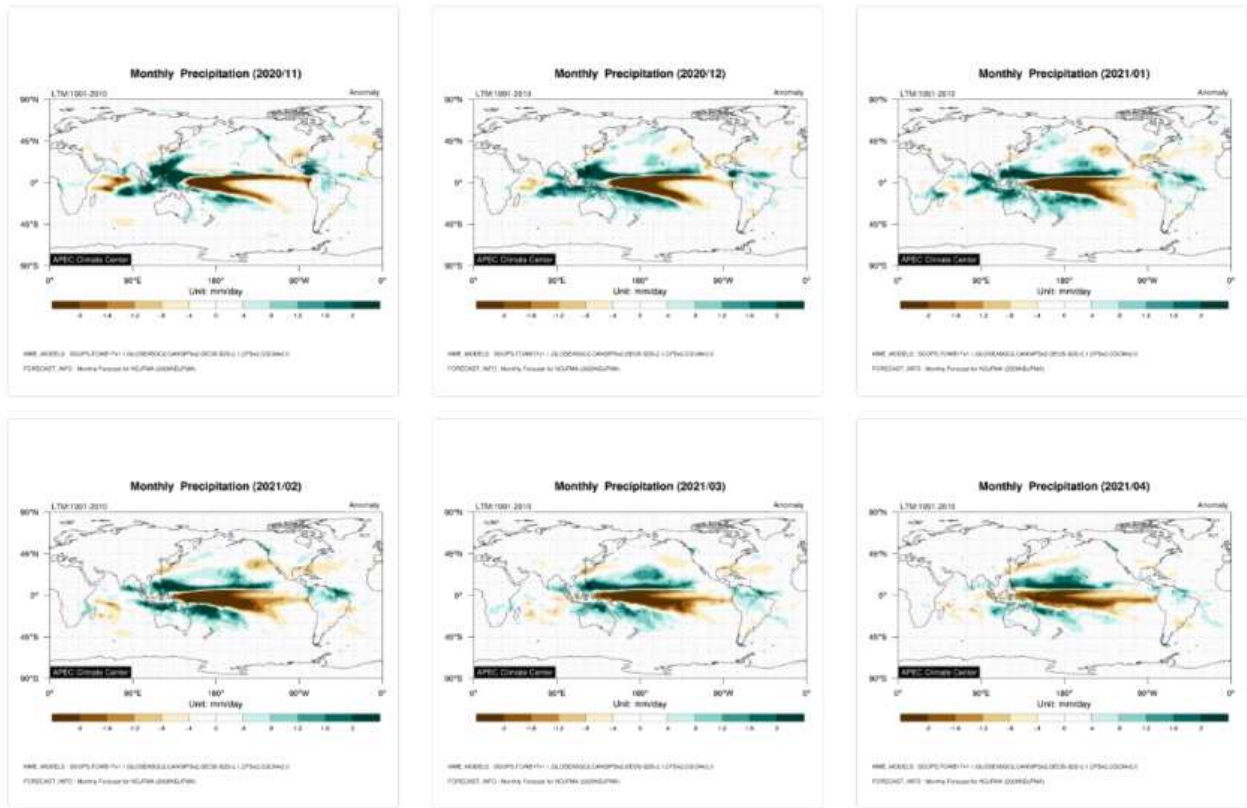
Issued: /

Coordinate:

Region:

Mean
 Anomaly
 Climatology Period: -

Data Plot



Data Clipping (rectangle)

Forecast Hindcast : -

Output file format:

Run for data clipping

Figure 81 . User web interface for Seasonal forecast data extraction.

(나) 기후자료 및 관측자료 합성 서비스 웹 인터페이스 개발

자료합성 서비스는 APCC의 예측자료와 관측 자료를 비교할 수 있도록 하나의 화면을 나누어 각각 분석할 수 있도록 화면을 구성하였으며 예측자료는 다중모델 앙상블로 생성된 아노말리를 제공하고 관측 자료를 사용자가 평균값이나 특정기간을 선택하여 아노말리를 계산할 수 있도록 화면을 구성하였다. 사용자들의 편의를 위해 여러 조건은 추가하고 삭제할 수 있도록 장바구니 기능을 구현하여 여러 조건을 합성할 수 있도록 기능을 구성하였다. 자료처리 서

비스는 모든 결과 값을 NetCDF나 엑셀 파일 형태로 내려 받을 수 있는 기능을 제공하도록 구성하였다. 자료추출 서비스는 계절예측 자료를 대상으로 사용자가 가공하고 표출하는 서비스를 제공한다. 자료합성서비스는 계절예측자료와 관측 자료를 상호 비교할 수 있도록 관측 자료에 대한 편차 값이나 평균값을 제공하고 전 지구 기반으로 시각화 이미지를 표출하였다. 사용자는 필요에 따라 아스키 형태로 파일을 제공받거나 이미지를 다운받아 사용할 수 있도록 사용자 인터페이스를 구성하였다.

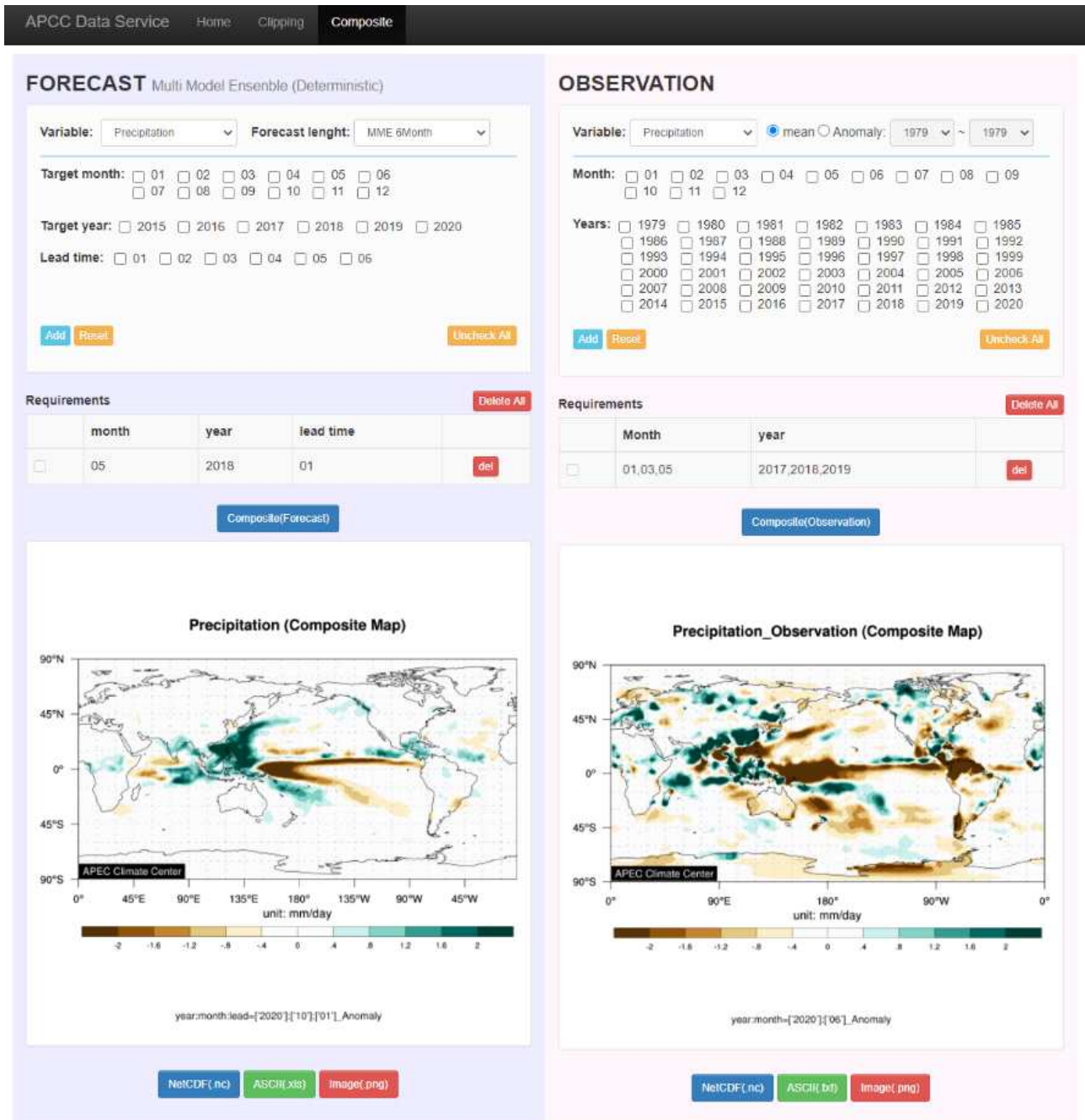


Figure 82 . User web interface for Seasonal forecast data and observation data composite

(다) 계절예측 자료 지역 및 변수 추출 서비스 개발

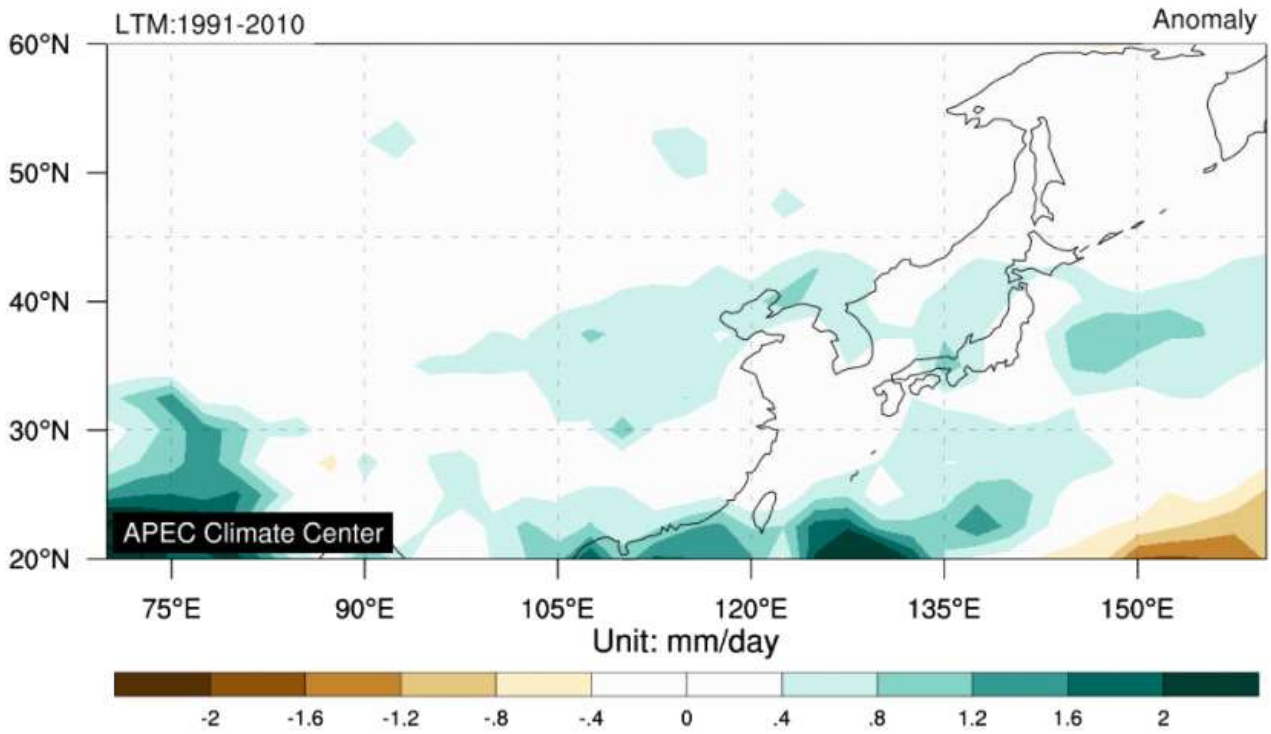
기후자료처리서비스의 자료추출 기능은 계절예측자료를 대상으로 사용자가 원하는 지역,

변수, 평균값, 아노말리 값을 계산하여 추출하는 서비스이다. 아래 Figure 83와 같이, 사용자는 APCC에서 생산된 MME나 MME참여기관에서 생산된 개별모델을 선택할 수 있으며 선택된 종류의 특성에 맞도록 하위메뉴가 변경되도록 하였다. 제공되는 변수는 강수, 8미터 온도, 해수면 온도, 해면압력, 850Hpa 온도 등 다중모델 및 개별모델에서 보유하고 있는 모든 변수를 제공한다. 다중모델의 경우, 기간에 따라 6개월과 3개월로 나누어 제공하고 결정론적 다중모델앙상블이나 확률론적 다중모델 앙상블로 나누어 제공하였다. 사용자의 편리성을 위해 지역은 전 지구, 동아시아, 호주, 호주 및 남태평양지역, 중동지역, 북아메리카, 러시아, 남아메리카, 남아시아로 구분하여 제공하지만, 사용자가 특정지역을 원한 때는 직접좌표 입력을 통해 자료와 변수를 추출하여 사용할 수 있도록 하였다.

Figure 83. Provides condition input function for region and variable extraction.

아래 화면은 MME 다중모델 앙상블 자료를 대상으로 월별 강수자료에 대해 결정론적 방법을 이용한 Anomaly 자료에 대해 동아시아 지역만 추출한 결과이다. MME 다중모델 앙상블 자료의 구성정보를 제공하기 위해 HindCast 기간 정보와 사용된 모델 종류, 예측에 대한 정보를 같이 제공하였다. 다음 화면은 호주지역 계절평균 해수면 온도를 추출하여 지도기반 시각화 한 예시이다.

Monthly Precipitation (2020/08)

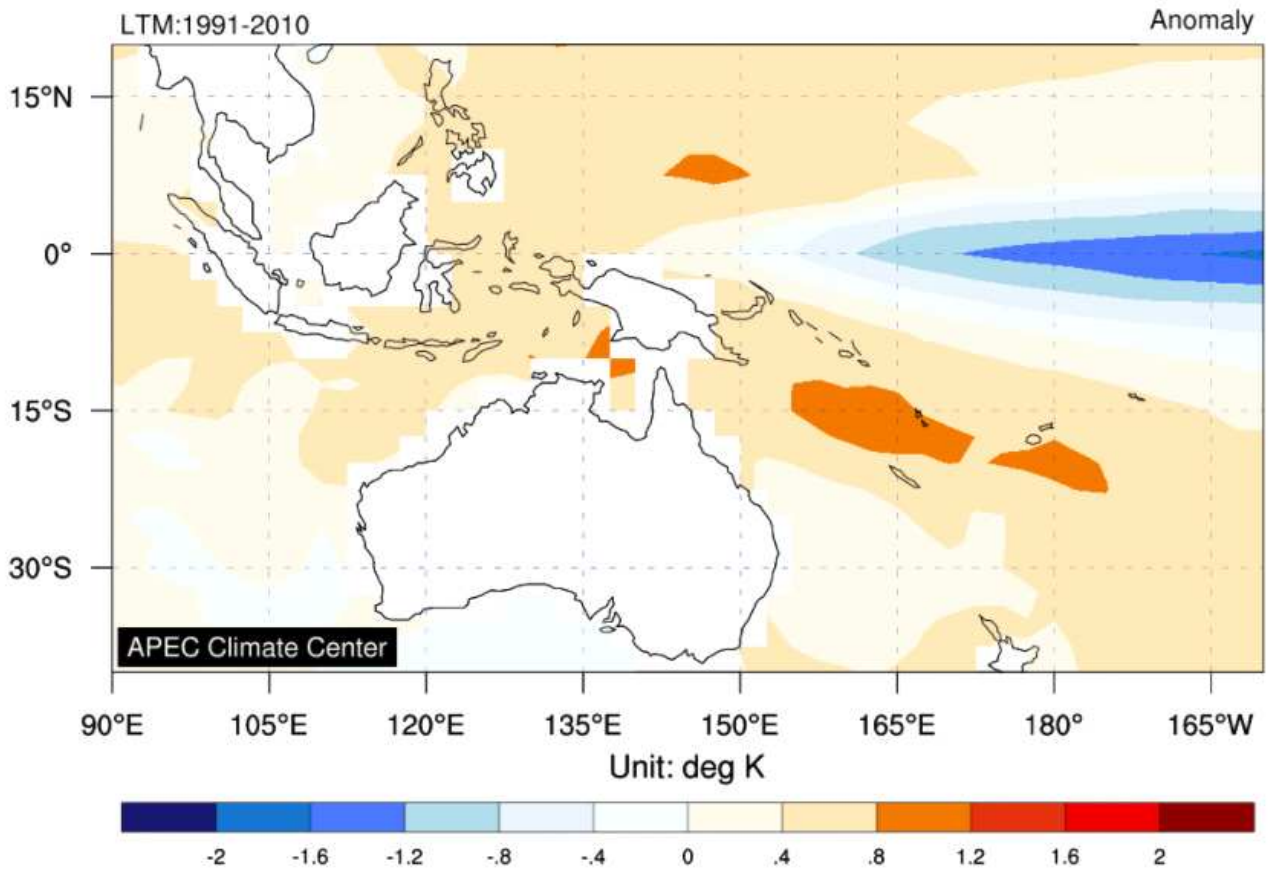


MME_MODELS : SCOPS,CSM1.1M,GLOSEA5GC2,CANSIPsv2,GEOS-S2S-2.1,CFSv2,CGCMv2.0

FORECAST_INFO : Monthly Forecast for ASONDJ (2020ASONDJ)

Figure 84. This is a visualized result by extracting precipitation data from East Asia.

Seasonal mean Sea Surface temperature (OND)



MME_MODELS : SCOPS,GLOSEA5GC2,CANSIPsv2,GEOS-S2S-2.1,CFSv2,CGCMv2.0

FORECAST_INFO : Seasonal Mean Forecast for OND (2020ONDJFM)

Figure 85. This is a visualized result by extracting Sea surface temperature data from Australia.

(라) 서비스를 위한 계절예측 자료 관리 프로그램 개발

기후자료처리 서비스는 데이터베이스를 사용하지 않고 정제된 파일만을 이용하여 자료처리 기능을 수행하도록 하였다. 그리하여 MME 현업에서 생산된 자료를 서비스용 스토리지로 동기화시키면서 오류파일을 점검하는 프로그램을 개발하였다. 전송된 파일의 무결성 검증이 완료되면 미리 정해진 규칙성을 이용하여 파일을 관리하고 누락된 파일은 재복사하여 최적의 파일구조 형태를 유지하도록 하였다. 누락된 파일은 시스템의 목록에서 제외시켜 데이터베이스를 수정하는 번거로움을 최소화하였다. 아래의 화면은 자료관리 프로그램 코드와 무결성 검증이 끝난 파일을 관리하는 예시이다.

```
curyear = datetime.datetime.now().year
curmonth = datetime.datetime.now().month
#curyear = 2015

if curmonth == 12:
    curyear = curyear +1

def search(dirname):
    try:
        filenames = os.listdir(dirname)
        Months = {'JAN':1, 'FEB':2, 'MAR':3, 'APR':4, 'MAY':5, 'JUN':6, 'JUL':7, 'AUG':8, 'SEP':9, 'OCT':10, 'NOV':11, 'DEC':12 } #APR 'AUG' DEC FEB JAN J
        Seasons= {'JFM':1, 'FMA':2, 'MAM':3, 'AMJ':4, 'MJJ':5, 'JJA':6, 'JAS':7, 'ASO':8, 'SON':9, 'OND':10, 'NDJ':11, 'DJF':12}
        #Lead = {'3-MON':2, '6-MON':5 }
        ex_list = ['olr', 'ts', 'u925', 'v925']

        print("filenames=", filenames)

        for filename in filenames:
            full_filename = os.path.join(dirname, filename)
            if os.path.isdir(full_filename):
                #print("full_filename----?", full_filename)
                search(full_filename)
            else:
                file_extension = os.path.splitext(full_filename)[-1] # split
                split_dir_file = os.path.split(full_filename) # separate directory and file
                path_dir = os.path.dirname(full_filename) # extract directory name
                split_each_dir_file = full_filename.split(os.path.sep) # separate each directory and file
                split_each_dir = path_dir.split(os.path.sep) # separate each directory
                ext4 = os.path.exists(full_filename) # check the file exists
                ext5 = os.path.isdir(full_filename) #check the path exists
                ext6 = os.path.isfile(full_filename) #check file path
                filesize = os.path.getsize(full_filename) #File size

                k = int(Months.get('JAN'))
                #print("z-->", str(Months.get('JAN'))+ "test")

                #print ('%02d' %k )
                if str(os.path.splitext(filename)[-2]) in ex_list:
                    continue

                if file_extension == ".nc":
                    nc_f = full_filename # Your filename
                    nc_fid = Dataset(nc_f, 'r') # Dataset is the class behavior to open the file and create an instance of the ncCDF4 class

                    # Extract data from NetCDF file
                    time = nc_fid.variables['time'][:]
                    print("leadtime= ", len(time), "-MON")
```

Figure 86. Development the data management for seasonal forecast and observation data

(마) 개별모델의 예측자료에 대한 편차 계산 프로그램 개발

APCC의 MME 참여기관의 개별모델에 대해 사용자가 원하는 기후기간을 선택하여 Anomaly를 계산하고 결과를 확인할 수 있는 시스템을 제공한다. 각 모델이 갖고 있는 여러 변수에 대해 평균값이나 평년값에 대한 Anomaly를 계산하여 Lead time별로 표출하는 서비스로 각 Lead time을 선택하면 팝업을 통해 확대된 자세한 지도기반 결과화면을 확인할 수 있다.

아래의 화면은 SCOPS 모델의 2020년도 6월에 예측한 강수량자료에 대해 1980년부터 2010년의 Hindcast를 이용하여 전지구의 강수량Anomaly를 계산한 결과와 Mean 평균값 계산 결과로 Leadtime 1인 2020년 7월을 선택하여 자세한 화면으로 확인한 예시이다.

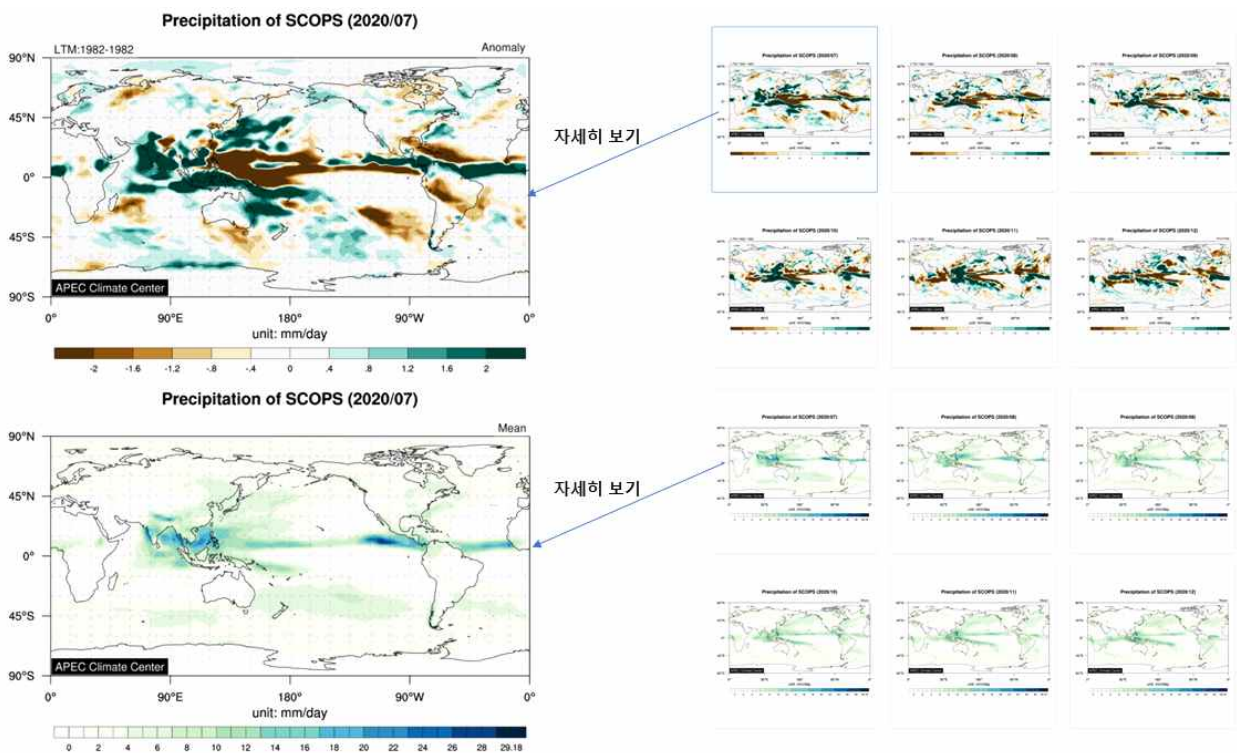


Figure 87. The results of calculating Anomaly and Mean for seasonal prediction data of 6-month multi-model ensemble.

(바) 가공자료의 지도기반 시각화 서비스 개발

기후자료처리 시스템의 처리된 자료에 대한 시각화 기능은 지도를 기반으로 각 Lead time 별로 강수, 해수면온도, 2m 온도 등 각 변수의 특성에 맞도록 Color map 과 범례를 조정하여 서비스를 제공한다. 추출된 자료의 특성에 따라 Anomaly는 APCC에서 정한 범원 내에서 표출이 되지만, Mean의 경우, 추출된 지역의 최솟값과 최댓값을 기준으로 지도에 표출되도록 하였다. 아래는 6개월 MME 자료와 6개월 개별모델 예측자료를 이용하여 태평양도서국 지역의 해수면온도와 전 지구 강수정보, 전 지구 강수 평균정보, 해면기압에 대해 표출된 예시이다.

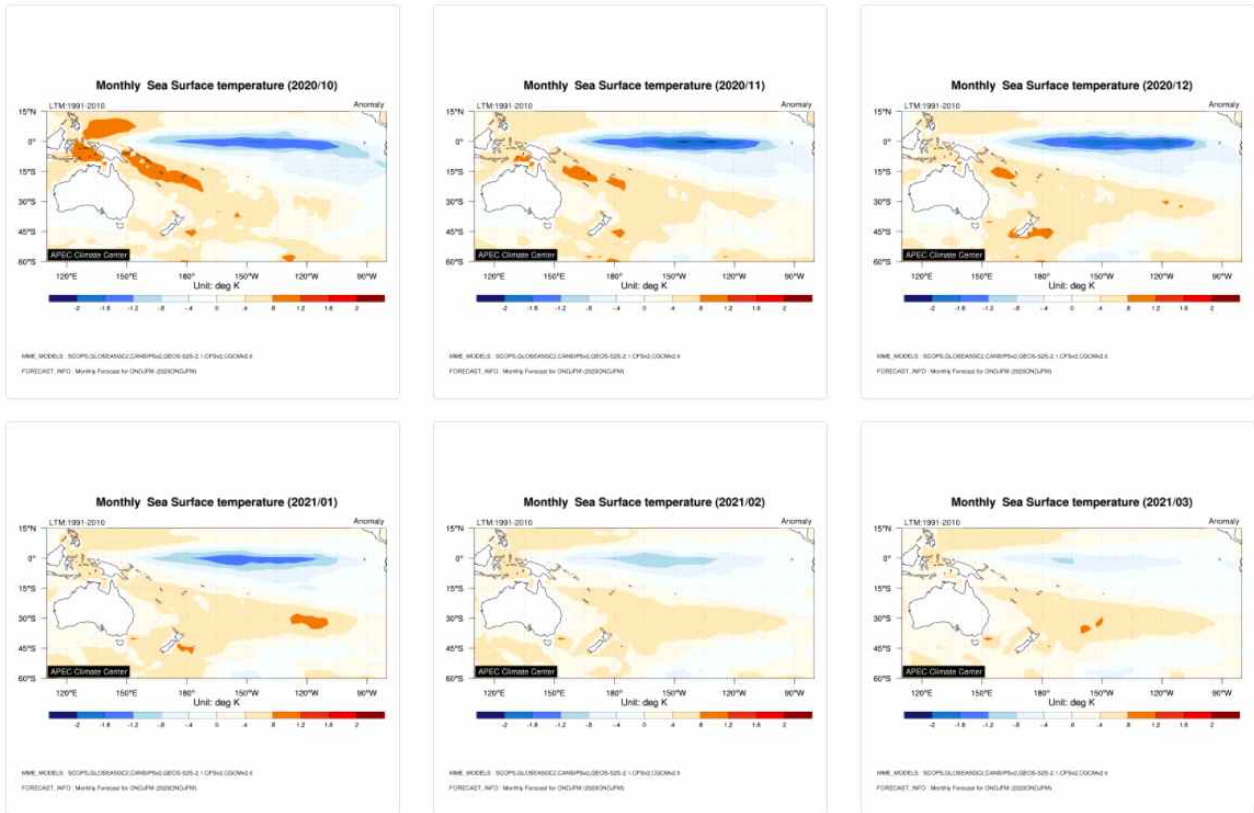


Figure 88. Visualized results Sea surface temperature from Australia and Pacific islands.

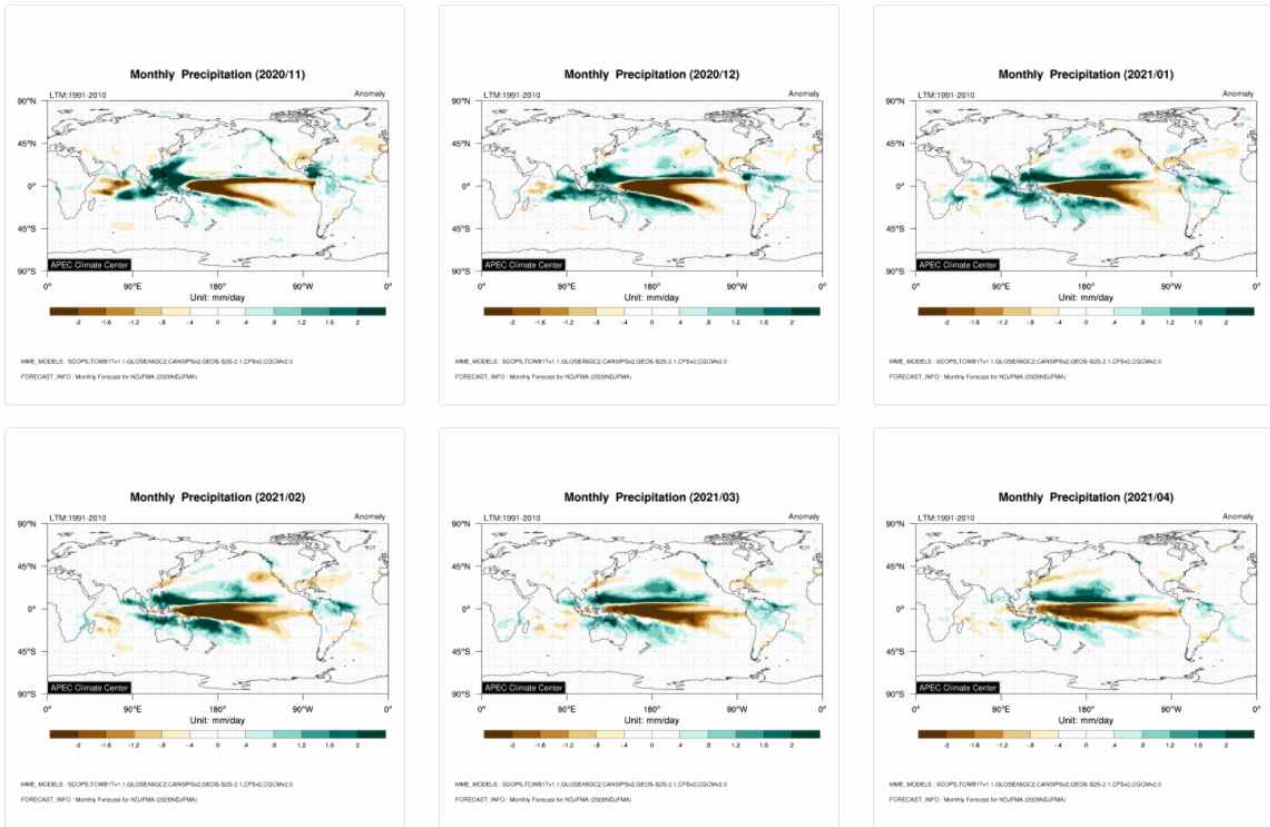


Figure 89. Visualized results precipitation from global.

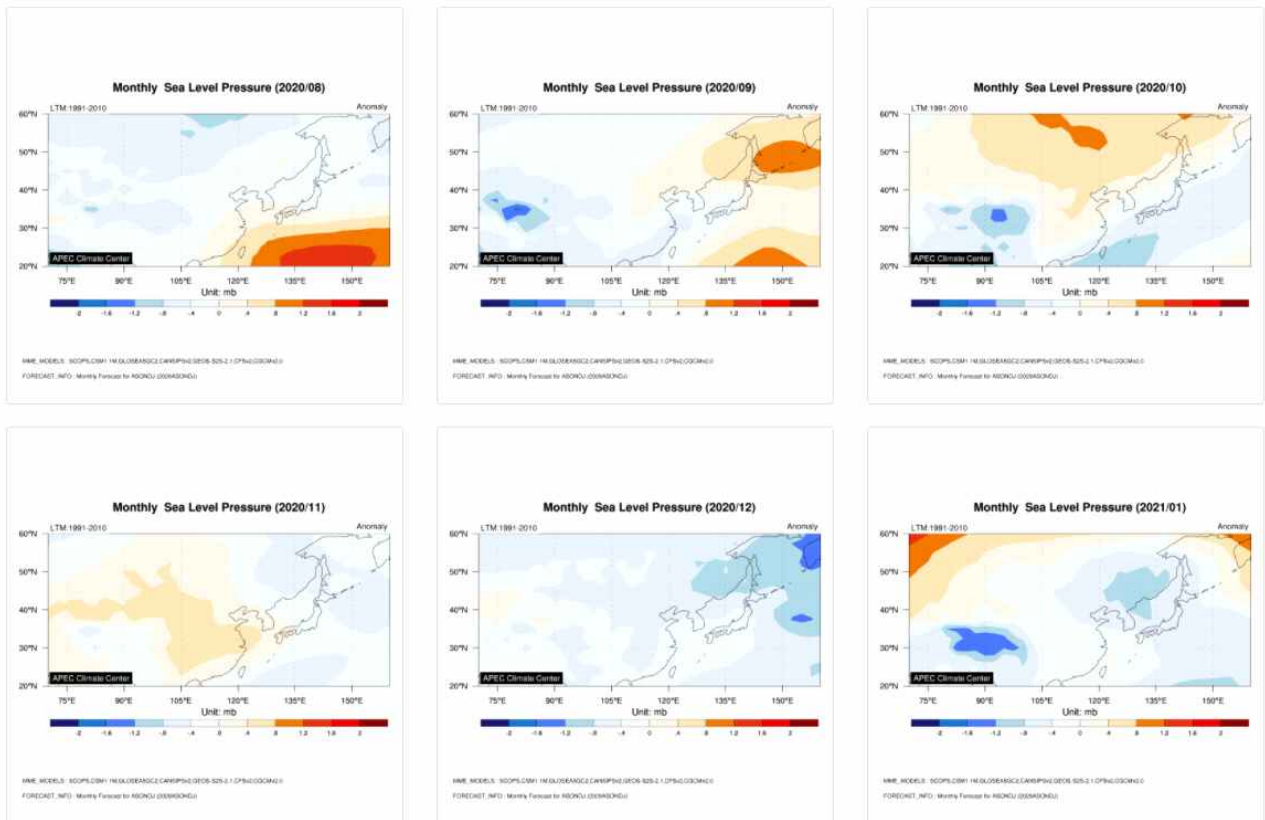


Figure 90. Visualized results Sea level pressure from East Asia.

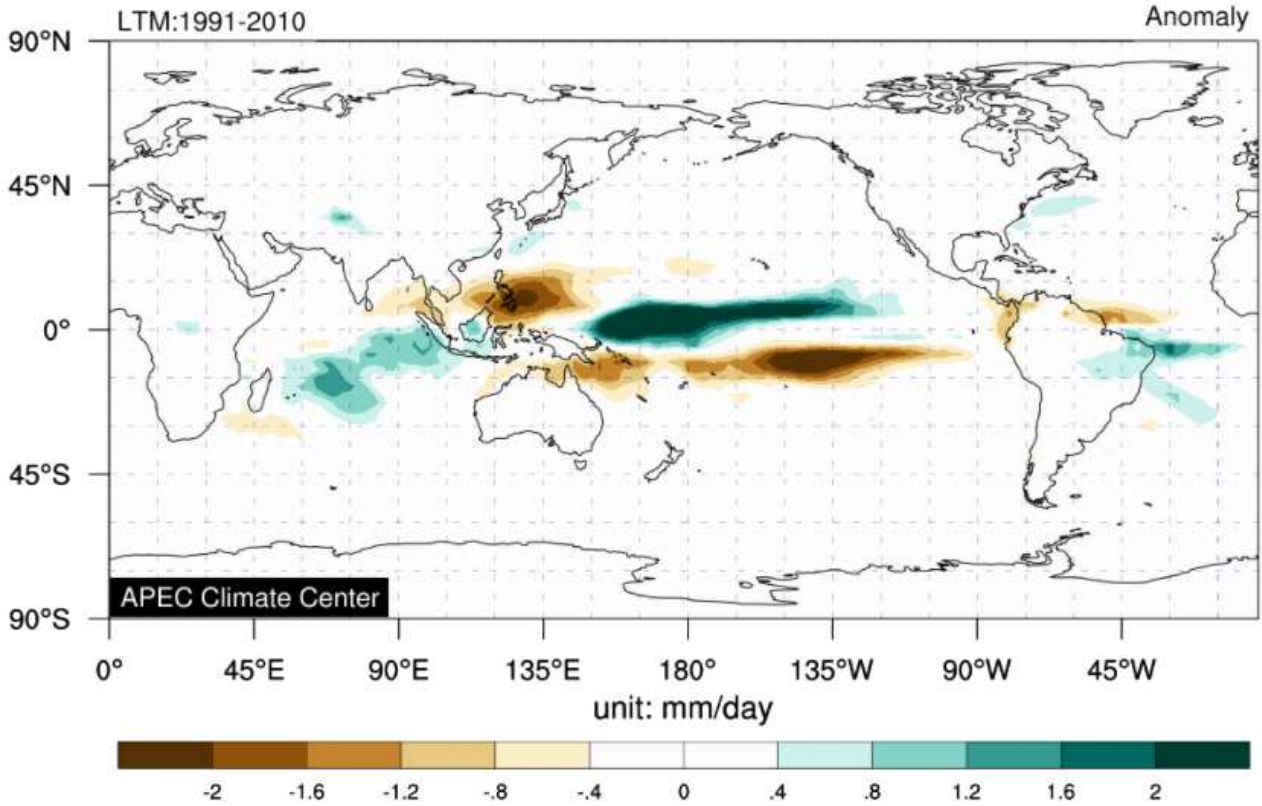
(사) 계절예측 자료 합성서비스 개발

APCC의 계절예측자료와 관측 자료를 비교할 수 있는 서비스를 제공하기 위해 사용자가 선택한 변수와 원하는 연도와 월을 선택하여 합성된 결과를 볼 수 있는 서비스를 개발하였다. 계절예측자료를 아노말리를 기본으로 정보를 제공하기 때문에 평균값이나 아노말리 값을 선택하지 않고 각 자료의 Lead time별로 합성할 수 있도록 기능을 제공하였다. 각 연도와 월을 대상으로 합성하는 기능을 제공하고 사용자의 조건을 추가하는 장바구니 기능을 제공하여 다양한 조건을 조합하여 합성해 볼 수 있도록 구성하였다.



Figure 91. It provides a cart function for the composite of seasonal forecast data

Precipitation_Prediction (Composite Map)



year:month:lead=['2019','2020'];['03','04'];['01']_Anomaly

Figure 92. Composite visualization result of Seasonal forecast precipitation data

(아) 관측 자료 합성서비스 개발

APCC의 계절예측자료와 관측 자료를 비교할 수 있는 서비스를 제공하기 위해 사용자가 선택한 변수와 원하는 연도 및 월을 선택하여 합성된 결과를 볼 수 있는 서비스를 개발하였다. 관측 자료의 합성은 예측자료와 비교를 위한 기능으로 평균값뿐만 아니라 사용자가 지정한 날짜의 연도를 평년값으로 아노말리를 계산하여 제공하고 사용자의 조건을 추가하는 장바구니 기능을 제공하여 다양한 조건을 조합하여 합성해 볼 수 있도록 구성하였다.

아래는 관측 자료를 이용하여 엘니뇨가 있던 해의 1988년, 1993년, 2016년의 7월, 8월 해수면온도 평균 합성결과와 1990년부터 2020년까지 평년값에 대한 아노말리 합성결과를 지도로 표출된 화면이다.

Monthly Mean of Sea Surface Temperature (Composite Map)

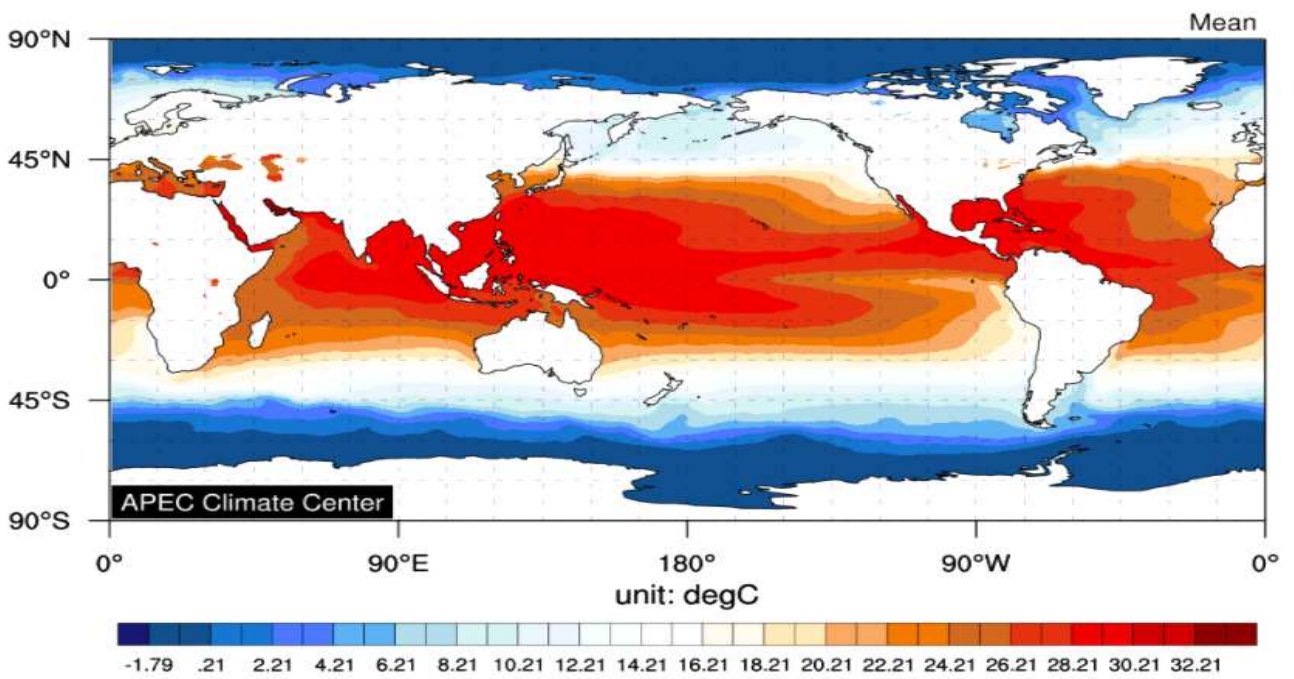


Figure 93. Mean composite visualization result of observation sea surface temperature data

Monthly Mean of Sea Surface Temperature (Composite Map)

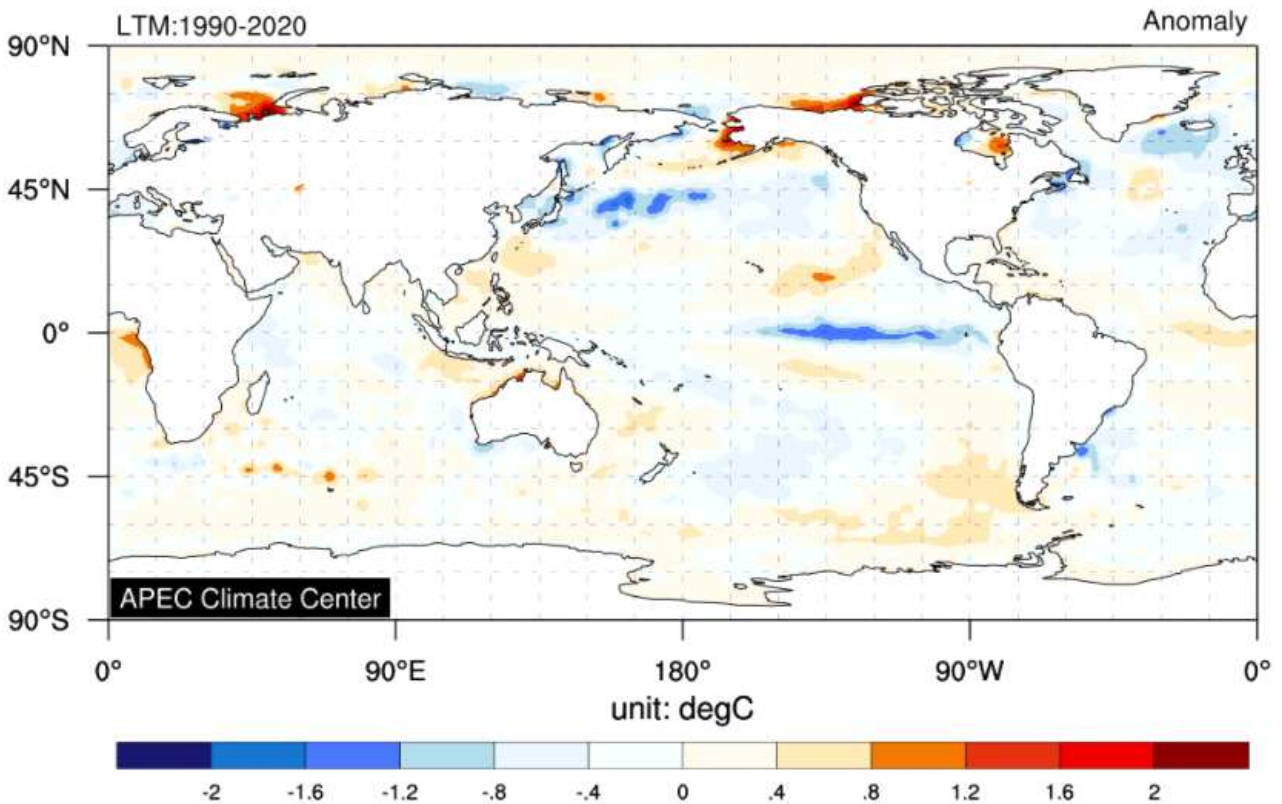


Figure 94. Anomaly composite visualization result of observation sea surface temperature data

(자) 가공자료의 파일포맷 변환 제공 서비스 개발

사용자들이 선택한 지역과 변수, 계산된 결과 값의 활용도 향상을 위해 추출된 자료를 파일형태로 내려 받기 기능을 제공하였다. 사용자는 Forecast를 선택하면 지도기반으로 제공된 결과를 파일형태로 내려 받을 수 있고, Hindcast의 기간을 선택하면 대상 변수의 Hindcast파일을 추출된 형태로 내려 받을 수 있다.

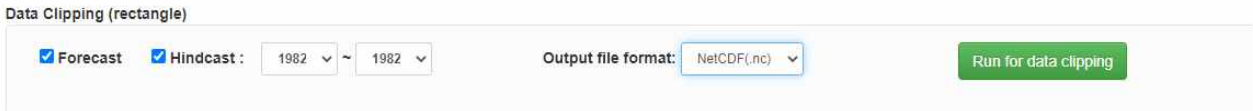


Figure 95. Target data and file formats input function.

NetCDF 파일에 익숙하지 않은 사용자를 위해 엑셀파일 포맷으로 자료를 내려 받을 수 있다. 아래의 자료를 추출하기 위한 조건을 입력하는 화면(상)과 파일형태로 내려 받은 파일목록과 엑셀파일에서 저장된 파일의 구조를 보여주는 예시이다.

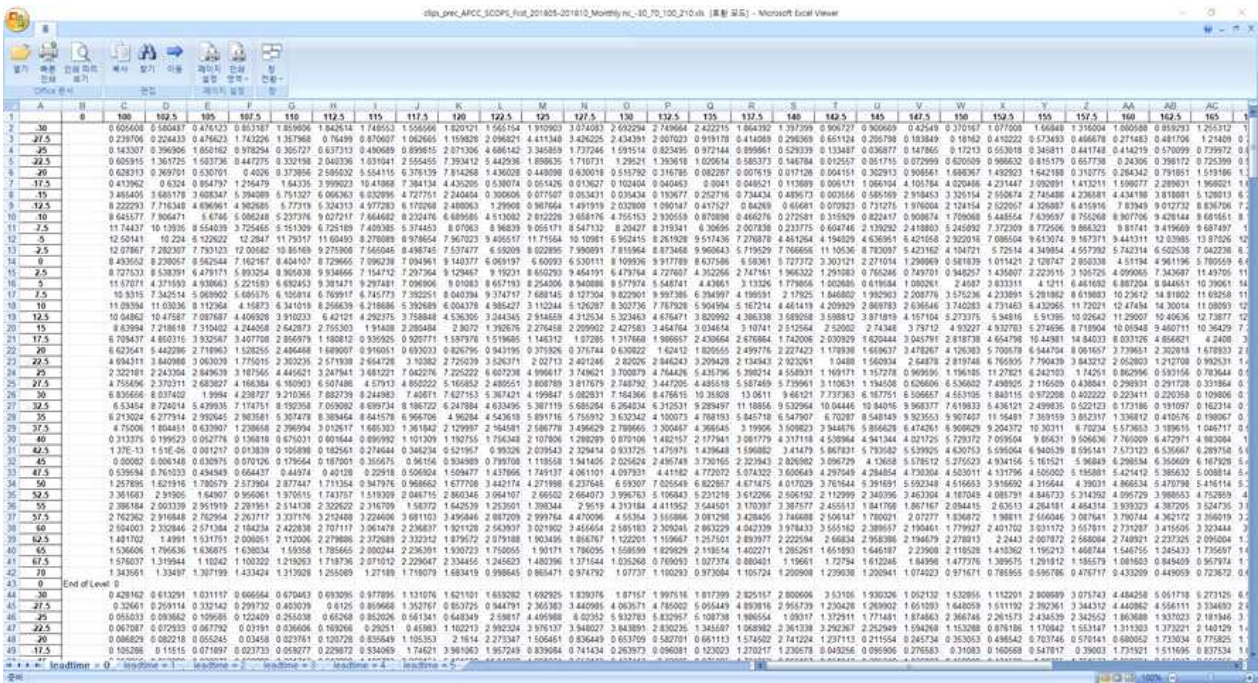


Figure 96. Data structure in Excel file format by coordinates , level, lead-time.

4. 기후예측시스템 연구개발허브 구축 및 기술지원

가. 연구개발허브 공동구축 및 운영

(1) 개요 및 필요성

현재 기후예측시스템 연구개발에서는 GC(Global Couple Model)개발 버전에 대한 대응이 늦은 편이며, 현업화 과정에서 코드 적용 결과 검증 등의 이유로 현업화 과정이 지연되는 문제점이 있다. 이런 문제점을 근본적으로 해결하기 위해 APEC 기후센터에서는 국립기상과학원과 함께 국내의 기후 예측 연구 활성화와 협업 개발 체계를 위한 연구개발허브(기후예측시스템 파일럿 현업화 시스템)를 공동구축하여 운영하기로 하였다. 해당 공동구축을 위해 이슈관리시스템인 TRAC 및 소스코드 형상관리시스템인 SVN/Git을 시험 설치 및 테스트하였으며, 향후 이슈관리시스템을 이용한 개발이력 추적 및 현황 파악을 하고, 소스코드 관리시스템을 활용하여 결합모델, 자립판 모형, 진단/분석/평가 툴 등 원시코드에 대한 관리를 수행할 목적으로 시험이 진행되었다. APCC 플랫폼 내부 가상머신에 시험적으로 버전관련 작업 및 설치를 수행 후 국립기상과학원에서 이관한 파일럿 현업화 서버 및 스토리지에 이관 설치하고 시범 운영을 실시하였다.

(2) 이슈관리시스템 (TRAC)

연구 개발시 사용되는 대표적인 이슈관리시스템인 Edgewall Software 사의 무료 소프트웨어인 TRAC은 버그, 요구사항, 작업 내용 등이 있을 때 해당 시스템에 게시물 형태로 티켓을 발행하여 게시하고 개발자, 테스터들이 작업 진행 상황을 기록하는 시스템을 기본으로 하여 사용법이 쉬운 편에 속하고, 버전 관리시스템인 Subversion(SVN) 등에 연동하여 사용 가능한 도구이다. 주요기능은 Wiki, 이슈(티켓) 관리, 마일스톤 관리이며, 다음과 같은 특징을 가진다.

- 프로젝트 진행시 티켓 발행을 통해 개발자/연구자간의 원활한 의사소통을 제공
- 프로젝트 진행상황을 마일스톤 및 시간이력 등을 통한 도식화 정보 제공
- 자체 Wiki 엔진을 통해 위키 문법을 지원하며 주요 프로젝트 페이지 링크 기능 제공
- Subversion과 연동하여 버전관리 기능 제공

유사한 도구로는 BugZilla, Mantis, FogBugz 등이 있으나, 가장 쉽게 적용가능하며, 기존 기상과학원에서 익숙하게 사용해온 TRAC을 사용하여 이슈관리시스템을 구성하도록 한다.

(가) Apache 설치

TRAC은 웹 서비스를 제공하기 위해서 자체 웹 구동을 위한 Stand alone 모드를 제공하고 있지만, 다양한 환경 지원과 확장성을 위해 웹 서버를 연동이 가능한 형태로 구성하고자 하였다. 이를 위해 먼저 Apache 설치를 Table 21과 같이 진행하였다.

Table 21. Apache Installation

```
//기본 패키지 설치
# yum -y groupinstall "Development Tools"

// 최신 버전의 httpd를 설치 (apxs 설치를 위해 httpd-devel 추가 설치)
# yum -y install httpd httpd-devel

// SELinux 해제
SELinux를 해제하지 않으면 나중에 8000 포트를 httpd에 연동할때 permission denied 발생
# vi /etc/selinux/config

// enforcing를 disabled로 수정
SELINUX=enforcing
--->
SELINUX=disabled

// 설정 완료 후에 재부팅
```

먼저 Apache를 구성하기 위한 기본적인 개발자 도구들을 ‘# yum -y groupinstall “Development Tools” ’ 명령어를 통해 group 설치를 진행한다. 해당 설치를 완료 후에 보안상의 문제가 최소화 될 수 있도록 최신의 웹 서버와 apxs 설치를 위한 개발자 도구를 설치하였다. 또한 웹서버 데몬 httpd와 포트 연동을 할 수 있도록 Security Enhanced Linux (SELinux) 설정을 기본설정인 ‘enforcing’ 에서 ‘disabled’ 로 변경하여 준다. 향후 보안에 관련하여 IPtables 등을 활용하여 해당 부분에 대한 대안을 마련할 수 있다.

(나) 형상관리 설치 및 환경설정

Table 22. Subversion Installation and Configuration

```
// Subversion 설치 (yum 이용)
# yum install -y subversion

// Subversion 환경 설정을 위한 기본 디렉토리 설정 (사용자 추가 포함)
# useradd esrd
# cd /home/esrd
# mkdir svn
# cd svn
# svnserve -d -r /home/esrd/svn/
```

```

# vi ~/.bash_profile

-----

export SVN_EDITOR=/usr/bin/

-----

// Subversion glosea 프로젝트 생성
# cd /home/esrd/svn/
# svnadmin create glosea
# ls glosea/conf/
authz    passwd    svnserv.conf

// Subversion glosea 프로젝트 환경 설정 (일반 설정)
# vi glosea/conf/svnserv.conf

-----시작-----

#### This file controls the configuration of the svnserve daemon, if you
#### use it to allow access to this repository. (If you only allow
#### access through http: and/or file: URLs, then this file is
#### irrelevant.)

#### Visit http://subversion.tigris.org/ for more information.

[general]
#### These options control access to the repository for unauthenticated
#### and authenticated users. Valid values are "write", "read",
#### and "none". The sample settings below are the defaults.
anon-access = none
auth-access = write
#### The password-db option controls the location of the password
#### database file. Unless you specify a path starting with a /,
#### the file's location is relative to the directory containing
#### this configuration file.
#### If SASL is enabled (see below), this file will NOT be used.
#### Uncomment the line below to use the default password file.
password-db = passwd
#### The authz-db option controls the location of the authorization
#### rules for path-based access control. Unless you specify a path

```

```
### starting with a /, the file's location is relative to the the
### directory containing this file. If you don't specify an
### authz-db, no path-based access control is done.
### Uncomment the line below to use the default authorization file.
authz-db = authz
### This option specifies the authentication realm of the repository.
### If two repositories have the same authentication realm, they should
### have the same password database, and vice versa. The default realm
### is repository's uuid.
realm = CICE Repository
```

```
[sas]
```

```
### This option specifies whether you want to use the Cyrus SASL
### library for authentication. Default is false.
### This section will be ignored if svnserve is not built with Cyrus
### SASL support; to check, run 'svnserve --version' and look for a line
### reading 'Cyrus SASL authentication is available.'
# use-sasl = true
### These options specify the desired strength of the security layer
### that you want SASL to provide. 0 means no encryption, 1 means
### integrity-checking only, values larger than 1 are correlated
### to the effective key length for encryption (e.g. 128 means 128-bit
### encryption). The values below are the defaults.
# min-encryption = 0
# max-encryption = 256
```

-----끝-----

```
// Subversion glosea 프로젝트 환경 설정(사용자)
```

```
# vi glosea/conf/passwd
```

-----시작-----

```
### This file is an example password file for svnserve.
### Its format is similar to that of svnserve.conf. As shown in the
### example below it contains one section labelled [users].
### The name and password for each user follow, one account per line.
```

```
[users]
```

```
# harry = harryssecret
```

```
# sally = sallysecret
```

```
test = test
```

-----끝-----

```
// Subversion glosea 프로젝트 환경 설정 (인증 및 권한)
```

```
# vi glosea/conf/authz
```

-----시작-----

```
### This file is an example authorization file for svnserve.
```

```
### Its format is identical to that of mod_authz_svn authorization
```

```
### files.
```

```
### As shown below each section defines authorizations for the path and
```

```
### (optional) repository specified by the section name.
```

```
### The authorizations follow. An authorization line can refer to:
```

```
### - a single user,
```

```
### - a group of users defined in a special [groups] section,
```

```
### - an alias defined in a special [aliases] section,
```

```
### - all authenticated users, using the '$authenticated' token,
```

```
### - only anonymous users, using the '$anonymous' token,
```

```
### - anyone, using the '*' wildcard.
```

```
###
```

```
### A match can be inverted by prefixing the rule with '~'. Rules can
```

```
### grant read ('r') access, read-write ('rw') access, or no access
```

```
### ('').
```

```
[aliases]
```

```
# joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, Ltd./OU=Research Institute/CN=Joe  
Average
```

```
[groups]
```

```
# harry_and_sally = harry,sally
```

```
# harry_sally_and_joe = harry,sally,&joe
```

```
glosea_group= test
```

```
# [/foo/bar]
```

```
# harry = rw
```

```
# &joe = r
```

```
# * =
```

```

# [repository:/baz/fuz]
# @harry_and_sally = rw
# * = r

// glosea_group은 read / write 권한, 전체는 read 권한만
[/]
@glosea_group = rw
* = r

```

-----끝-----

Table 22는 Subversion의 설치와 그에 대한 다양한 환경설정에 관하여 표기하고 있다. 먼저 yum 명령어를 이용하여 subversion을 설치하고 해당 환경을 설정하기 전 기본적으로 사용자 추가를 포함한 디렉토리 설정을 해준다. 해당 사용자의 명칭은 국립기상과학원과 협의를 통해 esrd를 사용하기로 하였으며 따라서 /home/esrd 디렉토리를 기본으로 하여 svnserve 디렉토리 설정을 해주고 SVN_EDITOR 경로 설정을 하였다. 현재 과학원에서 연구개발 중인 glosea 소스코드를 가져오기 위한 준비를 위해서 Subversion에 glosea 프로젝트를 생성하고 해당 프로젝트를 위한 일반적인 환경설정을 완료하였다. 일반 환경 설정에서는 anon-access를 none 셋팅하고 auth-access를 write로 만들어주는 것을 포함하여 패스워드 db, Authz-db, 저장소 고유 아이디를 표기하는 realm을 설정해준다.

gloase/conf/passwd를 편집하여 subversion glosea 프로젝트의 사용자 환경 설정을 진행하게 되는데 이를 통해 관리자 및 사용자를 등록하게 된다. 본 보고서에서는 실험을 위한 test 계정을 생성한 것을 살펴볼 수 있다. 해당 사용자 등록 완료 후에는 glosea 프로젝트에 대한 인증 및 권한을 설정을 위하여 authz 파일에 group 권한을 설정해 주었다.

(다) Apache와 Subversion 연동

웹 서버 Apache와 svn 연동을 위한 mod_dav_svn을 설치하고 /etc/httpd/modules/ 폴더에서 관련 공유 라이브러리들인 *.so 파일이 있는지 확인을 한다. svn 디렉토리에 대하여 Apache 계정이 접근할 수 있도록 권한 및 모드를 변경하여 준다. 그 후 실제 연동을 위한 환경을 설정하여 주기 위해 subversion.conf 파일을 생성하고 필요한 정보를 입력해 주었다. 해당 설정 부분은 Table 23의 상술된 내용을 참조하도록 한다.

Table 23. Apache and SVN Configuration

```

// Apache와 svn 연동을 위한 mod_dav_svn 설치
# yum -y install mod_dav_svn

// 설치 완료 후에 /etc/httpd/modules/ 폴더에서 아래 파일 확인

```

```

mod_dav_svn.so
mod_authz_svn.so
mod_dontdothat.so

// SVN root 디렉토리에 Apache 계정이 접근할 수 있도록 권한을 부여
# cd /home/esrd
# chown -R apache.apache svn/
# chmod -R 755 svn

// esrd 홈 디렉토리가 700 임
# chmod 755 /home/esrd

// 환경 설정
# vi /etc/httpd/conf.modules.d/10-subversion.conf

// 해당 파일 안에는 새로 추가된 .so 파일 세 개의 설정만 있음 (최초 파일 내용)
LoadModule dav_svn_module      modules/mod_dav_svn.so
LoadModule authz_svn_module    modules/mod_authz_svn.so
LoadModule dontdothat_module   modules/mod_dontdothat.so

// subversion.conf 설정 파일 생성
# vi /etc/httpd/conf.d/subversion.conf

<Location /svn>
    DAV svn
    SVNParentPath /home/esrd/svn

    <LimitExcept GET PROPFIND OPTIONS REPORT>
        AuthType Basic
        AuthName "Authorization Realm"
        AuthUserFile /home/esrd/trac/trac.htpasswd
        AuthzSVNAccessFile /etc/svn-access-control
        Require valid-user
    </LimitExcept>
</Location>

// 설정 파일 생성

```

```
# vi /etc/svn-access-control

[groups]
admin          = *****
glosea_group = @admin,rnd075,rnd081
[glosea:/]
* = r
@glosea_group = rw
```

(라) TRAC 설치

TRAC은 현재 1.4 버전이 출시되어 있으나, 플러그인 호환성 등을 담보하기 위해서는 1.2 버전으로 설치하고 실험하였다. 해당 버전의 설치를 위해서 TRAC의 기본이 되는 python 관련 도구들을 Table 24과 같이 ‘# sudo yum install python-setuptools python-setuptools-devel python-devel’ 명령어를 통하여 설치해 주었다. 해당 설치 완료 후 easy_install을 사용하여 —always-unzip 옵션을 추가 및 설치를 진행하고, tacd 1.2.3이 제대로 동작함을 살펴볼 수 있다. 마지막으로 trac 에 접근하기 위해 Linux 운영체제 내의 방화벽 옵션을 해제하는 것으로 trac 활용 준비는 마무리되었다.

Table 24. Trac Installation

```
// Trac 1.2.3 버전의 설치를 위하여 easy_install 필요
// easy_install 설치를 위한 관련 dependency 설치
# yum -y install python-setuptools python-setuptools-devel python-devel

// easy_install 설치 확인
# whereis easy_install
easy_install: /usr/bin/easy_install

// easy_install을 이용한 trac 1.2.3 설치
# /usr/bin/easy_install --always-unzip trac==1.2.3 [옵션 추가]

// trac 버전 확인
# tracd --version
tracd 1.2.3

// trac 접근을 위한 CentOS 방화벽 해제
# systemctl disable firewalld
# systemctl stop firewalld
```

(마) TRAC 자료 이관

설치 작업이 완료된 TRAC에 국립기상과학원에서 운영해오던 glosea 프로젝트를 이관하기 위하여 Table 25과 같이 수신한 백업자료를 서버로 전송하여 자료의 압축을 풀고 tracgs 템플릿을 이용하여 glosea 프로젝트를 생성하였다. 해당 작업을 수행 후에 정상동작 여부를 살펴보기 위해 단독 실행 모드인 standalone 모드로 백그라운드 실행을 한 다음 브라우저를 통해 확인을 실시하였다. 또한 htpasswd 설정을 통해 사용자 id와 password를 설정해주게 된다.

Table 25. Trac glosea project migration

```
// 기상과학원 trac 자료 복사
// 기상과학원 자료가 있는 곳으로 이동
# mv glosea_nims.tar.gz /home/esrd/
# mv tracgs_nims.tar.gz /home/esrd/

# cd /home/esrd
# tar xvfz tracgs_nims.tar.gz (tracgs 폴더 안에 압축풀림)
# mkdir trac
# cd trac

// 기상과학원 tracgs를 템플릿으로 glosea 프로젝트 생성
// 생성 후 trac/glosea 폴더가 생성
# trac-admin glosea initenv ₩
                --inherit=/home/esrd/tracgs/conf/trac.ini

Project Name [My Project]> GloSea
Database connection string [sqlite:db/trac.db]> 엔터

// 백업받은 파일 압축 해제 ( .glosea_bak 폴더에 압축 풀림)
# cd /home/esrd
# mv glosea_nims.tar.gz trac/

# cd /home/esrd/trac
# tar xvfz glosea_nims.tar.gz

// 압축 해제 된 파일을 /home/esrd/trac/glosea/ 경로에 덮어 씌우고 trac daemon 실행
// 원래 있던 glosea 폴더를 삭제하면 되는데, 백업을 원하는 경우 다른 폴더로 복사한다.
// 같은 폴더에 있게되면 프로젝트가 한개 더 생기게 된다.
# rm -rf glosea
```

```

# mv .glosea_bak glosea
# tracd --port 8000 /home/esrd/trac/glosea (standalone)
# nohup tracd --port 8000 /home/esrd/trac/glosea & (백그라운드 실행)

// htpasswd 설정
# htpasswd -cm /home/esrd/trac/trac.passwd username
New password:
Re-type new passwd:
Addin password for user username

```

(바) TRAC과 Apache 연동

TRAC과 Apache를 연동하기 위해서는 크게 두 가지의 작업을 거쳐야 한다. 먼저 Table 26과 같이 mod_python을 설치하고, 10-python.conf에 Module Load 부분을 삽입해주고, httpd.conf 설정파일에 Port 번호를 추가 및 home 디렉터리 권한을 변경하여 준다.

Table 26. Trac and Apache Configuration

```

// mod_python 설치

# cd /usr/local/src
# wget http://repo.iotti.biz/CentOS/7/noarch/lux-release-7-1.noarch.rpm
# rpm -Uvh lux-release-7-1.noarch.rpm
# yum -y install mod_python

# vi /etc/httpd/conf.modules.d/10-python.conf

//추가된 10-python.conf 파일 내용
LoadModule python_module modules/mod_python.so

// httpd.conf 설정
# vi /etc/httpd/conf/httpd.conf

// 추가
Listen 80
Listen 8000

// home 디렉토리 권한 변경
# cd /home/esrd
# chown apache.apache -R trac

```

Python을 기반으로 작동하는 TRAC을 Apache와 연동하여 주기 위해 Table 27과 같이 Python 관련 설정도 마무리해 준다.

Table 27. Python Configuration

```
# vi /etc/httpd/conf.d/python.conf (원본)
#
# Mod_python is a module that embeds the Python language interpreter
# within the server, allowing Apache handlers to be written in Python.
#
# Override type-map handler for /var/www/manual
<Directory "/var/www/manual/mod/mod_python">
    <Files *.html>
        SetHandler default-handler
    </Files>
</Directory>
# This will cause files beneath /var/www/html with the extension .spam
# to be handled by the Python script /var/www/html/eggs.py
#
#<Directory /var/www/html>
#    AddHandler python-program .spam
#    PythonHandler eggs
#</Directory>
# This will cause all requests to the /python heirachy of your
# webserver to be handled by the python script /path/to/myhandler.py
#
#<Location /python>
#    SetHandler python-program
#    PythonPath "sys.path + ['/path/to']"
#    PythonHandler myhandler
#</Location>
# This will cause all requests to the /python hierachy of your
# webserver to be handled by mod_python's Publisher handler
# (see http://localhost/manual/mod/mod_python/hand-pub.html)
```

```

#
#<Location /python>
#   SetHandler python-program
#   PythonHandler mod_python.publisher
#</Location>

# This will cause the output of all requests to files beneath
# /var/www/html with the extension .flt to be filtered through
# the Python script /var/www/html/filter.py
#
#<Directory /var/www/html>
#   PythonOutputFilter filter MYFILTER
#   AddOutputFilter MYFILTER .flt
#</Directory>

# This will enable use of "Python Server Pages", HTML pages with
# inline Python code which is interpreted by the server:
#
#<Directory /var/www/html>
#   AddHandler mod_python .psp
#   PythonHandler mod_python.psp
#</Directory>

```

원본에서 추가 및 수정된 부분만 빨간색

```

#
# Mod_python is a module that embeds the Python language interpreter
# within the server, allowing Apache handlers to be written in Python.
#

LoadModule python_module modules/mod_python.so

ServerName localhost
<VirtualHost *:8000>
#   DocumentRoot /var/www/html
#   PythonOption TracEnvParentDir /home/esrd/trac
#   <Location />
#       SetHandler mod_python
#       PythonInterpreter main_interpreter
#       PythonHandler trac.web.modpython_frontend
#       PythonOption TracUriRoot /

```

```

        <IfModule mod_authz_core.c>
            Require all granted
        </IfModule>
    </Location>
    <LocationMatch '[^/]+/login'>
#       AuthType Basic
       AuthName "Trac"
#       AuthUserFile /home/esrd/trac/trac.htpasswd
#       Require valid-user
    </LocationMatch>

Alias /css /home/esrd/tracgs/htdocs/common/css

<Location /css>
    SetHandler None
</Location>

</VirtualHost>

##### Alias /trac/chrome/common
/home/esrd/tracgs/htdocs/common
##### <Directory "/home/esrd/tracgs/htdocs">
##### # For Apache 2.2
##### <IfModule !mod_authz_core.c>
##### Order allow,deny
##### Allow from all
##### </IfModule>
##### # For Apache 2.4
##### <IfModule mod_authz_core.c>
##### Require all granted
##### </IfModule>
##### </Directory>
##### <Location "/trac/chrome/common">
##### SetHandler None
##### </Location>

# Override type-map handler for /var/www/manual
#<Directory "/var/www/manual/mod/mod_python">

```

```

# <Files *.html>
# SetHandler default-handler
# </Files>
#</Directory>

# This will cause files beneath /var/www/html with the extension .spam
# to be handled by the Python script /var/www/html/eggs.py
#
#<Directory /var/www/html>
# AddHandler python-program .spam
# PythonHandler eggs
#</Directory>

# This will cause all requests to the /python heirachy of your
# webserver to be handled by the python script /path/to/myhandler.py
#
#<Location /python>
# SetHandler python-program
# PythonPath "sys.path + ['/path/to']"
# PythonHandler myhandler
#</Location>

# This will cause all requests to the /python hierachy of your
# webserver to be handled by mod_python's Publisher handler
# (see http://localhost/manual/mod/mod\_python/hand-pub.html)
#
#<Location /python>
# SetHandler python-program
# PythonHandler mod_python.publisher
#</Location>

# This will cause the output of all requests to files beneath
# /var/www/html with the extension .flt to be filtered through
# the Python script /var/www/html/filter.py
#
#<Directory /var/www/html>
# PythonOutputFilter filter MYFILTER
# AddOutputFilter MYFILTER .flt

```

```
#</Directory>

# This will enable use of "Python Server Pages", HTML pages with
# inline Python code which is interpreted by the server:
#
#<Directory /var/www/html>
#   AddHandler mod_python .psp
#   PythonHandler mod_python.psp
#</Directory>
```

마지막으로 아래와 같이 환경변수를 추가하면 Trac과 Apache의 연동을 완료할 수 있게 된다.

Table 28. Trac and Apache Configuration

```
# vi /etc/profile
export PKG_RESOURCES_CACHE_ZIP_MANIFESTS=1
```

(사) Plug-in 설치

연구개발허브 구축을 위한 Trac 설정의 마지막 단계는 플러그인을 설치하여 좀 더 Trac을 활용하기 편리하도록 만들어 주는 것이다. 이런 플러그인들은 아래 Table 29,30,31과 같이 크게 필수 플러그인, 유용한 플러그인, 기타 플러그인 이렇게 3종류로 나누어 볼 수 있다.

Table 29. Essential Plug-in list for Trac

프로그램	목적
accountmanagerplugin 0.5.0	Account Manager
trac-subtickets-plugin 0.5.4dev	Subtickets support for Trac tickets
masterticketsplugin 4.0.2	Add support for ticket dependencies to Trac
timingandestimationplugin 1.6.0b	Estimation and Time Tracking
ganttcalendarplugin 0.6.4	Ticket-based Gantt chart and calendar
tracjsganttpugin 1.2.0.0dev	Display a jsGantt chart in a wiki page
Pygments 1.6	Code Highlighting

Table 30. Useful Plug-in list for Trac

프로그램	목적
sectionEditplugin 1.2.0	Edit a single wiki section
autocompleteusersplugin 0.4.7dev	Autocomplete user fields (assigned and CC) on tickets
cloneTicketPlugin	Add a "Clone" button tickets
tracmathplugin 0.6	Support for LaTeX Math Formulas in wiki pages
UserList.py	List of users

LastModified.py	Last modified wiki page
-----------------	-------------------------

Table 31. Auxiliary Plug-in list for Trac

프로그램	목적
extendedversionplugin 1.0dev	Extending the concept of a 'version' in Trac
downloadsplugin 1.0.0dev	Adds a downloads section to Trac
fullblogplugin 0.1.6.3	Add s self-contained blog to Trac
noteboxmacro 1.0	Note boxes with icons
trac-emoji 0.5	Emoji plugin for trac implementing
ticketmoverplugin 0.1.3	Move Trac tickets from one project to another
LibreOffice 4.4.1.1	Preview PDF files
graphvizplugin 1.0.0.9dev	Graphviz for dynamic generation of diagrams
tracdragdrop 0.12.0.14	Drag-and-Drop attachments in Trac
tracwysiwygplugin 0.12.0.7	Track Wiki WYSIWYG Editor
onsitenotifications 1.0	On-site notifications
tracmenusplugin 0.3.0dev	Hierarchica navigation menus

상기 플러그인들 중 우선적으로 필수적인 플러그인에 대하여 설치를 수행하였으며 먼저 계정관리를 지원해주는 Account Manager의 설치 및 구성은 Table 32과 같이 진행된다.

Table 32. Install account manager plug-in

```
// accountmanagerplugin 0.5.0 설치
// 참고 사이트 [ https://trac-hacks.org/wiki/AccountManagerPlugin]
# cd /usr/local/src
# pip install svn+https://trac-hacks.org/svn/accountmanagerplugin/tags/acct\_mgr-0.5.0

// plugin 활성화 (이관 프로젝트 내 활성화 완료)
# vi /home/esrd/tracgs/conf/trac.ini
```

Ticket 활용을 도와주는 필수 플러그인은 trac-subtickets-plugin과 masterticecktsplugin 두 종류가 있는데 각각 보조 Ticket 지원, Ticket 의존성 지원기능을 제공해 주는 플러그인으로 그 설치와 구성은 Table 33, 34과 같다.

Table 33. Install subtickets plug-in

```
// trac-subtickets-plugin 0.5.4dev 설치
// 참고 사이트 [ https://trac-hacks.org/wiki/SubticketsPlugin]
//      참고      사이트에서      다운로드      [다운로드      링크      :
https://github.com/trac-hacks/trac-subtickets-plugin/zipball/master]
// 다운로드 받은 파일을 서버로 옮김
# mv /root/trac-hacks-trac-subtickets-plugin-0.5.4-4-g8ab0c1d.zip /usr/local/src/
# unzip trac-hacks-trac-subtickets-plugin-0.5.4-4-g8ab0c1d.zip
# cd trac-hacks-trac-subtickets-plugin-0.5.4-4-g8ab0c1d
# python setup.py bdist_egg
```

```
// 기능 활성화
# vi /home/esrd/tracgs/conf/trac.ini

[components]
tracsubtickets.api.* = enabled
tracsubtickets.web_ui.* = enabled
```

Table 34. Install masterticket plug-in

```
// masterticketsplugin 4.0.2 설치
// 참고 사이트 [ https://trac-hacks.org/wiki/MasterTicketsPlugin]

# pip install svn+https://trac-hacks.org/svn/masterticketsplugin/trunk/

// 기능 활성화 (참고 사이트에 있는 내용)
# vi /home/esrd/tracgs/conf/trac.ini

[components]
mastertickets.* = enabled

[mastertickets]
dot_path = /opt/local/bin/dot

** 이관된 trac.ini 설정 파일에는 해당 플러그인이 disabled 되어 있음.

[components]
mastertickets.* = disabled

[mastertickets]
dot_path = /usr/lib/python2.6/site-packages/graphviz
/usr/lib/python2.7/site-packages/graphviz
```

Trac 활용시의 시간 예상 및 이에 대한 추적 기능을 제공해 주는 플러그인은 직접 다운로드 받아서 서버에 복사해 주는 형태로 설치가 끝나며, 그에 대한 설치 과정 및 설정은 아래 Table 35와 같다.

Table 35. Install Timing and estimation plug-in

```
//timingandestimationplugin 1.6.0b 설치
// 참고 사이트 [ https://trac-hacks.org/wiki/TimingAndEstimationPlugin]

// 직접 zip 파일을 다운 받아서 서버로 복사해서 설치 진행

# cd /usr/local/src
# unzip trac0.12-17953.zip
# cd trac0.12
# python setup.py bdist_egg

// 기능 활성화 (참고 사이트에 있는 내용)
# vi /home/esrd/tracgs/conf/trac.ini

[components]
timingandestimationplugin.* = enabled

** 이관 대상 trac.ini 설정 파일에는 해당 플러그인이 disabled

[components]
timingandestimationplugin.* = disabled
```

Trac에서 달력관련 기능과 더불어 프로젝트 관리를 위한 차트인 Gantt Chart를 활용할 수 있도록 해주는 플러그인이 있으며 해당 플러그인은 아래 Table 36과 같이 설치를 진행하고 trac.ini 설정 파일을 수정 후에 사용가능한 형태가 된다. 또한 이를 Trac내 Wiki 페이지에 표출해 주기 위한 Java script 플러그인은 Table 37과 같이 설치 및 설정하여 연구과제 관리를 위한 차트로 Wiki내에서 활용할 수 있도록 구성하였다.

Table 36. Install Gantt and calendar plug-in

```
// ganttcalendarplugin 0.6.4 설치
// 참고사이트 [ https://trac-hacks.org/wiki/GanttCalendarPlugin]

# cd /usr/local/src
# mkdir ganttcalendar
# cd ganttcalendar
# svn export http://svn.osdn.jp/svnroot/shibuya-trac/plugins/ganttcalendarplugin/trunk
# cd trunk
# python setup.py bdist_egg
# easy_install dist/*.egg
```

```
// 기능 활성화 (참고 사이트에 있는 내용)
```

```
# vi /home/esrd/tracgs/conf/trac.ini
```

```
[components]
```

```
ganttcalendar.admin.holidayadminpanel = enabled
```

```
ganttcalendar.complete_by_close.completeticketobserver = enabled
```

```
ganttcalendar.ticketcalendar.ticketcalendarplugin = enabled
```

```
ganttcalendar.ticketganttticketgantttchartplugin = enabled
```

```
ganttcalendar.ticketvalidator.ticketvalidator = enabled
```

```
[mainnav]
```

```
ticketcalendar.label = Ticket Calendar
```

```
ticketganttt.label = Ticket Gantt
```

```
[ticket-custom]
```

```
complete = select
```

```
complete.label = Completed [%]
```

```
complete.options = |0|5|10|15|20|25|30|35|40|45|50|55|60|65|70|75|80|85|90|95|100
```

```
complete.order = 3
```

```
due_assign = text
```

```
due_assign.label = Start (YYYY-MM-DD)
```

```
due_assign.order = 1
```

```
due_close = text
```

```
due_close.label = End (YYYY-MM-DD)
```

```
due_close.order = 2
```

```
[ganttcalendar]
```

```
complete_conditions = fixed, invalid
```

```
default_zoom_mode = 3
```

```
first_day = 0
```

```
format = %Y-%m-%d
```

```
show_ticket_summary = false
```

```
show_weekly_view = false
```

```
[ganttcalendar]
```

```
complete_conditions = fixed, invalid
```

```
default_zoom_mode = 3
```

```
first_day = 0
```

```
format = %Y-%m-%d
```

```
show_ticket_summary = false
```

```
show_weekly_view = false
```

**** 실제 trac.ini 파일안의 설정 내용**

```
[components]
```

```
ganttcalendar.admin.holidayadminpanel = enabled
```

```
ganttcalendar.complete_by_close.completeticketobserver = enabled
```

```
ganttcalendar.ticketcalendar.ticketcalendarplugin = enabled
```

```
ganttcalendar.ticketganttticketgantttchartplugin = enabled
```

```
ganttcalendar.ticketvalidator.ticketvalidator = enabled
```

```
[mainnav]
```

```
ticketcalendar.label = Ticket Calendar
```

```
ticketganttt.label = Ticket Gantt
```

```
[ticket-custom]
```

```
complete = select
```

```
complete.label = Completed [%]
```

```
complete.options = |0|5|10|15|20|25|30|35|40|45|50|55|60|65|70|75|80|85|90|95|100
```

```
complete.order = 3
```

```
due_assign = text
```

```
due_assign.label = Start (MM/DD/YYYY)
```

```
due_assign.order = 1
```

```
due_close = text
```

```
due_close.label = End (MM/DD/YYYY)
```

```
due_close.order = 2
```

```
[ganttcalendar]
```

```
complete_conditions = fixed, invalid
```

```
default_zoom_mode = 3
```

```
first_day = 0
```

```
format = %m/%d/%Y
```

```
show_ticket_summary = false
```

```
show_weekly_view = false
```

Table 37. Install jsGantt chart plug-in

```
// tracjsganttplugin 1.2.0.0dev 설치
// 참고 사이트 [ https://trac-hacks.org/wiki/TracJsGanttPlugin]

# pip install svn+https://trac-hacks.org/svn/tracjsganttplugin/1.2/

// 기능 활성화
# vi /home/esrd/tracgs/conf/trac.ini
[components]
tracjsgantt.* = enabled
```

마지막 필수 플러그인은 소스코드의 Syntax highlighting을 제공해 주는 플러그인으로 연구개발허브에서 공유되는 기후예측모델 코드에 대하여 좀 더 가독성 높은 결과를 제공할 수 있는 플러그인이다.

Table 38. Install code highlighting plug-in

```
// Pygments 1.6 설치

// pip를 이용해서 설치를 진행한다.
# pip install Pygments==1.6
```

필수 플러그인이외에도 유용하게 생각하는 플러그인들도 설치를 진행하였으며, 해당 플러그인은 향후 정식 운영 및 사용자 피드백을 거치면서 필요한 조합으로 유연하게 설치 및 구성하도록 한다.

(아) 연구개발허브 시범 운영실시

연구개발허브 구축을 위한 이슈관리시스템(TRAC)과 소스코드 관리시스템(SVN)을 상기와 같이 설치 및 설정을 완료하고, 국립기상과학원에서 TRAC 및 Wiki 백업자료를 이관설치 및 테스트를 완료하였다. 해당 시스템의 이슈관리시스템을 이용하여 개발이력 추적 및 현황 파악을 더욱 용이하게 할 수 있고, 소스코드 관리시스템을 활용한 결합모델, 자립판 모형, 진단/분석/평가 등 원시코드 관리의 기반을 마련하였다. 해당 시스템의 제원은 Table 39에 나타나 있는 것과 같으며, 소스코드 관리를 위한 리파지터리 구성은 Table 40과 같다.

Table 39. Configuration of Pilot Operational Server

대상	서버('15년 도입)	스토리지('12년 도입)
제원	CPU: Intel Xeon 28코어 메모리: 128 GB 스토리지: 4.3 TB ※ 소스코드 공유, 개발이력 추적 등	스토리지: 4.0 PB(예상) ※ 민감도 실험자료 공유 등

Table 40. Configuration of Repository for R&D pilot system

저장소	glosea
디렉토리 구성	하위 디렉토리 : ancil, coupler, gcom, glosea, oper, papers, utils 각 디렉토리 하위에 tags, trunk, branches/dev, branches/test 생성
권한	admin_group, glosea_admin, glosea_group, anonymous

상기 기술된 여러 설치와 설정을 통해 구성된 해당 시스템은 Figure 97에 나타난 것처럼 웹 기반의 TRAC이 잘 운영되는 것을 살펴볼 수 있다.

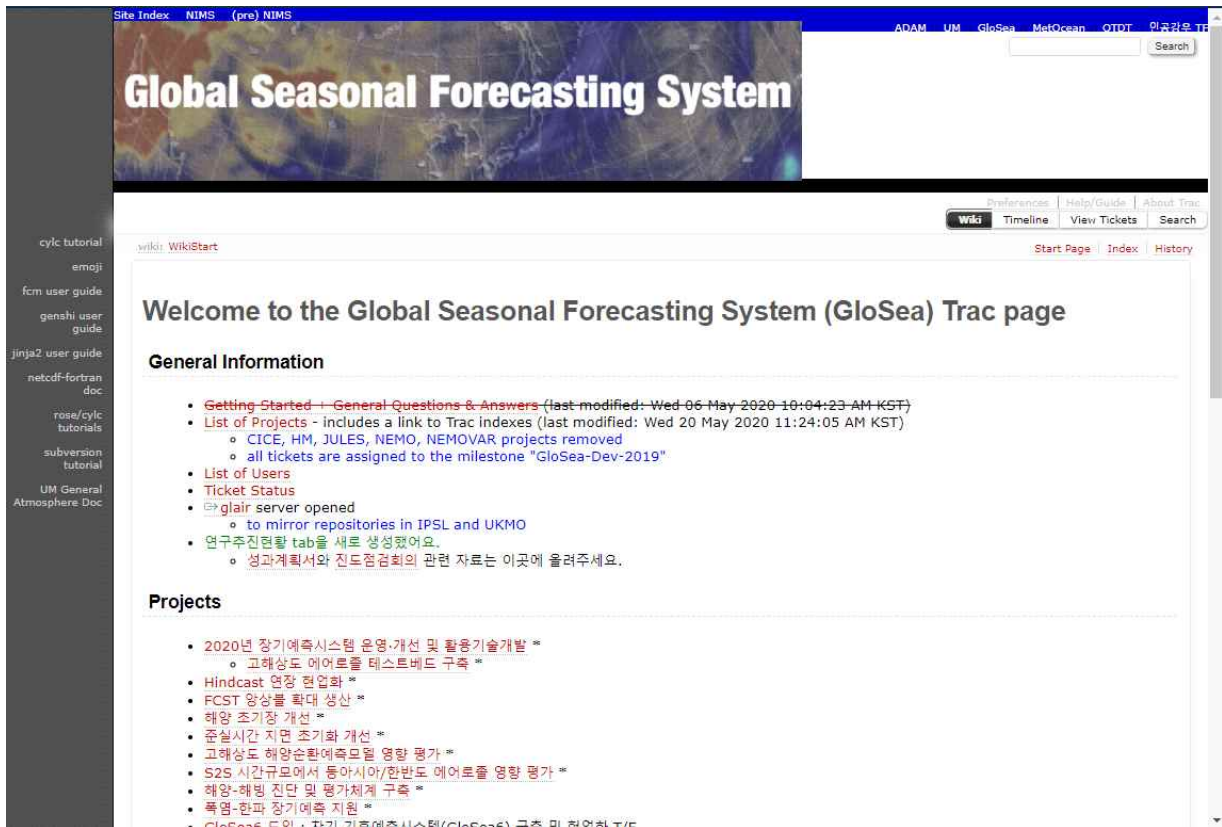


Figure 97. R&D Hub (Climate Prediction System Pilot operational server)

나. 연구개발허브 구축을 위한 기술지원

(1) 개요 및 필요성

기후예측시스템 연구개발허브 구축을 위한 기후예측 파일럿 시스템을 국립기상과학원과 공동으로 구축하였다. 해당 허브 Pilot 시스템 구축을 위한 장비(국립기상과학원 자산 - 서버 1조, 스토리지 18조)를 APCC로 이전, 설치를 위한 전산실 환경개선 사업을 추진하였으며, 연구개발허브 구축용 서버는 APCC 보유 캐비닛에 우선 설치하였으며, 네트워크 설정 및 기본 운영

체제, 시스템소프트웨어 설치에 대한 기술지원을 실시하였다. 연구개발허브 구축용 스토리지는 전산실 환경개선으로 수행된 전기, 공조 설비 확충에 대한 지원사업 후속으로 전산실 상면공간에 비치하고 서버와 연동을 완료하였다. 본 절에서는 연구개발허브 공동구축을 위한 기술지원에 대하여 기술하고자 한다.

(2) APCC 전산실 장비 배치 및 기술지원

기후예측시스템 연구개발허브 구축을 위한 기후예측 파일럿 시스템을 국립기상과학원과 공동으로 구축하였다. 해당 허브 Pilot 시스템 구축을 위한 장비(국립기상과학원 자산 - 서버 1조, 스토리지 18조)를 APCC로 이전, 설치를 위한 전산실 환경개선 사업을 추진하였으며, 네트워크 및 소프트웨어 설정 등 기술지원을 실시하였다.

먼저 대규모 스토리지 이전을 위한 필요 사용 환경인 전력, 상면공간 등을 면밀히 파악하기 위해 다수의 업무협의 및 국가기상슈퍼컴퓨터 센터 실사 방문을 실시하였으며, 기후예측시스템 파일럿 스토리지 설치를 위한 APCC 전산실 환경개선 사업을 기획하였으며, 기후예측시스템 파일럿 현업화 서버 소프트웨어/라이브러리 설치 및 구성을 수행하였다. 또한 해당 시스템에 대하여 APCC 전산실 상면공간내 배치를 구성하고 상호연결 방안 및 네트워크 구축 등에 대한 기술 지원을 실시하였다. Figure 98은 기존 APCC 전산실 장비배치 대비 이전 장비들에 대한 장비배치 시 변경된 배치도를 나타낸다.

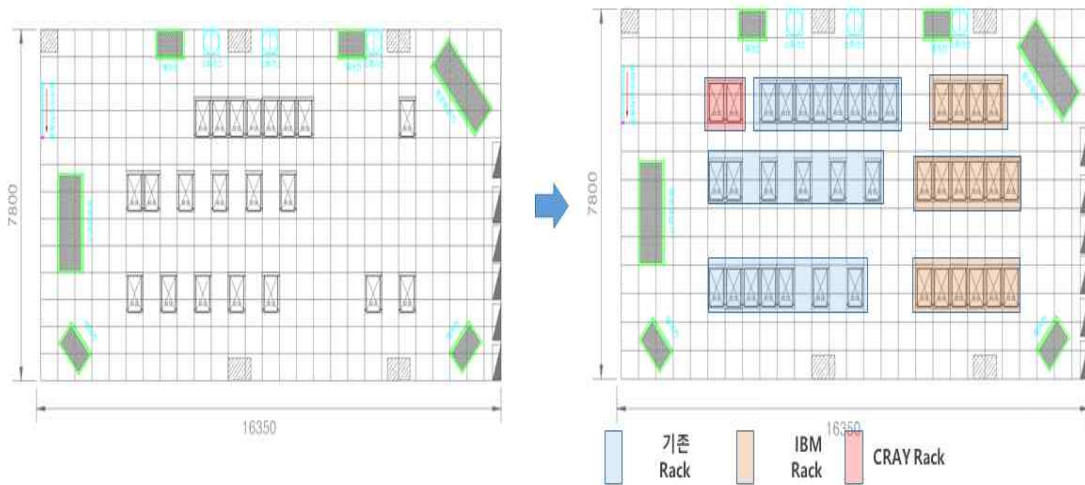


Figure 98 Layout schematic for storage rack

5. 결론 및 향후 과제

기후예측정보 활용 기술개발을 통해 축적된 기술과 산출물을 기후서비스에 효과적으로 활용하기 위해서 APEC기후센터에서는 목적에 맞는 개별 기후정보서비스를 통해 기후예측자료의 생산, 가공, 제공 서비스를 실시하고 있다. 하지만, 다양한 사용자의 요구를 반영하기 위한 기후서비스를 개발하기 위해서는 해당 정보서비스들의 통합적인 활용이 필요하며 이를 반영한 체계가 필요하다. 즉, 다중모델 기반의 기후예측자료 제공, 해당 자료의 표출 및 가공을 포함하는 일련의 서비스가 연계되어 구성된 클라우드 기술 기반의 기후서비스 플랫폼이 체계적으로 구축되어야 한다. 해당 과제를 통해서 APCC의 기후정보서비스의 점진적인 통합을 위하여 APCC의 대표적인 기후정보서비스인 CLIK, ADSS, CLIPs에 대응하는 기후자료, 기후자료처리, 계절예측 및 검증서비스들을 플랫폼 환경으로 통합 구축하였다. 해당 서비스들에서 공급 중인 기후자료는 APCC 다중모델 앙상블을 통해 생산된 결정론적/확률론적 기후예측자료 및 검증자료들을 기본으로 하여, 사용자가 자신이 원하는 개별 기후모델과 변수를 선택하여 맞춤형으로 계절예측 및 검증자료를 생산할 수 있는 온라인 사용자 맞춤형 서비스를 플랫폼 내에 구축하여 서비스를 시작하였으며, 각 기후자료들을 지역 및 변수 추출, 합성할 수 있는 기후자료 처리서비스로 플랫폼을 확장하였다. 또한 기후예측시스템 파일럿 현업화 서버를 활용한 연구개발 허브를 기상과학원과 공동으로 구축하고 해당 구축을 위한 기술지원을 성공적으로 실시하였다.

해당 과제를 통해 기존 기후자료제공서비스인 ADSS, 온라인 맞춤형 계절예측 및 검증정보 생산을 위한 기후정보응용도구인 CLIK, 지역추출 및 표출을 위한 기후자료처리도구인 CLIPs를 플랫폼 기반 기술을 활용하여 확장 구축 완료하였다. 이는 기존의 목적 지향적 기후정보서비스 시스템을 구축하는 과제들과는 달리 시스템 자원 운영 효율성을 높이고 예산절감 효과를 거둘 수 있는 혁신적인 방안이며, 또한 산재해 있는 APCC 기후정보서비스의 점진적인 통합을 이루어 증가하는 기후정보 수요에 대응할 수 있는 방안이다. 향후 APCC의 기후정보서비스 중 일부 남아있는 기능 및 서비스 통합을 완료하고 APCC 기후정보서비스 로드맵을 기반으로 중장기적 관점에서의 APCC 기후서비스 개발 방향 및 계획에 대하여 면밀한 검토가 필요하다.

사사

이 연구는 APCC의 지원을 통해 수행 되었습니다.

References

- Trac time and estimation 플러그인 제공 사이트,
<https://trac-hacks.org/wiki/TimingAndEstimationPlugin>, 2021.1.
- Trac accountManager 플러그인 제공 사이트, <https://trac-hacks.org/wiki/AccountManagerPlugin>,
2021.1.
- Trac Masterticket 플러그인 제공 사이트, <https://trac-hacks.org/wiki/MasterTicketsPlugin>, 2021.1.
- Trac subticket 플러그인 제공 사이트, <https://trac-hacks.org/wiki/SubticketsPlugin>, 2021.1.
- Trac Gantt and calendar 플러그인 제공 사이트, <https://trac-hacks.org/wiki/GanttCalendarPlugin>,
2021.1.
- Trac jsGantt 플러그인 제공 사이트, <https://trac-hacks.org/wiki/TracJsGanttPlugin>, 2021.1.
- Trac Section Edit 플러그인 사이트, <https://trac-hacks.org/wiki/SectionEditPlugin>, 2021.1.
- Trac Auto complete users 플러그인 사이트, <https://trac-hacks.org/wiki/AutocompleteUsersPlugin>,
2021.1.
- Trac clone ticket 플러그인 사이트, <https://trac-hacks.org/wiki/CloneTicketPlugin>, 2021.1.
- Trac math 플러그인 사이트, <https://trac-hacks.org/wiki/TracMathPlugin>, 2021.1.
- Trac LastModified 플러그인 사이트, <https://trac-hacks.org/wiki/LastModifiedMacro>, 2021.1.
- Trac 플러그인 가이드 및 링크 제공 사이트, <https://trac.edgewall.org/wiki/1.2/TracPlugins>,
2021.1.
- APCC CLIK Web site, <http://clik.apcc21.org>, 2021.1.
- APCC 홈페이지 기후전망 제공 웹사이트, <https://www.apcc21.org>, 2021.1.
- Peter Mell and Timoth Grance, The NIST definition of cloud computing, January 2011,
Communications of ACM 53(6).
- PaaS-TA 플랫폼서비스, 한국정보화진흥원, <https://paas-ta.kr/>, 2021.1.
- 플랫폼 컨테이너 오케스트레이션 웹 사이트, <https://kubernetes.io/>, 2021.1.
- Docker-java support & open source GitHub, <https://github.com/docker-java/docker-java/>,
2020.1.
- Web UI support & reference web site, <http://getbootstrap.com/>, 2021.1.

APCC Climate Service Platform web site, <https://cliks.apcc21.org/>, 2021.1.

Spring, Spring Boot, spring.io/projects/spring-boot

Spring, Spring Tool Suit, spring.io/tools

Maven, Maven Repository, mvnrepository.com

Toast Meetup, REST API, meetup.toast.com/posts/92

Spring, Building REST services with Spring, spring.io/guides/tutorials/rest

Spring, Spring Rest Docs, spring.io/projects/spring-restdocs

Spring, Spring Data MongoDB Docs,
docs.spring.io/spring-data/mongodb/docs/current/reference/html/#reference

Spring, Messaging with RabbitMQ, spring.io/guides/gs/messaging-rabbitmq

Spring, mybatis, mybatis.org/spring/ko/getting-started.html

Wikipedia, MVC design pattern, ko.wikipedia.org/wiki/모델-뷰-컨트롤러

Wikipedia, Ajax, ko.wikipedia.org/wiki/Ajax

Taudes, A. 1998. "Software Growth Options," *Journal of Management Information Systems* (15:1), pp.165-185.

Tiwana, A. 2015. "Evolutionary Competition in Platform Ecosystems," *Information Systems Research* (26:2), pp. 266-281.

Parker, G., Van Alstyne, M., & Choudary, S. (2016). *Platform Revolution*.

Venuti, F, 2017. "The future of Data Services" *Workshop on Open data week 2017*.

부록 A. CLIK (Climate Information Toolkit) 사용자 매뉴얼

1. 개요

APCC의 플랫폼 기반 웹 서비스인 CLIK (Climate Information Toolkit)은 APCC 계절예측 자료인 다중모델 앙상블(Multi-Model Ensemble, MME), 모델, 그리고 대기해양접합 대순환모델 상호비교 프로젝트(Coupled Model Intercomparison Project Phase5, CMIP5) 자료를 제공한다. 또한 개별 모델 자료를 기반으로 하는 사용자 맞춤형 계절예측 및 검증 서비스도 제공하고 있다.

사용자는 웹 페이지에서 원하는 자료를 선택하여 내려받거나 wget과 같은 자료 전송 프로그램을 이용하여 직접 내려받을 수 있다. 또한 파이썬, 자바 언어를 이용한 Open API 사용에 익숙한 사용자는 개발 중인 프로그램 내에서 API를 이용하여 자료를 내려받을 수 있다.

2. 메인 화면

Figure 1은 CLIK의 메인 화면을 보여준다. 메인 화면은 CLIK에 대한 기본 정보와 메뉴 등을 제공한다.

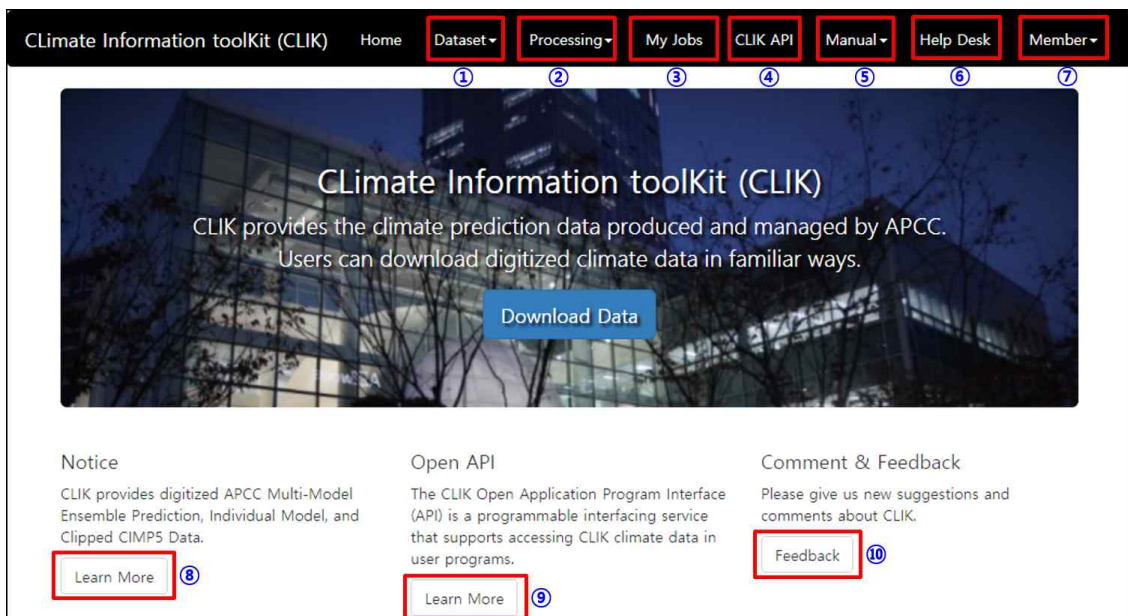


Figure 1. Main page

Table 1에 Figure 1의 상단 메뉴의 기능에 대하여 나타내었다.

Table 1. Head menus

메뉴	설명
① Dataset	각 자료의 개요(Overview)와 다운로드 서비스를 제공한다.
② Processing	계절예측, 검증 등의 사용자 맞춤 자료처리 서비스를 제공한다.
③ My Jobs	사용자의 Job 목록 및 처리 현황을 확인할 수 있다. 이 메뉴는 로그인 후 이용할 수 있다.
④, ⑨ CLIK API	API를 사용하는 방법 및 예제를 제공한다.
⑤ Manual	사용자 매뉴얼을 보거나 다운로드할 수 있다.
⑥, ⑩ Help Desk	서비스에 대한 이용 문의를 하거나 서비스에 대하여 제안 사항이 있거나 오류 및 불편 사항이 있으면 문의할 수 있다.
⑦ Member	로그인 및 신규 가입할 수 있다.
⑧ Notice	서비스의 공지를 확인할 수 있다.

메인 화면의 하단에서는 최신 다중모델앙상블, 대한민국 기상청의 GloSea5GC2 모델 자료를 확인할 수 있다. Figure 2의 그림이나 링크를 선택하면 해당 데이터셋 페이지로 바로 연결된다.

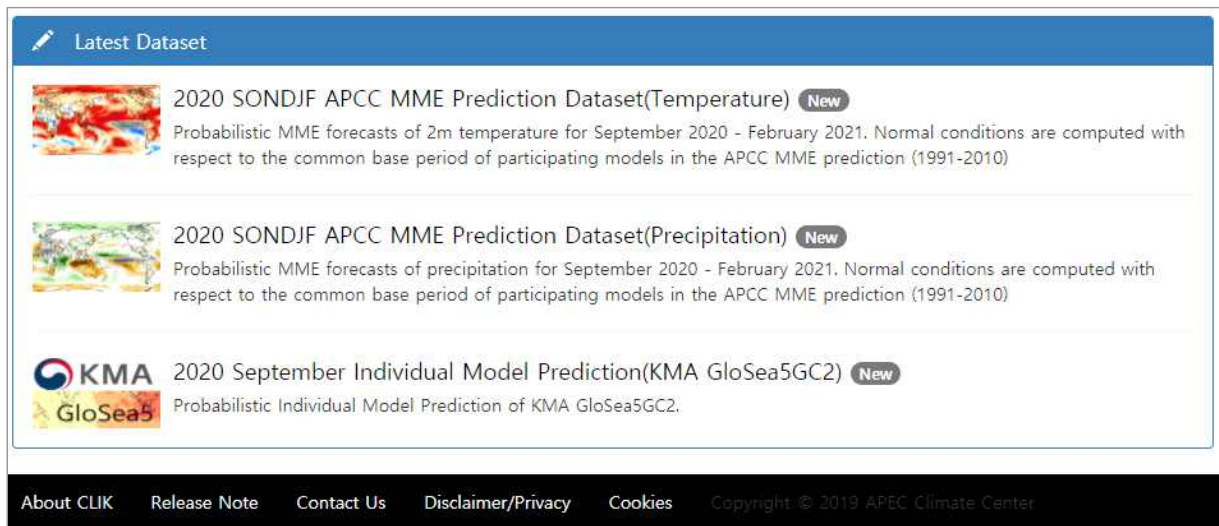


Figure 2. Latest dataset and tail menus

Table 2. Tail menus

메뉴	설명
About CLIK	CLIK 서비스에 대한 기본 정보를 제공한다.
Release Note	CLIK 서비스의 Release 기록 정보를 제공한다.
Contact Us	연락 가능한 e-mail 주소이다.
Disclaimer/Privacy	개인정보 보호 정책에 대하여 소개한다.
Cookies	쿠키를 생성하고 사용하는 정보를 제공한다.

3. Member 메뉴

Figure 3은 Member 메뉴의 하위 메뉴를 보여준다.

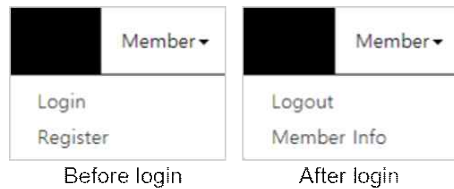


Figure 3. Member menu

가. 사용자 계정 등록

Figure 4. Member registration in APCC Single Sign On System (SSO)

APCC 사용자 계정이 없으면 **Register**를 선택하여 사용자 계정을 등록 할 수 있다(Figure 3). CLIK의 사용자 등록 및 관리는 APCC Single Sign On (SSO) 시스템을 이용한다.

APCC Single Sign On (SSO) 시스템은 모든 APCC 정보 시스템의 사용자 계정을 관리한다. 사용자는 APCC SSO에 등록한 계정을 이용하여 대부분의 웹 서비스에 로그인할 수 있다.

사용자가 Figure 4에서와 같이 이름, e-mail을 입력하고 “**apply** “ 버튼을 선택하면, APCC SSO는 등록을 승인하기 위한 e-mail을 전송한다. 사용자가 가입 승인 e-mail의 링크를 선택하여 승인이 완료되면 CLIK에 로그인할 수 있다.

나. 로그인

Figure 3의 **Login** 메뉴를 이용하여 CLIK에 로그인할 수 있다. Figure 5는 **Login**을 선택했을 때 사용자 아이디, 비밀번호를 입력할 수 있는 팝업을 보여준다.



Figure 5. Popup for login

사용자 아이디와 비밀번호를 입력하고 “**Login**” 버튼을 선택하면 웹 서비스에 로그인이 완료된다. 일부 사용자 정보는 쿠키(Cookie)의 형태로 사용자의 컴퓨터에 저장된다.

다. 사용자 정보

사용자 정보에서는 APCC SSO에 입력한 기본정보와 사용자의 API Key를 확인할 수 있다. CLIK API를 이용하려면 “**Get key**” 버튼을 이용하여 key를 발급 받아야 한다.

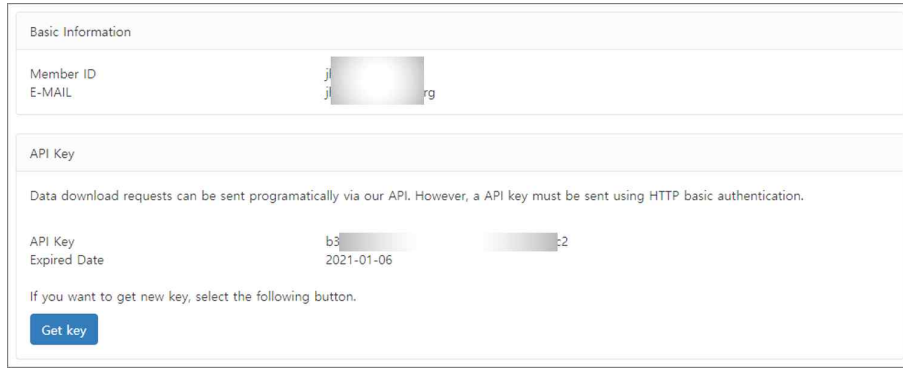


Figure 6. Member Info

4. Dataset 메뉴

Figure 7은 Dataset 메뉴를 보여준다. CLIK은 현재 Multi-Model Ensemble Forecast (MME), Individual Model, CMIP5 자료를 제공한다. MME 자료의 경우 3개월, 6개월 예측 자료의 형태로 제공하고 있다.



Figure 7.
Dataset menu

Dataset 화면은 Overview와 Download로 나뉘어져 있다(Figure 8). Overview 탭은 자료에 대한 설명을 보여준다. 사용자는 Download 탭에서 원하는 자료를 선택하여 요청할 수 있다.

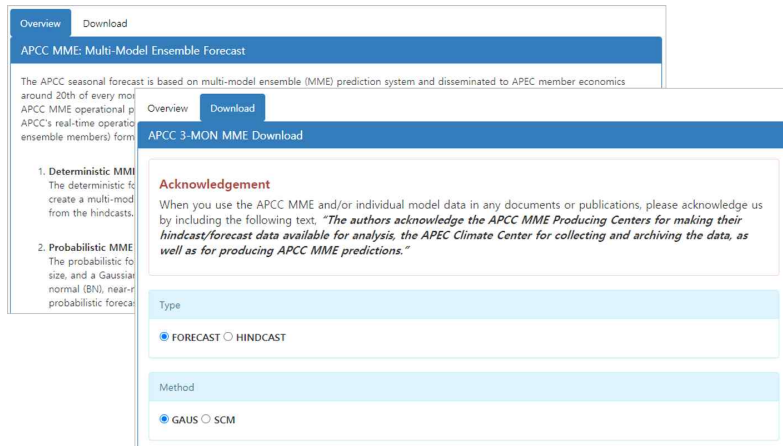


Figure 8. Dataset page

가. Multi-Model Ensemble (MME) dataset

사용자는 Download 탭에서 method, variable, period 등의 속성을 선택하여 다운로드를 요청할 수 있다.

	JFM	FMA	MAM	AMJ	MJJ	JJA	JAS	ASO	SON	OND	NDJ	DJF
2015	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2017	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2018	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2019	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2020	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 9. Selecting options for MME dataset

사용자는 다음과 같은 방법으로 자료를 요청할 수 있다.

- ① Type: 기후자료 타입 선택
- ② Methods: MME 예측을 위한 방법 선택. SCM (DMME, Simple Composite Method) and GAUS (PMME, GAUSSIAN approximation)
- ③ Variables: 변수 선택(변수의 상세 내용은 Overview 탭 참고)
- ④ Period: 자료 통계 방법 선택(Seasonal mean은 Monthly mean의 평균)
- ⑤ Date: 다운로드할 자료의 기간 선택(테이블의 년도, 시즌을 선택하면 전체 년, 시즌을 한번에 선택 가능)
- ⑥ “Request” 버튼 선택

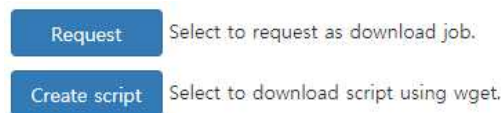


Figure 10. Request data

Figure 10에서 “Request” 버튼을 선택하면 자료 다운로드를 위한 Job이 등록된다. 사용자가 로그인하지 않았을 경우 Figure 11과 같이 “Request” 버튼은 비활성화된다.



Figure 11. Request button when not logged in

Figure 10의 ” Create script “ 버튼은 사용자의 로컬 서버나 컴퓨터에서 wget 명령을 이용하여 자료를 직접 다운로드할 수 있는 스크립트(Figure 12)를 전송하여 스크립트(리눅스 시스템 용) 파일이 즉시 다운로드 된다.

```

#-----
# Personal setting
#-----
# chane to your user id
userid="userid"
# change to your password
password="password"
# cookie file path(You can change to the other file.)
cookie_path="apcc_cookies"
# option to save cookies. If you want to save cookies, don't use the cookie_option.
#cookie_option=""
cookie_option="--load-cookies ${cookie_path} --save-cookies ${cookie_path} --keep-session-cookies "

# You can set verifying the certificate or not.
#certificate_option="--no-check-certificate"
certificate_option=""

echo `date +%F %T` " Now start to download."

#-----
# Each file of the same variable has the same file name.
# So please set(change) the folder to save file, or set file path to use '-O' option
#-----

wget ${cookie_option} --user=${userid} --password=${password} ${certificate_option}
https://sdownload.apcc21.org/MME/3-MON/FORECAST/GAUS/MAY/2020/prec.nc -O 3-MON_FORECAST_GAUS_MAY_2020_prec.nc
wget ${cookie_option} --user=${userid} --password=${password} ${certificate_option}
https://sdownload.apcc21.org/MME/3-MON/FORECAST/GAUS/MAY/2020/slp.nc -O 3-MON_FORECAST_GAUS_MAY_2020_slp.nc
wget ${cookie_option} --user=${userid} --password=${password} ${certificate_option}
https://sdownload.apcc21.org/MME/3-MON/FORECAST/GAUS/MAY/2020/t2m.nc -O 3-MON_FORECAST_GAUS_MAY_2020_t2m.nc
wget ${cookie_option} --user=${userid} --password=${password} ${certificate_option}
https://sdownload.apcc21.org/MME/3-MON/FORECAST/GAUS/MAY/2020/t850.nc -O 3-MON_FORECAST_GAUS_MAY_2020_t850.nc
wget ${cookie_option} --user=${userid} --password=${password} ${certificate_option}
https://sdownload.apcc21.org/MME/3-MON/FORECAST/GAUS/MAY/2020/z500.nc -O 3-MON_FORECAST_GAUS_MAY_2020_z500.nc

```

Figure 12. Downloading script using wget

Figure 12의 스크립트를 이용하여 자료를 다운로드하기 위해서 사용자는 스크립트를 자신의 환경에 맞게 수정할 필요가 있다.

- ① userid, password: 자신의 아이디, 비밀번호를 입력한다.
- ② cookie_option: 웹 사이트의 쿠키 저장을 원하지 않으면 cookie_option의 내용을 비우면 된다. 쿠키를 저장하지 않으면 자료 파일을 다운로드 할 때마다 사용자 인증을 거치게 되어 자료 전송이 조금 지연될 수 있다.
- ③ certificate_option: https 통신에 사용되는 인증서를 사용자의 서버나 컴퓨터에서 검증하지 않을 경우 " --no-check-certificate "를 설정한다.
- ④ " -O " 옵션: 스크립트의 마지막에 위치한 wget 명령에서 " -O " 옵션을 사용하는데 " -O " 옵션을 통하여 실제로 저장할 파일의 위치 및 파일명을 지정할 수 있다.

다운로드할 자료의 URL은 Table 3과 같이 구성된다. Table 3에서 대괄호(" [] ") 안의 값은 Figure 9의 옵션값을 참고한다. Month는 JAN, FEB와 같은 각 월의 약자이며, Season은 Figure 9에서 Date 테이블 상단의 JFM, FMA 등과 같은 값이다.

Table 3. Download url of MME

Period	URL
Monthly Mean	https://sdownload.apcc21.org/MME/ [Lead Month] / [Type] / [Method] / [Month] / [Year] / [Variable].nc
Seasonal Mean	https://sdownload.apcc21.org/MME/ [Lead Month] / [Type] / [Method] / [Month] / [Season] / [Year] / [Variable].nc

나. Individual Model

사용자는 Download 탭에서 institute, model, variable 등의 속성을 선택하여 다운로드를 요청할 수 있다.

The screenshot displays a web form for selecting options for an Individual Model dataset. The form is organized into five sections, each with a light blue header:

- Type:** Radio buttons for FORECAST (selected) and HINDCAST.
- Institute:** Radio buttons for APCC (selected), BCC, BOM, CMCC, CWB, HMC, KMA, MGO, MSC, NASA, NCEP, PNU, and UKMO.
- Model:** Radio buttons for CCSM3 and SCOPS (selected).
- Variable:** Checkboxes for prec, slp, sst, t2m, t850, u200, u850, v200, v850, and z500. All are checked.
- Date:** A grid for selecting years and months. A note reads: "* If you want to get data of each year or season at once, select year or month heads." The months 01-12 are listed at the top. Below, years 2017-2020 are shown with checkboxes for each month. For 2020, all months (01-12) are checked.

Figure 13. Selecting options for Individual Model dataset (FORECAST type)

사용자는 다음과 같은 방법으로 자료를 요청할 수 있다.

- ① Type: 자료 타입 선택
- ② Year: 자료 생산 년도 선택(Type에서 HINDCAST를 선택했을 때 선택 가능, Figure 14)
- ③ Institute: 참여 model의 기관명 선택

- ④ Model: model 선택
- ⑤ Variables: 변수 선택(변수의 상세 내용은 Overview 탭 참고)
- ⑥ Date: 다운로드할 자료의 기간 선택(테이블의 년도, 시즌을 선택하면 전체 년, 시즌을 한번에 선택 가능)
- ⑦ “Request” 버튼 선택

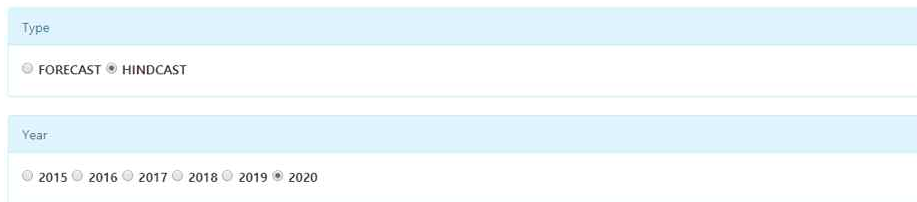


Figure 14. Year option for Individual Model dataset (HINDCAST type)

wget을 이용한 스크립트를 다운로드하려면 “Create script” 버튼을 선택한다. 자료의 다운로드 URL은 Table 4와 같이 구성된다. Table 4에서 대괄호(“ [] “) 안의 값은 Figure 13, 14의 옵션값을 참고한다. Month는 JAN, FEB와 같은 각 월의 약자에 해당한다.

Table 4. Download url of individual model

	URL
FORECAST	https://sdownload.apcc21.org/MODEL/FORECAST/[Institute]_[Model] / [Month] / [Year] / [Variable].nc
HINDCAST	https://sdownload.apcc21.org/MODEL/ HINDCAST[Hindcast Year] / [Institute]_[Model] / [Month] / [Year] / [Variable].nc

※ 개별모델 및 다중모델양상블(MME) 자료를 사용하여 연구를 수행하시는 경우 보고서 및 논문에 들어갈 감사문구를 다음과 같이 요청 드립니다. “연구에 사용된 다중모델 양상블 자료는 APCC MME Producing Centres (PCs)에 의해 생산된 hindcast/forecast 자료를 바탕으로 APCC가 수집, 재가공하였으며 자료를 생산/제공해주신 APEC 기후센터에 감사 드립니다.”

다. CMIP5

CMIP5 페이지에서는 지역별로 CMIP5 자료를 선택하여 요청할 수 있다. 각 지역의 CODE를 마우스로 가리키면 해당 지역의 경위도 값을 확인할 수 있다(그림 15).

You can download the list of clipping areas [here](#).

CODE	NATION	NATION CODE	STATE	STATE CODE
<input type="radio"/> BF	Burkina Faso	BF		
<input type="radio"/> BI	Burundi	BI		
<input type="radio"/> KH	Cambodia	KH		
<input type="radio"/> CM	Cameroon	CM		
<input type="radio"/> CAA	Canada	CA	Alberta	AB
<input type="radio"/> CABC	Canada	CA	British Columbia	BC
<input type="radio"/> CAMB	Canada	CA	Manitoba	MB

Figure 15. Nations (States) list of CMIP5

- ① 다운로드할 국가 혹은 주 선택
- ② “Request” 버튼 선택

국가가 아닌 주 단위로 제공되는 국가는 다음과 같다.

- 미국
- 러시아
- 중국
- 캐나다

wget을 이용한 스크립트를 다운로드하려면 “Create script” 버튼을 선택한다. 다른 자료와 달리 CMIP5 자료는 사용자 인증 없이 다운로드 가능하므로 사용자 설정, 쿠키 설정이 필요하지 않다. 자료의 다운로드 URL은 Table 5와 같이 구성된다. Table 5에서 대괄호(“ [] “) 안의 값은 Figure 15의 CODE 값이다.

Table 5. Download url of CMIP5

URL
http://download.apcc21.org/CMIP5/cmip5_daily_[CODE].zip

※ APCC의 자료를 사용하여 연구를 수행하신 경우 보고서 및 논문에 들어갈 감사문구를 다음과 같이 요청 드립니다. “연구에 사용된 ‘CMIP5’ 자료제공에 도움을 주신 APEC 기후센터 (APCC)에 감사 드립니다.” 또한, 원본 데이터가 요구하는 인용 문구 및 사용 정책을 각 웹사이트에서 확인하셔야 함을 유념하시기 바랍니다.

5. Processing

Figure 16은 Processing 메뉴를 보여준다. Processing 메뉴에서는 Prediction(계절예측), Verification(검증) 서비스를 제공한다. Prediction과 Verification은 사용자가 직접 선택한 Model을 기반으로 생성한 계절예측 MME, 검증 결과를 이미지, Netcdf 파일 형태로 제공한다.



Figure 16.
Processing menu

가. Prediction

사용자는 Prediction 페이지에서 년도, 시즌, MME 종류, 모델을 선택하여 계절예측 MME(Multi-Model Ensemble) 자료를 요청할 수 있다.

 A screenshot of the 'Prediction' web interface. The page has a blue header with the title 'Prediction'. Below the header, there are three main sections:

- Lead Month:** A dropdown menu set to '3-MON'.
- Year / Season:** Two dropdown menus set to '2020' and '9'.
- Methods:** Radio buttons for 'Deterministic' (selected) and 'Probabilistic'.

 Below these sections is a 'Models' section with a list of model names and checkboxes:

- ALL
- APCC_SCOPS
- BOM_ACCESS-S1
- CMCC_SPS3
- CWB_GFST119
- HMC_SL-AV
- KMA_GLOSEA5GC2
- MGO_MGOAM-2
- MSC_CANSIPsv2
- NASA_GEOS-S25-2.1
- NCEP_CFSv2
- PNU_CGCMv2.0
- UKMO_GLOSEAS

 At the bottom of the model list, there is a red-bordered 'Predict' button. Below the model list are two buttons: 'Download (.png)' and 'Download (.nc)'. At the very bottom, there is a 'Product Description' section with a paragraph of text.

Figure 17. Selecting options for prediction

사용자는 다음과 같은 방법으로 계절예측 자료를 요청할 수 있다.

- ① Year / Season: 생성하고자 하는 년, 시즌 선택
- ② Methods: MME 예측 방법 선택
- ③ Models: 계절예측 MME 생성에 참여할 모델 선택
- ④ “Predict” 버튼 선택

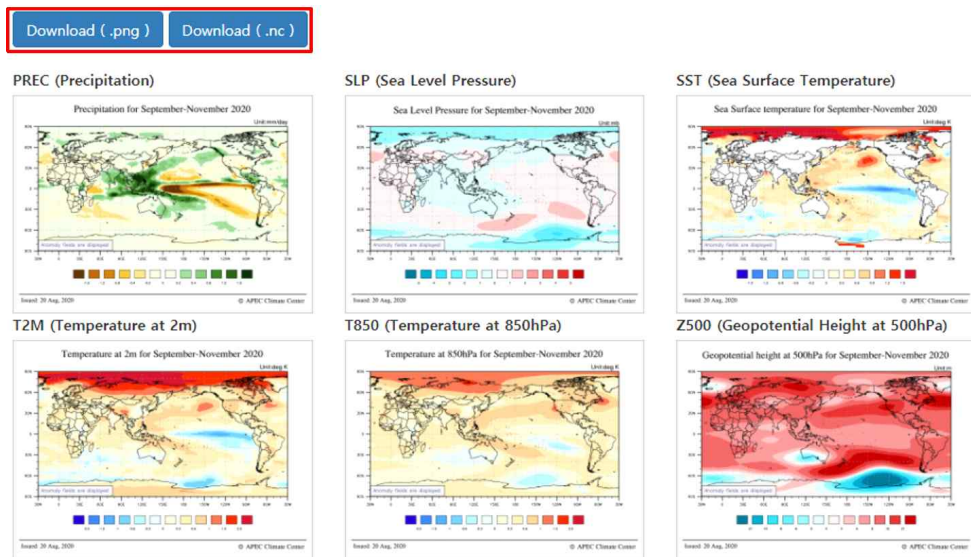


Figure 18. Prediction plot images

사용자가 입력한 선택에 해당하는 계절예측 자료가 생성되어 있는 경우 Figure 18과 같이 페이지 하단에 각 변수의 plot 이미지가 출력된다. 각 변수를 선택하면 큰 이미지로 볼 수 있다. 또한 “Download” 버튼을 선택하여 전체 변수의 plot 이미지나 Netcdf 파일을 다운로드할 수 있다.

사용자가 입력한 조건에 따라 생성된 자료가 없는 경우 Figure 19와 같이 신규 job이 생성되어 자료 생성이 시작된다. 사용자는 My Jobs 메뉴로 이동하여 입력된 job의 진행상황을 확인할 수 있으며 자료 생성이 끝나면 다운로드할 수 있다. 또한 동일한 조건을 입력하고 “Predict “ 버튼을 선택하여 Figure 18과 같이 plot 이미지를 확인할 수 있다.

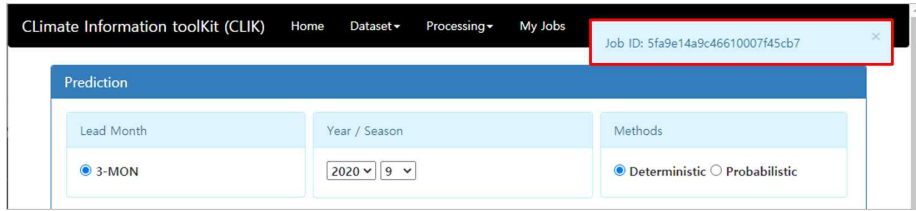


Figure 19. Registration of new prediction job

사용자가 입력한 조건과 동일한 자료 생성이 이미 진행되고 있는 경우에는 Figure 20과 같은 메시지가 출력될 수 있다. 일정 시간이 지난 후 “Predict “ 버튼을 눌러 자료를 확인할 수 있다.

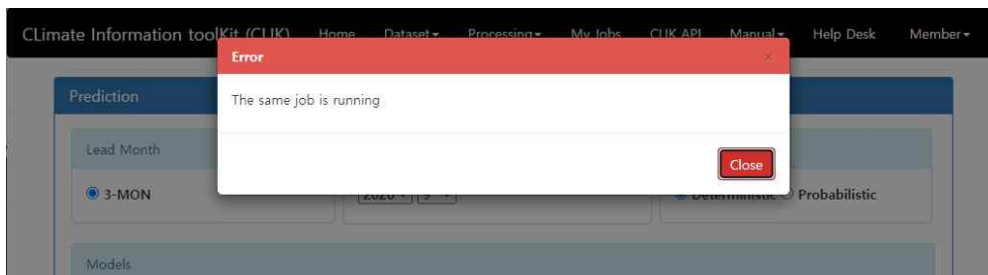


Figure 20. Notification of the running job

나. Verification

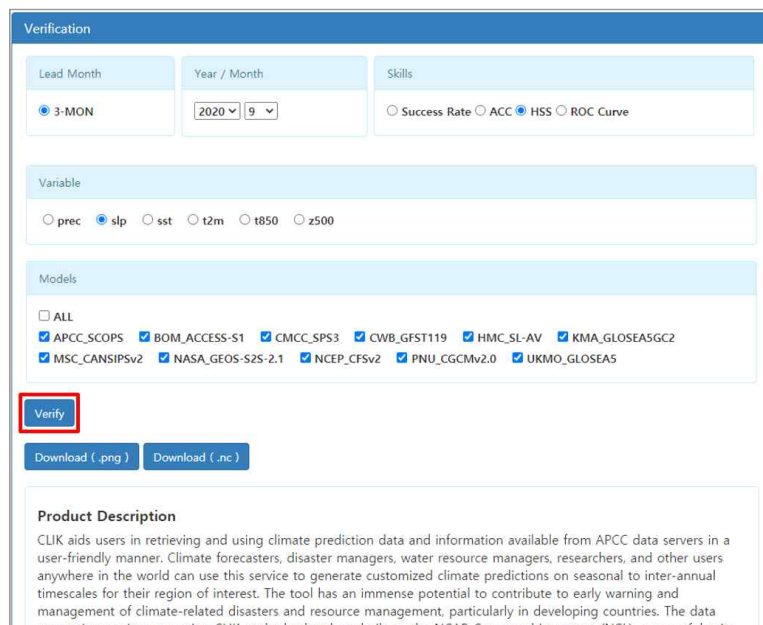


Figure 21. Selecting options for Verification

사용자는 Verification 페이지에서 년도, 시즌, 검증 기법, 변수, 모델을 선택하여 계절예측 (Hindcast) 검증 자료를 요청할 수 있다.

- ① Year / Season: 생성하고자 하는 년, 시즌 선택
- ② Skills: 검증기법 선택(Table 6참고)
- ③ Variable: 생성하려는 변수 선택
- ④ Models: 검증 자료 생성에 참여할 모델 선택
- ⑤ “Verify” 버튼 선택

Table 6. Verification skills for hindcast

Skills	MME Class	Long name
SR	Deterministic	Success Rate
ACC	Deterministic	Anomaly Correlation Coefficient
HSS	Probabilistic	Heidke Skill Score
ROC Curve	Probabilistic	Relative Operating Characteristics Curve

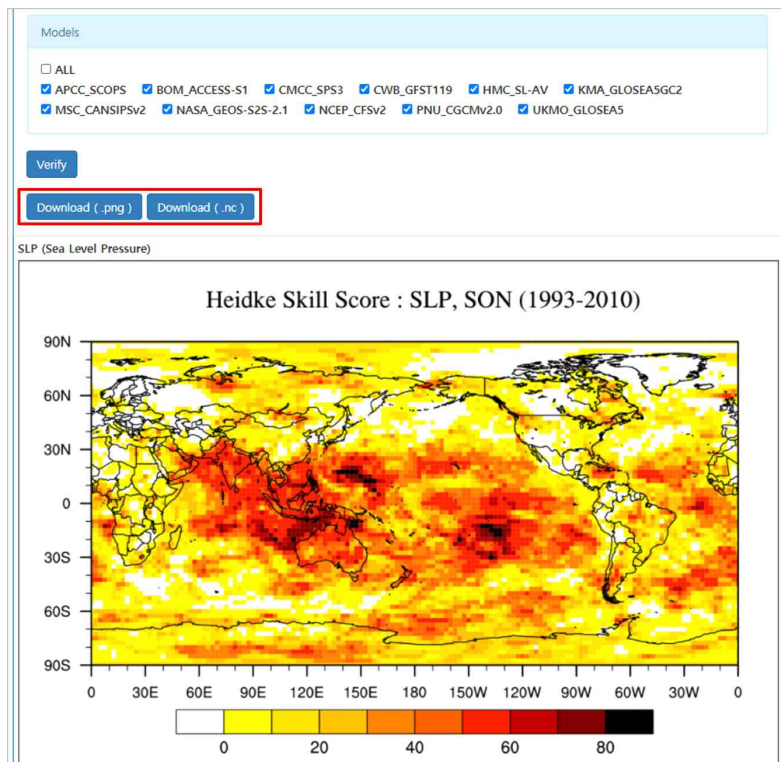


Figure 22. Verification plot image

사용자가 입력한 선택에 해당하는 검증 자료가 생성되어 있는 경우 Figure 22와 같이 페이지 하단에 plot 이미지가 출력된다. 이미지를 선택하면 큰 이미지로 볼 수 있다. 또한 “Download” 버튼을 선택하여 plot 이미지나 Netcdf 파일을 다운로드할 수 있다.

사용자가 입력한 조건에 따라 생성된 자료가 없는 경우 Figure 23과 같이 신규 job이 생성되어 자료 생성이 시작된다. 사용자는 **My Jobs** 메뉴로 이동하여 입력된 job의 진행상황을 확인할 수 있으며 자료 생성이 끝나면 다운로드할 수 있다(Figure 24). 또한 동일한 조건을 입력하고 “Verify “ 버튼을 선택하여 Figure 22와 같이 plot 이미지를 확인할 수 있다.

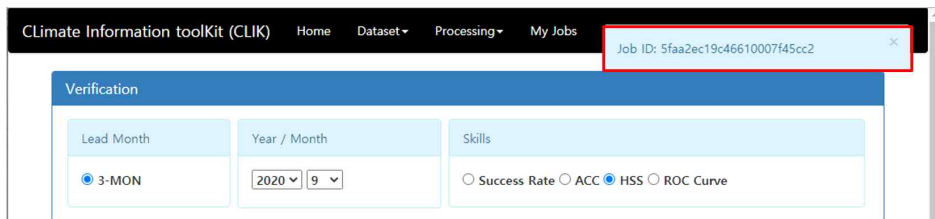


Figure 23. Registration of new verification job

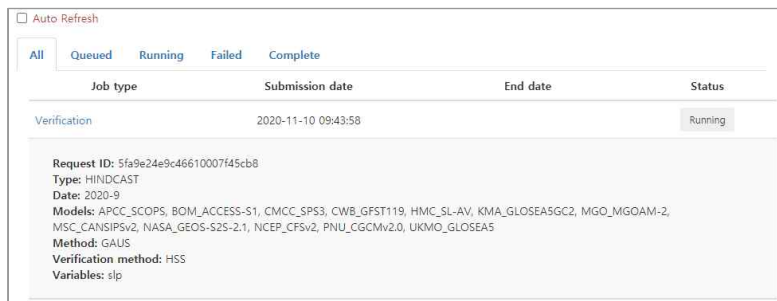


Figure 24. Details of the verification job

6. My Jobs

Dataset의 Download 탭 및 Processing 메뉴에서 자료를 요청하면 화면의 우측 상단에서 Job 아이디를 확인할 수 있다(Figure 25). Job 아이디 알림 메시지는 출력 후 곧 사라진다.



Figure 25. Notification of Job ID

My Job 메뉴에서는 사용자가 요청한 job 목록 및 진행상황을 확인할 수 있다.

<input checked="" type="checkbox"/> Auto Refresh				
All	Queued	Running	Failed	Complete
Job type	Submission date	End date	Status	
MODEL	2020-04-20 15:51:28	2020-04-20 15:51:31	Download	
MME_3MONTH	2020-04-17 15:45:29	2020-04-17 15:45:33	Download	

All	Queued	Running	Failed	Complete
Job type	Submission date	End date	Status	
MME_3MONTH	2020-04-17 14:50:18		Queued	
MME_3MONTH	2020-04-17 14:43:50		Queued	

All	Queued	Running	Failed	Complete
Job type	Submission date	End date	Status	
MME_3MONTH	2020-04-17 15:07:05	2020-04-17 15:07:11	Failed	
MME_3MONTH	2020-04-17 14:56:42	2020-04-17 14:56:45	Failed	

Figure 26. Job list

- 상단의 “Auto Refresh” 를 선택하면 Job 목록은 20초마다 자동으로 업데이트 된다.
- Job 목록은 Job의 종류, 입력 시간, 종료 시간, 상태를 보여준다.
- Job은 대기(Queued), 실행(Running), 실패(Failed), 완료(Completed)의 4가지 상태로 분류된다.
- Job이 성공적으로 완료되었을 때 Job의 상태는 Download로 변경되고, “Download” 버튼을 선택하면 결과 자료가 다운로드 된다.
- “Auto Refresh” 를 선택하지 않은 상태에서 목록의 Job type(MME_3MONTH, MODEL, Prediction, Verification 등)을 선택하면 Job의 상세 내용을 확인할 수 있다. 실패한 Job의 경우 오류의 원인이 출력된다(Figure 27).

MODEL	2020-04-17 15:15:19	2020-04-17 15:15:58	Download
Request ID: 5e994977d7e3f000659eec4 Dataset: MODEL Type: FORECAST Institute: APCC Model: SCOPS Variables: u850, v200, v850, z500 Date: 202002, 202003, 202004			
MME_3MONTH	2020-04-17 15:07:05	2020-04-17 15:07:11	Failed
Request ID: 5e9947897d7e3f000659eec1 Dataset: MME_3MONTH Type: FORECAST Method: GAUS Variables: prec, slp Period: Monthly mean Date: 202001 Status: Failed Last log: [2020-04-17 15:07:11.018] [ERROR] Failed to prepare data.			

Figure 27. The details of Job

7. CLIK API

CLIK Open Application Program Interface (API)는 사용자가 프로그래밍을 이용하여 자료를 사용할 수 있도록 하는 서비스이다. 이 장에서는 사용자가 어떻게 API를 사용하는지 설명하고 사용 예제를 소개한다.

가. API key 설정

API key는 API를 사용할 때 필수적이다. Key는 다음과 같이 설정할 수 있다.

- ① 사용자 계정이 없으면 “**Registration**” 메뉴에서 사용자 가입
- ② 로그인한 후 “**Member Info**” 메뉴에서 key 발급
- ③ 다음의 내용을 \$HOME/apccapi.properties 파일에 기록

```
key=810050f2-727e-5ed3-a871-b7a881a04d34
request_url=https://request.apcc21.org/apccdata
status_url=https://request.apcc21.org/status
```

Figure 28. apccapi.properties

나. API Client 설치

CLIK은 Python, Java 언어로 작성된 API Client를 제공하고 있다(Python client 사용을 추천한다). 다음과 같은 방법으로 API Client를 설치할 수 있다.

- ① CLIK API 페이지에서 API client를 다운로드하거나 wget을 이용하여 직접 다운로드
- ② 작업 folder에 Table 7과 같은 방법으로 API Client 설치

Table 7. Download and install commands for API Client

Python	\$ wget http://download.apcc21.org/pythonapi -O apccapi.tar.gz \$ tar xvf apccapi.tar.gz
Java	\$ wget http://download.apcc21.org/javaapi -O apccapi-1.0.jar

다. API Client 사용

API Client를 설치한 후 사용자는 프로그램 내에서 자료를 요청하기 위해서 사용할 수 있다. 만일 2개 이상의 파일을 요청하면 zip 형식의 파일로 다운로드 받는다.

사용자는 Python을 사용하여 다음과 같이 자료를 다운로드할 수 있다. 예제의 type, method, variable 등의 선택사항은 Dataset, Processing 페이지에서 확인할 수 있다.

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'MME',
        'dataset': 'MME_3MONTH',
        'type': 'FORECAST',
        'method': 'SCM',
        'variable': ['prec', 't2m'],
        'period': ['Monthly mean'],
        'yearmonth': ['201909', '201910']
    },
    'mme3.zip'
)
```

Figure 30. Sample using Python: MME (3-Month)

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'MME',
        'dataset': 'MME_6MONTH',
        'type': 'HINDCAST',
        'method': 'GAUS',
        'variable': ['prec', 't2m'],
        'period': ['Monthly mean', 'Seasonal mean'],
        'yearmonth': ['201909']
    },
    'mme6.zip'
)
```

```
)
```

Figure 29. Sample using Python: MME (6-Month)

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'MODEL',
        'dataset': 'MODEL',
        'type': 'FORECAST',
        'institute': 'APCC',
        'model': 'SCOPS',
        'variable': ['prec', 't2m'],
        'yearmonth': ['201909']
    },
    'model.zip'
)
```

Figure 31. Sample using Python: Model

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'CMIP5',
        'dataset': 'CMIP5',
        'code': 'AD',
    },
    'cmip5.zip'
)
```

Figure 32. Sample using Python: CMIP5

```
#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'Prediction',
```

```

        'type': 'FORECAST',
        'method': 'SCM',
        'model': ['APCC_SCOPS', 'BOM_ACCESS-S1',
        'KMA_GLOSEA5GC2'],
        'variable': ['prec', 'slp', 'sst', 't2m', 't850', 'z500'],
        'year': '2020',
        'month': '7'
    },
    'prediction.zip'
)

```

Figure 33. Sample using Python: Prediction

```

#!/usr/bin/env python
import apccapi
c = apccapi.Client()
c.retrieve(
    {
        'jobtype': 'Verification',
        'type': 'HINDCAST',
        'method': 'GAUS',
        'model': ['APCC_SCOPS', 'BOM_ACCESS-S1',
        'CWB_GFST119', 'KMA_GLOSEA5GC2'],
        'variable': ['prec', 'slp'],
        'verification': 'ROC_Curve',
        'year': '2020',
        'month': '9'
    },
    'verification.zip'
)

```

Figure 34. Sample using Python: Verification

다음의 그림은 Java를 이용한 예제 코드이다.

```

package my.package;
import org.apcc.api.client.APIClient;
import org.apcc.api.jobdetails.Dataset;
import org.apcc.api.jobdetails.JobDetailsMME;
public class ClientTest {

```

```

public static void main(String[] args) {
    try {
        APIClient client = new APIClient();
        JobDetailsMME mme = new JobDetailsMME();
        mme.setDataset(Dataset.DATASET_MME_6MON.datasetName);
        mme.setType("FORECAST");
        mme.setMethod("GAUS");
        mme.addPeriod("Monthly mean");
        mme.addYearmonth("201909");
        mme.addVariable("prec");
        mme.addVariable("t2m");
        client.run(mme, "mme6-1.zip");
        // json type
        String details = "{\"jobtype\": \"MME\", \"dataset\": \"MME_6MONTH\",
        \"type\": \"FORECAST\", \"method\": \"SCM\", \"variable\": [\"prec\", \"t2m\"],
        \"period\": [\"Monthly mean\"], \"yearmonth\": [\"201909\"]}";
        client.run(details, "mme6-2.zip");
    } catch(Exception ex) {
        ex.printStackTrace();
    }
}
}

```

Figure 35. Sample using Java: MME (6-Month)

```

package my.package;
import org.apcc.api.client.APIClient;
import org.apcc.api.jobdetails.Dataset;
import org.apcc.api.jobdetails.JobDetailsMME;
public class ClientTest {
    public static void main(String[] args) {
        try {
            APIClient client = new APIClient();
            JobDetailsMME mme = new JobDetailsMME();
            mme.setDataset(Dataset.DATASET_MME_3MON.datasetName);
            mme.setType("FORECAST");
            mme.setMethod("GAUS");
            mme.addPeriod("Monthly mean");
            mme.addYearmonth("201909");

```

```

        mme.addVariable("prec");
        mme.addVariable("t2m");
        client.run(mme, "mme3.zip");
    } catch(Exception ex) {
        ex.printStackTrace();
    }
}
}

```

Figure 36. Sample using Java: MME (3-Month)

```

package my.package;
import org.apcc.api.client.APIClient;
import org.apcc.api.jobdetails.Dataset;
import org.apcc.api.jobdetails.JobDetailsModel;
public class ClientTest {
    public static void main(String[] args) {
        try {
            APIClient client = new APIClient();
            JobDetailsModel model = new JobDetailsModel();
            model.setDataset(Dataset.DATASET_MODEL.datasetName);
            model.setType("HINDCAST");
            model.setHindcast_year("2020");
            model.setInstitute("APCC");
            model.setModel("SCOPS");
            model.addYearmonth("201102");
            model.addYearmonth("201202");
            model.addVariable("prec");
            model.addVariable("t2m");
            client.run(model, "model.zip");
        } catch(Exception ex) {
            ex.printStackTrace();
        }
    }
}

```

Figure 37. Sample using Java: Model

```

package my.package;
import org.apcc.api.client.APIClient;

```

```

import org.apcc.api.jobdetails.Dataset;
package my.package;
import org.apcc.api.client.APIClient;
import org.apcc.api.jobdetails.Dataset;
import org.apcc.api.jobdetails.JobDetailsCMIP5;
public class ClientTest {
    public static void main(String[] args) {
        try {
            APIClient client = new APIClient();
            JobDetailsCMIP5 cmip5 = new JobDetailsCMIP5();
            cmip5.setDataset(Dataset.DATASET_CMIP5.datasetName);
            cmip5.setCode("AD");
            client.run(cmip5, "cmip5.zip");
        } catch(Exception ex) {
            ex.printStackTrace();
        }
    }
}

```

Figure 38. Sample using Java: CMIP5

```

package my.package;
import org.apcc.api.client.APIClient;
import org.apcc.api.jobdetails.JobDetailsPrediction;
public class ClientTest {
    public static void main(String[] args) {
        try {
            APIClient client = new APIClient();
            JobDetailsPrediction prediction = new JobDetailsPrediction();
            prediction.setType("FORECAST");
            prediction.setMethod("SCM");
            prediction.addModel("APCC_SCOPS");
            prediction.addModel("BOM_ACCESS-S1");
            prediction.addModel("KMA_GLOSEA5GC2");
            prediction.addVariable("prec");
            prediction.addVariable("slp");
            prediction.addVariable("sst");
            prediction.addVariable("t2m");
            prediction.addVariable("t850");
        }
    }
}

```

```

        prediction.addVariable("z500");
        prediction.setYear("2020");
        prediction.setMonth("7");
        client.run(prediction, "prediction.zip");
    } catch(Exception ex) {
        ex.printStackTrace();
    }
}

```

Figure 39. Sample using Java: Prediction

```

package my.package;
import org.apcc.api.client.APIClient;
import org.apcc.api.jobdetails.JobDetailsVerification;
public class ClientTest {
    public static void main(String[] args) {
        try {
            APIClient client = new APIClient();
            JobDetailsVerification verification = new JobDetailsVerification();
            verification.setType("HINDCAST");
            verification.setMethod("GAUS");
            verification.addModel("APCC_SCOPS");
            verification.addModel("BOM_ACCESS-S1");
            verification.addModel("CWB_GFST119");
            verification.addModel("KMA_GLOSEA5GC2");
            verification.addVariable("prec");
            verification.addVariable("t850");
            verification.addVariable("z500");
            verification.setVerification("ROC_Curve");
            verification.setYear("2020");
            verification.setMonth("9");
            client.run(verification, "verification.zip");
        } catch(Exception ex) {
            ex.printStackTrace();
        }
    }
}

```

Figure 40. Sample using Java: Verification