




전문가 육성 교육과정

교재

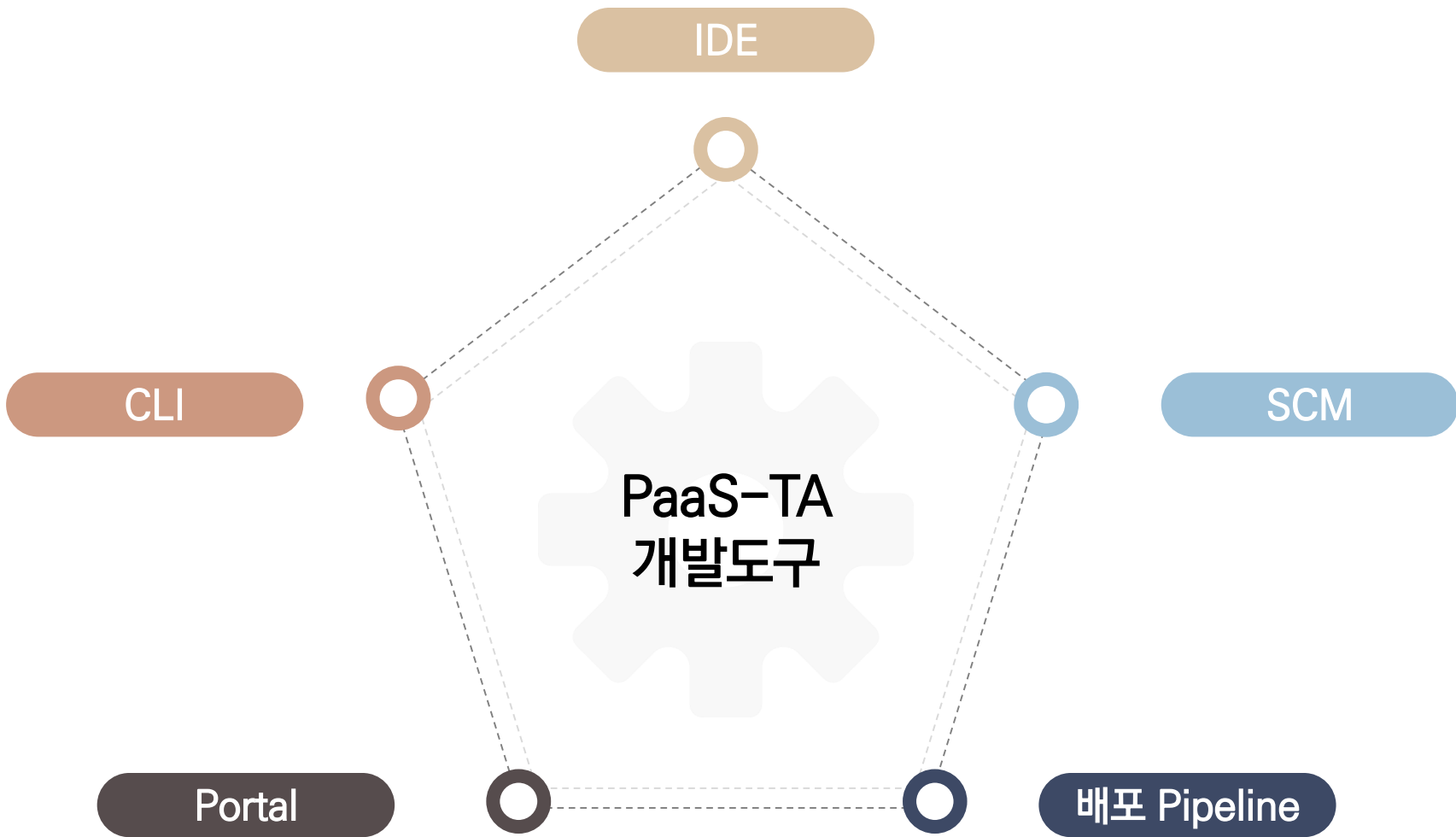
NIA 한국정보화진흥원



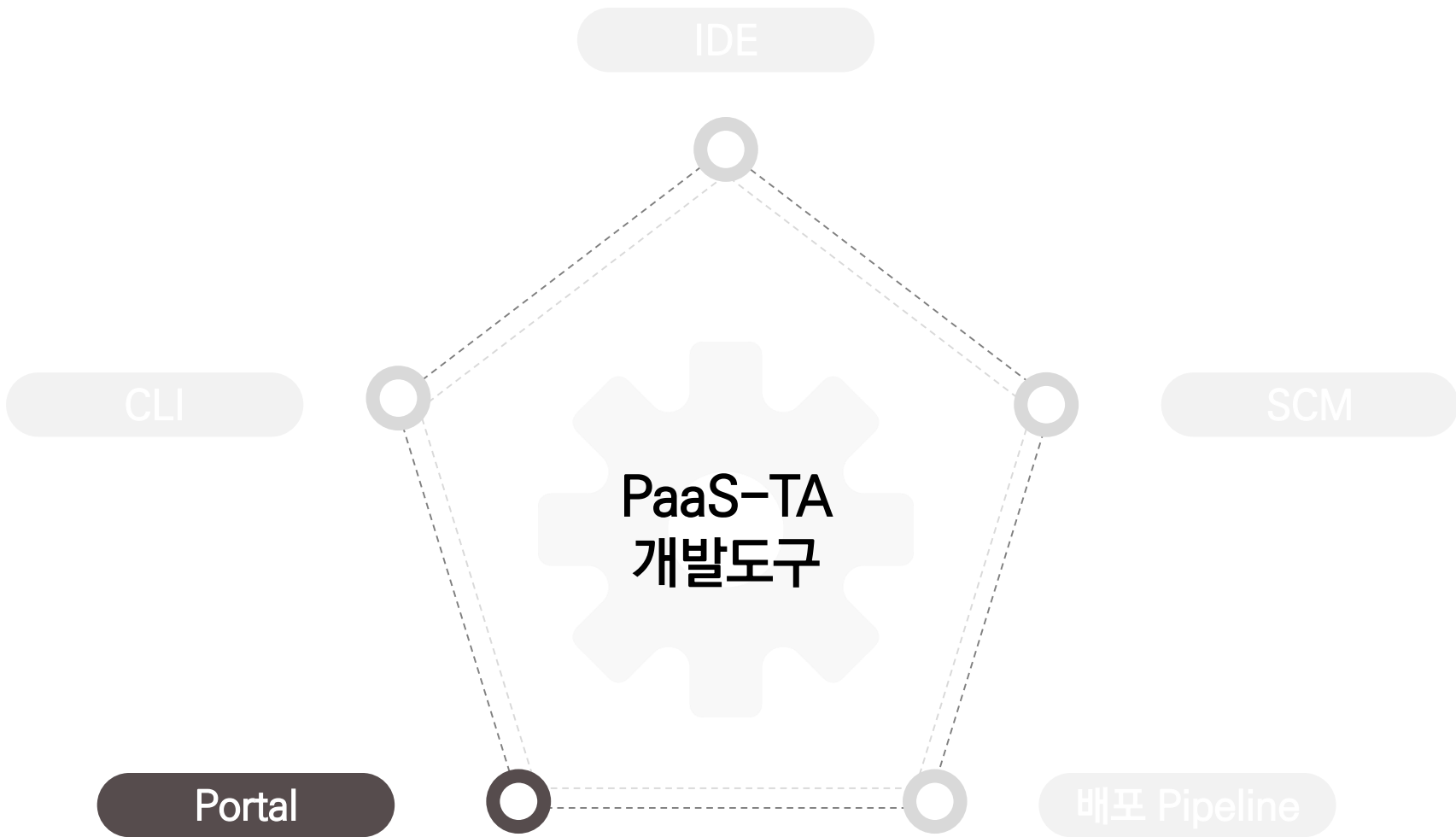
A top-down view of a wooden desk with a laptop, glasses, a coffee cup, and a small plant. The text "PaaS-TA 개발도구 이해 및 실습" is centered on the desk.

PaaS-TA 개발도구 이해 및 실습

PaaS-TA 개발도구



Portal



» 개요


 사용자
포탈

- 애플리케이션의 배포 및 관리, 개발에 필요한 서비스 인스턴스 관리 및 계정관리와 공지사항 같은 포탈 관리 기능을 제공
- 애플리케이션 플랫폼에 배포되는 응용애플리케이션을 대한 모니터링을 제공하여, **응용애플리케이션이 문제가 있는 부분을 포탈 사용자 관리자에게 제공**하여 원활한 운영을 할 수 있게 지원


 운영자
포탈

- CCDB, UAADB, PORTALDB로부터 데이터를 조회하고 클라우드 컨트롤러에서 제공하는 **REST API 호출을 통해 상호작용**
- 웹 사용자 인터페이스 환경에서 **플랫폼을 관리**할 수 있도록 함

» 개요

PaaS-TA 사용자 포탈 메뉴 구성

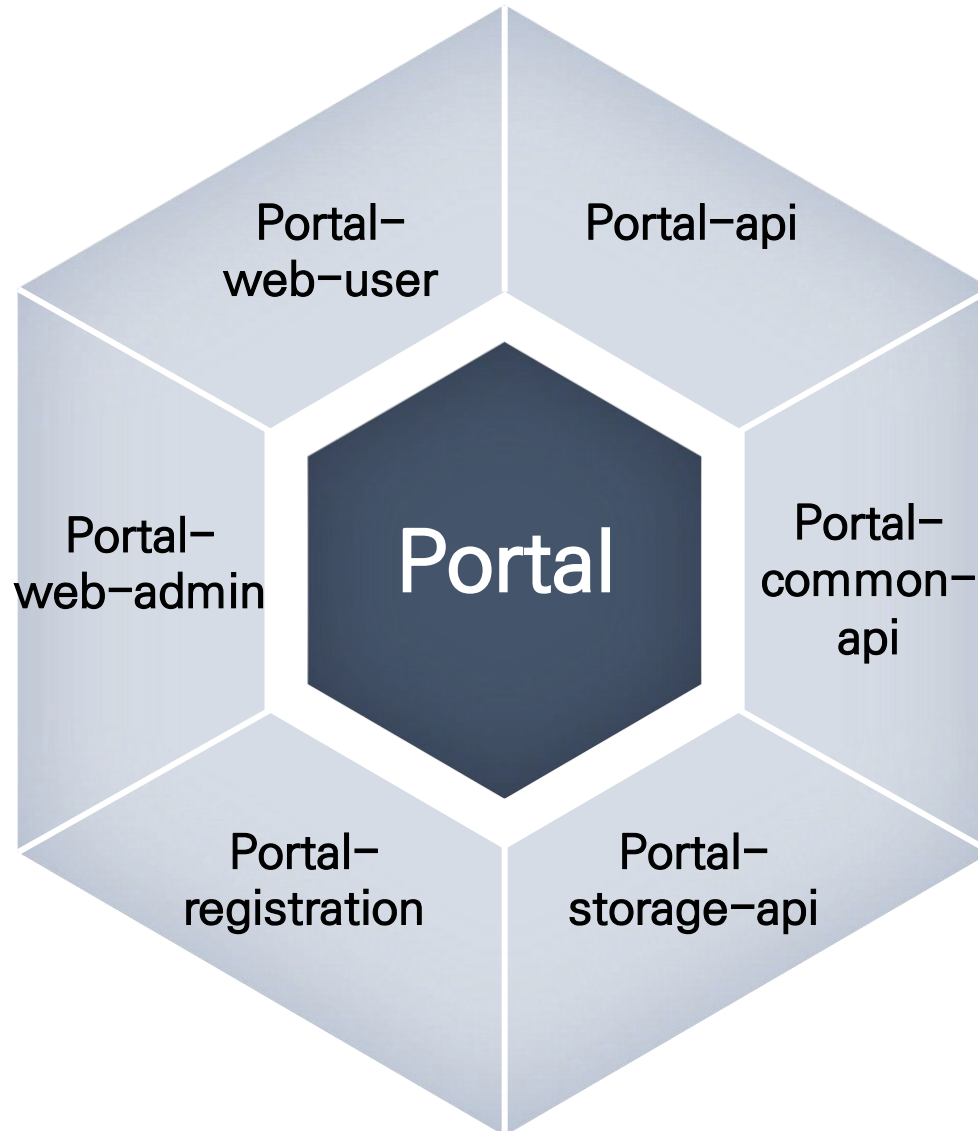
분류	메뉴	설명
Org & Space, 애플리케이션 관리	대시보드	Org & Space 생성 및 애플리케이션, 서비스, 도메인, 사용자 초대 등을 관리
개발환경, 서비스 생성	카탈로그	애플리케이션 개발환경 및 서비스 생성
정보 조회	도움말	외부 도움말 페이지로 이동
	문서	관리자가 등록한 문서 조회
	공지	관리자가 등록한 공지 조회
	커뮤니티	사용자 게시판
	나의 메뉴	내 계정 정보, 내 문의 내역 등을 관리

» 개요

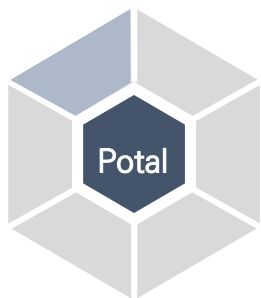
PaaS-TA 운영자 포탈 메뉴 구성

분류	메뉴	설명
Org & Space, APP, 사용자 통계 정보	운영자 포탈 대시보드	대시보드 정보 조회
PaaS-TA 포탈 운영 관리	운영 관리 메뉴	설정 정보 코드관리 조직 및 공간 조회 클라이언트 권한관리 사용자 관리 WEB IDE 유저 관리 모니터링 사이트
PaaS-TA 포탈 관리	포탈 관리	메뉴관리 카탈로그 관리(앱 템플릿/앱 개발환경/서비스) 지원(공지/문의/게시판) 문서관리
PaaS-TA 포탈 서비스 관리	서비스 관리	빌드팩 서비스 브로커

» 개요



» 개요

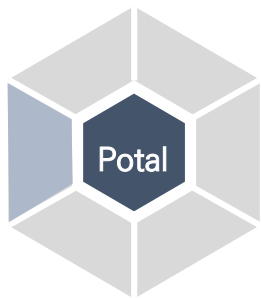


PaaS-ta-portal-webuser (사용자 포탈)

웹 인터페이스를 통해 PaaS-TA에서 개발자 환경을 구성 할 수 있도록 하는 웹 애플리케이션

- ✓ API 애플리케이션들간의 REST API 를 호출하여 그 결과를 화면에 제공
- ✓ 화면을 분리하여 UI 프레임워크 변경 시 API 서버와 영향도가 없이 구성

» 개요

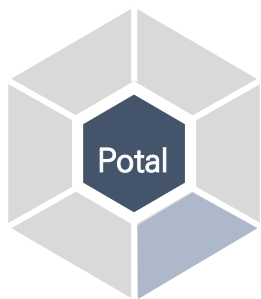


Paas-ta-portal-webadmin (운영자 포탈)

웹 인터페이스를 통해 PaaS-TA 운영 환경의 데이터를 관리할 수 있도록 하는 웹 애플리케이션

- ✓ API 애플리케이션들간의 REST API 를 호출하여 그 결과를 화면에 제공
- ✓ 화면을 분리하여 UI 프레임워크 변경 시 API 서버와 영향도가 없이 구성

» 개요

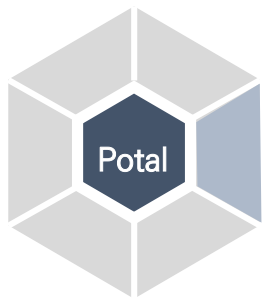


Paas-ta-portal-registration

Spring-cloud 프로젝트의 Service discovery server로 Eureka 를 사용한 애플리케이션

- ✓ 포탈에서 사용하는 애플리케이션을 portal-registration 에 서비스로 등록하여 등록된 서비스들은 registration 서버에 등록된 서비스 이름으로 데이터를 요청
- ✓ 등록된 서비스들의 상태를 확인 할 수 있는 UI를 제공

» 개요

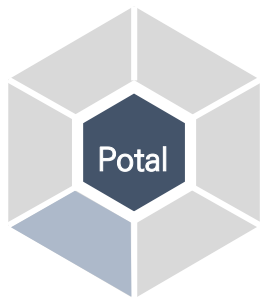


Paas-ta-portal-gateway

Spring-cloud 프로젝트의 Zuul 사용한 애플리케이션

- ✓ End-Point를 통합하는 기능을 제공
- ✓ Spring-cloud Eureka를 사용하지 않는 애플리케이션의 Gateway 역할 제공

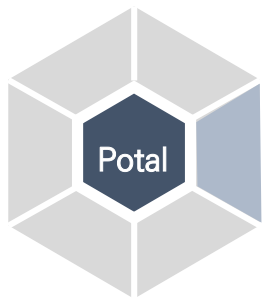
» 개요



Paas-ta-portal-api

PaaS-TA 실행 환경의 Cloud Controller에서 제공하는 최신의 CF 라이브러리를 이용하여, 포탈에서 필요한 REST API를 제공

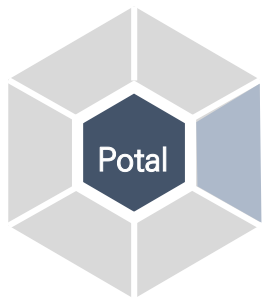
» 개요



Portal-log-api

PaaS-TA 실행 환경의 Cloud Controller에서 제공하는 최신의 CF 라이브러리를 이용하여, CF APP에 대한 로그를 제공

» 개요

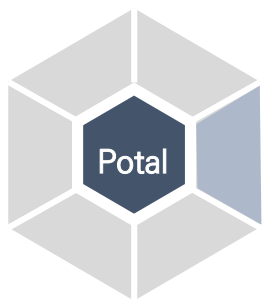


Paas-ta-portal-storage-api

PaaS-TA 실행 환경의 파일 관리하는 기능 제공

- ✓ Object-Storage를 이용하여, 파일을 관리
- ✓ RestAPI를 통하여, 파일 전송 및 파일 수신 관리

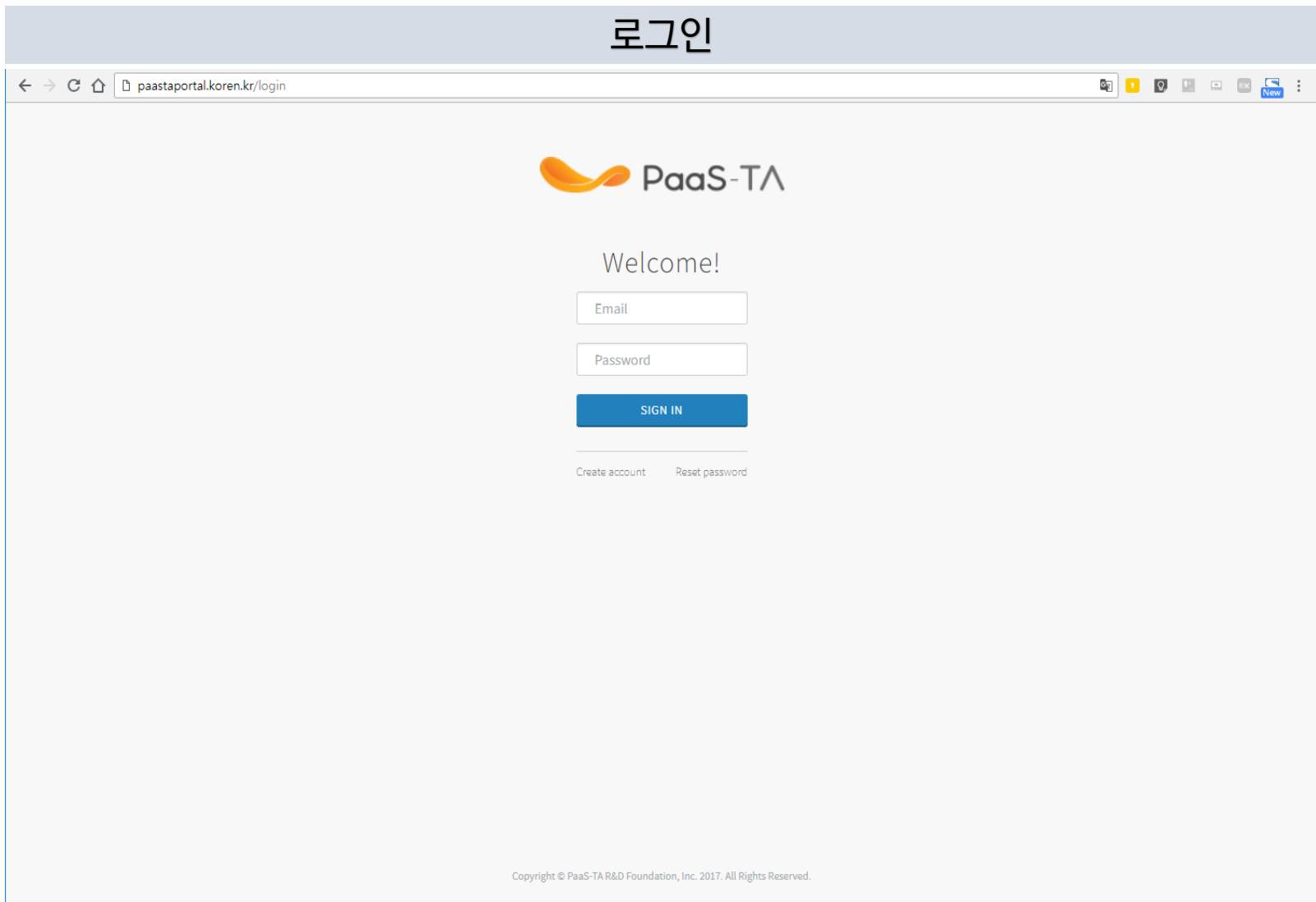
» 개요



Paas-ta-portal-common-api

PaaS-TA 실행 환경의 Spring-cloud jpa를
이용하여, 데이터베이스 관리 기능 제공

» 화면구성



» 대시보드

The dashboard displays the following information:

- Navigation:** PENNE logo, 대시보드 (Dashboard), 카탈로그 (Catalog), 문서 (Documents).
- Filters:** 대한민국 (South Korea), apcc, dev.
- Labels:** 지역 (Region), Org, Space.
- Usage:** 2.50 GB / 할당량 4 GB (63% usage), 할당량 (Quota).
- Application/Service Tabs:** 애플리케이션 (Applications), 서비스 (Services).
- Application Details:**
 - java_test:** 인스턴스 : 1, 디스크 : 2 GB, 메모리 : 1 GB. STARTED. DISK: 7%, Memory: 33%.
 - Py_Test:** 인스턴스 : 1, 디스크 : 1 GB, 메모리 : 512 MB. STARTED. DISK: 15%, Memory: 5%.
 - OpenClimate:** 인스턴스 : 1, 디스크 : 1.50 GB, 메모리 : 1 GB. STARTED. DISK: 9%, Memory: 33%.
- Deployment Status:** 배포 현황 (Deployment Status).

애플리케이션과 서비스의 전체 상황을 파악할 수 있음

» 카탈로그

The screenshot displays the PENNE PaaS-TA development tool interface. At the top, there is a navigation bar with '대시보드', '카탈로그', and '문서' tabs. The '카탈로그' tab is selected. Below the navigation bar, there is a search bar labeled 'Catalog Search' and a '전체보기' (View All) button. The main content area is divided into three sections: '최근작업' (Recent Works), '앱 템플릿' (App Templates), and '앱 개발환경' (App Development Environments). Each section contains several application templates, each with a logo, title, description, and status (e.g., '파스타' or '무료'). A callout box with a checkmark is overlaid on the right side of the interface, containing the text: '서비스와 애플리케이션을 클릭만으로 관리할 수 있음' (You can manage services and applications with a single click).

» 앱 개발환경 - 생성

카탈로그 → 앱 개발환경

대시보드 카탈로그 문서

Catalog 앱 개발환경

전체보기

앱 템플릿

앱 개발환경

Java8 온라인 앱 개발환경

Back

요약

Java8 온라인 앱 개발환경

문서보기

Java8

앱 생성 앱 시작 여부

지역: 대한민국 | 조직: apcc | 공간: dev

메모리: 512 | 디스크: 512

앱 이름: mytestapp

도메인: paasta.koren.kr

현재 이름으로 앱을 생성할 수 있습니다.

현재 호스트로 라우트를 생성할 수 있습니다.

생성 취소

유니크한 이름 설정

- PaaS는 대표 도메인 앞에 앱 이름이 들어감(ex. Playpark.paas-ta.co.kr)
- PaaS-TA는 들어왔던 도메인을 분석해 배포한 컨테이너를 찾음
- 대시보드 Org & Space의 앱과 구분될 수 있도록 중복되지 않는 앱 이름 설정이 필요함

» 앱 개발환경 - 생성 상세 화면

카탈로그 → 앱 개발환경

The screenshot shows a dashboard for managing applications. The main section is titled '앱 상태창' (App Status Window) and displays details for an application named 'mytestapp'. The application is in a 'STARTED' state. Key metrics include 1 instance (14% of a total of 7), 0% CPU usage (0% of 100%), 256 M memory usage (0% of 10G), and 512 M disk usage (0% of 10G). The application is running in the 'koren' namespace on the 'prd' environment. Below the main status window, there are three panels: '이벤트' (Events) showing a list of audit events, '상태' (Status) showing a table of application instances, and '라우트' (Routes) showing the application's endpoint.

앱 상태창

리소스 사용량

INSTANCE 1 / 최대 7개 (14%)

CPU 0% / 100% (0%)

MEMORY 256 M / 최대 10G (0%)

DISK 512 M / 최대 10G (0%)

이벤트

#	Status	Date	Type	Actor
1	+	2018-08-07 08:39:55	audit.app.droplet.create	admin
2	▶	2018-08-07 08:39:37	audit.app.update	admin
3	+	2018-08-07 08:39:36	audit.app.build.create	admin
4	+	2018-08-07 08:39:36	audit.app.upload-bits	admin
5	+	2018-08-07 08:39:35	audit.app.map-route	admin

상태

#	Status	CPU	Memory	Disk
1	STARTING	0.00 %	0 Bytes	0 Bytes

라우트

mytestapp.paasta.koren.kr


바로가기

» 앱 관리

카탈로그 → 앱 개발환경

d ☰ koren > prd

앱 관리



mytestapp

mytestapp 개요 페이지 입니다.

STARTED

■ ↻ ⋮

✎ 이름변경

↻ 재시작

🗑 삭제

INSTANCE

1

 / 최대 7개

⬆ ⬇ 직접입력 저장

MEMORY

256 M

 / 최대 10G

⬆ ⬇ 직접입력 저장

● 지역 ● 조직 ● 공간

대한민국 koren prd

· 앱 URL: URL Link

· 마지막 푸시: 2018-08-07 08:39:36

· 빌드팩: java_buildpack

이벤트 ☰

#	Status	Date	Type	Actor
1	❌	2018-08-07 08:42:17	app.crash	mytestapp
2	⬆	2018-08-07 08:42:17	audit.app.process.crash	web

상태

#	Status	CPU	Memory
1	CRASHED	0.00 %	0 Bytes

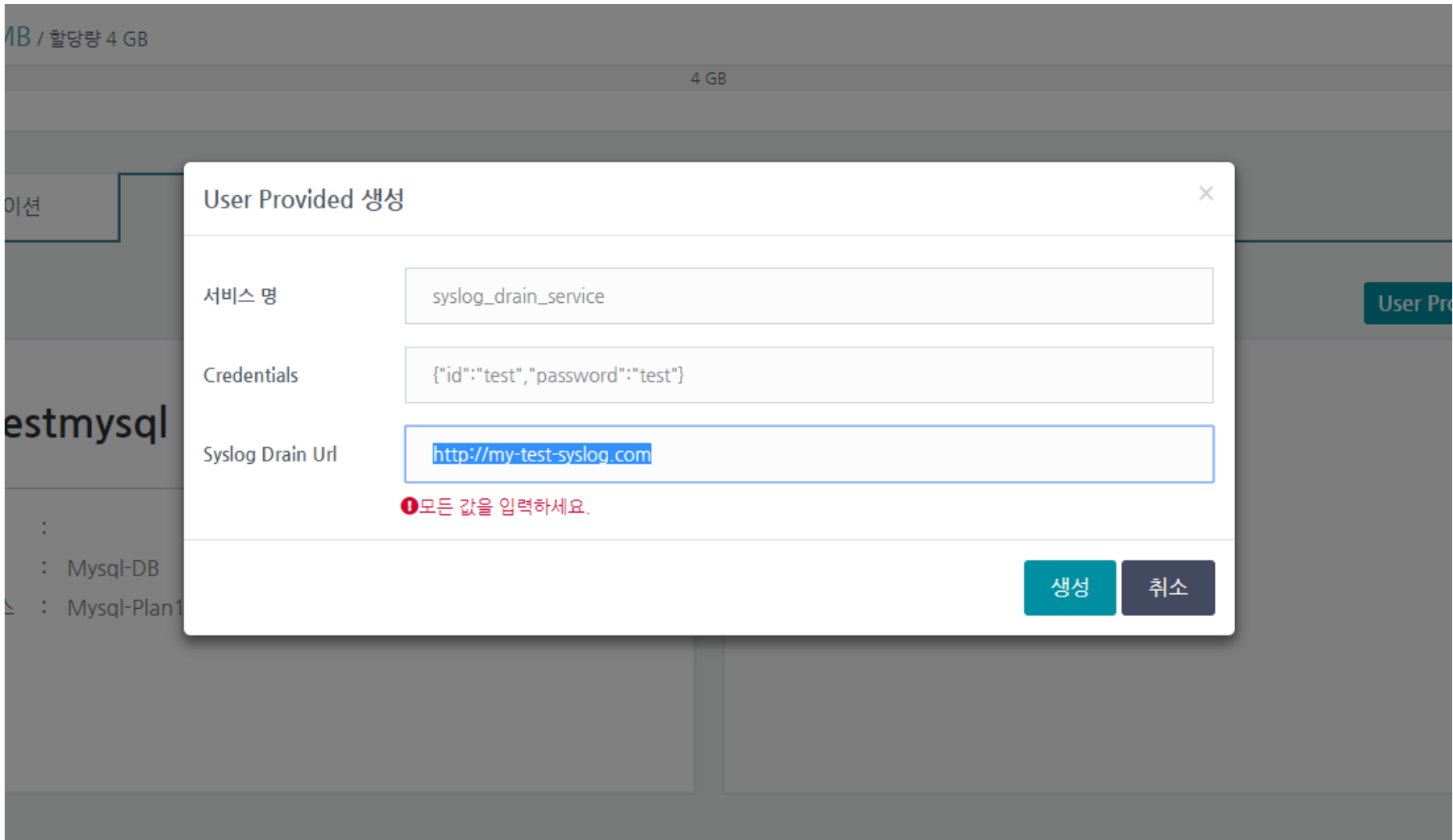
» User Provided 생성

[대시보드](#) → [서비스](#)

The screenshot shows the PENNE portal interface. At the top, there is a navigation bar with '대시보드', '카탈로그', and '문서' menus, and user profile, notification, and language icons. Below the navigation bar, a usage summary shows '256 MB / 할당량 4 GB' with a progress bar indicating 6% usage. A tabbed interface has '애플리케이션' and '서비스' tabs, with '서비스' selected. On the right side, a blue button labeled 'User Provided 생성' is highlighted with an orange border. Below this button, the text '사용자 서비스 정책' is visible. On the left, a service card for 'testmysql' is shown with details: '연결 앱 :', '서비스 명 : Mysql-DB', and '제공서비스 : Mysql-Plan1-10con'. A large plus sign is centered in the main content area.

» User Provided 생성

대시보드 → 서비스



» 서비스 생성

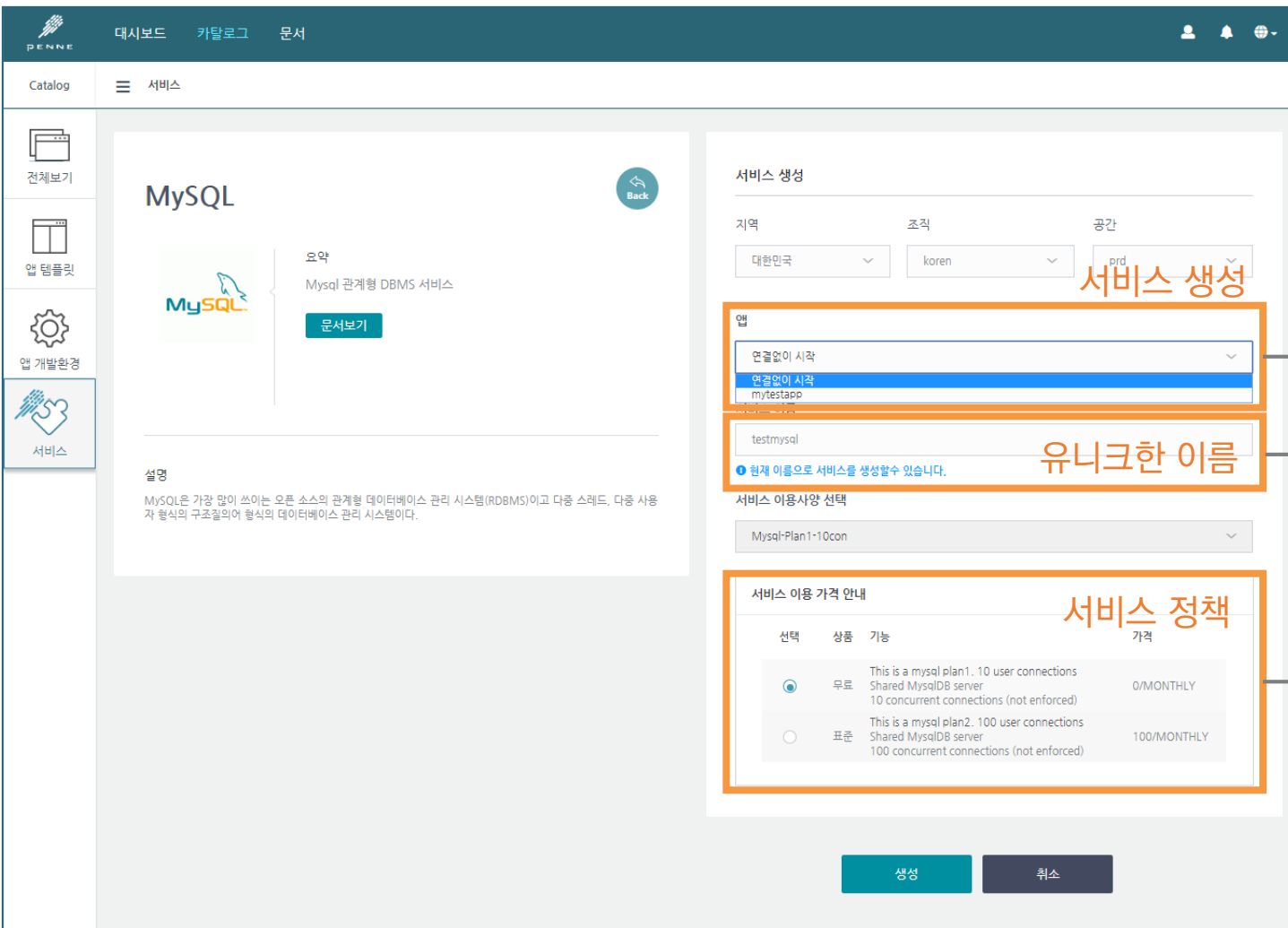
카탈로그 → 서비스

The screenshot shows the PENNE PaaS-TA development tool interface. The top navigation bar includes '대시보드', '카탈로그', and '문서'. The left sidebar contains navigation options: '전체보기', '앱 템플릿', '앱 개발환경', and '서비스'. The main content area is titled '서비스' and displays a grid of service cards. The 'MySQL' service card is highlighted with an orange border, and the text '서비스 생성' is overlaid on it. Other visible service cards include 'Java + MySQL', 'Java8 온라인 앱 개발환경', '형상관리', '파이프라인', 'RabbitMQ', and 'mongoDB'.

Service Name	Description	Price
Java + MySQL	Java Tomcat 환경의 MySQLDB 앱 템플릿	파스타
Java8 온라인 앱 개발환경	Java8 온라인 앱 개발환경	커뮤니티, 무료
형상관리	형상관리	파스타, 무료
파이프라인	개발용으로 만들어진 파이프라인	파스타, 무료
형상관리	형상관리	파스타, 무료
MySQL	MySQL 관계형 DBMS 서비스	파스타, 무료
redis	Redis NoSQL	파스타, 무료
RabbitMQ	RabbitMQ는 AMQP를 구현한 비동기 메시지 큐	파스타, 무료
mongoDB	Mongo-DB 도큐먼트 지향 NoSQL DBMS	파스타, 무료

» 서비스 생성

카탈로그 → 서비스



동일한 dev 안에서 갖고 있는 어떤 앱에 연결할 것인지 설정

서비스 이름이 달라야 함

내 앱을 제외한 모든 것이 서비스이므로 서비스 사용 스펙을 선택해야 함

» 서비스 연결

카탈로그 → 서비스

The screenshot shows the PENNE PaaS-TA management portal. The main content area displays the 'mytestapp' overview, which is in a 'STARTED' state. It includes performance metrics for INSTANCE (1 / 최대 7개, 14%), CPU (0% / 100%), MEMORY (256 M / 최대 10G, 0%), and DISK (512 M / 최대 10G, 0%). Below this, there is a '서비스 연결' (Service Connection) table with a '+ 서비스 연결' button highlighted in an orange box.

이름	제공서비스	서비스명	대시보드	연결해제	Credentials
+ 서비스 연결					

» 서비스 연결

카탈로그 → 서비스

The screenshot shows the PENNE portal dashboard. The top navigation bar includes '대시보드', '카탈로그', and '문서'. The main content area is divided into several sections:

- Dashboard:** Shows 'koren > prd' and a 'mytestapp' overview page with a 'STARTED' status.
- Resource Usage:** Displays metrics for CPU (1 / 최대 7개), Memory (256 M / 최대 10G), and Disk (512 M / 최대 10G).
- Service List:** A table listing services with columns for name, provider, service name, dashboard, connection status, and credentials.

이름	제공서비스	서비스명	대시보드	연결해제	Credentials
testmysql	Mysql-Plan1-10con	Mysql-DB		연결해제	[Edit] [Add]

A red box highlights the service entry for 'testmysql'. A '+ 서비스 연결' button is visible in the top right of the service list section.

서비스 세부 정보

» 서비스 연결

카탈로그 → 서비스

The screenshot shows a 'Credentials' dialog box with the following information:

Hostname	10.2.0.132
jdbcUrl	mysql://6e006f9be887bac7:7630eebbf9af3864@10.2.0.132:3306/op_e817c457_fe0f_4b87_a8de_e59c7c1f2d4e
Name	op_e817c457_fe0f_4b87_a8de_e59c7c1f2d4e
Port	3306
Username	6e006f9be887bac7
Password	7630eebbf9af3864

The background interface shows a service connection configuration table with columns: 이름, 제공서비스, 서비스명, 대시보드, and 연결해제.

이름	제공서비스	서비스명	대시보드	연결해제
testmysql	Mysql-Plan1-10con	Mysql-DB		연결해제

» 서비스 연결 - 환경변수에서 서비스 연결정보 확인

카탈로그 → 서비스

The screenshot shows the 'PENNE' portal dashboard. The left sidebar contains navigation icons for '개요', '이벤트', '상태', '서비스', '환경변수', '라우트', and '로그'. The '서비스' icon is highlighted with an orange box. The main content area shows a breadcrumb 'koren > prd' and a summary of environment variables including '지역: 대한민국', '조직: koren', '공간: prd', '앱 URL', '마지막 푸시: 2018-08-07 08:39:36', and '빌드팩: java_buildpack'. On the right, there are two resource usage cards: 'MEMORY 256 M / 최대 10G' and 'DISK 512 M / 최대 10G'. The central '환경변수' (Environment Variables) section is expanded, showing '사용자 정의 URL' and '시스템 정의' (System Defined) variables. A large grey callout box with a checkmark contains the text '서비스 연결을 통해 클라우드에 생성된 dev 서비스 접근이 가능함'. An orange box highlights the 'system_env_json' section, which contains a JSON object with 'VCAP_SERVICES' and 'Mysql-DB' details. A text label '서비스 연결정보 환경변수' points to this highlighted section. The footer contains the copyright notice: 'Copyright © 2018 PaaS-TA. All rights reserved.'

» 사용자 정의 환경변수 입력

카탈로그 → 서비스

● 지역: 대한민국
● 조직: koren
● 공간: prd

· 앱 URL: [URL Link](#)
 · 마지막 푸시: 2018-08-07 08:39:36
 · 빌드팩: java_buildpack

MEMORY 0%

256 M / 최대 10G

직접입력 저장

DISK 0%

512 M / 최대 10G

직접입력 저장

환경변수 ⌵

사용자 정의 URL + 추가

system_version

1.1.2

저장

취소

시스템 정의

```

"staging_env_json": {},
"system_env_json": {
  "VCAP_SERVICES": {
    "Mysql-DB": [
      {
        "name": "testmysql",
        "instance_name": "testmysql",
        "binding_name": null,
        "credentials": {
          "hostname": "10.2.0.132",
          "password": "7630eebb9af3864",
          "port": "3306",
        }
      }
    ]
  }
}
            
```

사용자 정의 URL

» 환경변수에서 사용자 정의 URL 확인

[카탈로그](#) → [서비스](#)

개요

이벤트

상태

서비스

환경변수

라우트

로그

빌드팩: java_buildpack

256 M / 최대 10G

직접입력 저장

환경변수

사용자 정의 URL

system_version

시스템 정의

```

"version": "97dccd23-65ff-406f-b2c2-da048523da31",
"application_version": "97dccd23-65ff-406f-b2c2-da048523da31"
},
"environment_json": {
  "system_version": "1.1.2"
},
"running_env_json": {},
"staging_env_json": {},
"system_env_json": {
  "VCAP_SERVICES": {
    "Mysql-DB": [
      {
        "name": "testmysql",
        "instance_name": "testmysql"
      }
    ]
  }
}

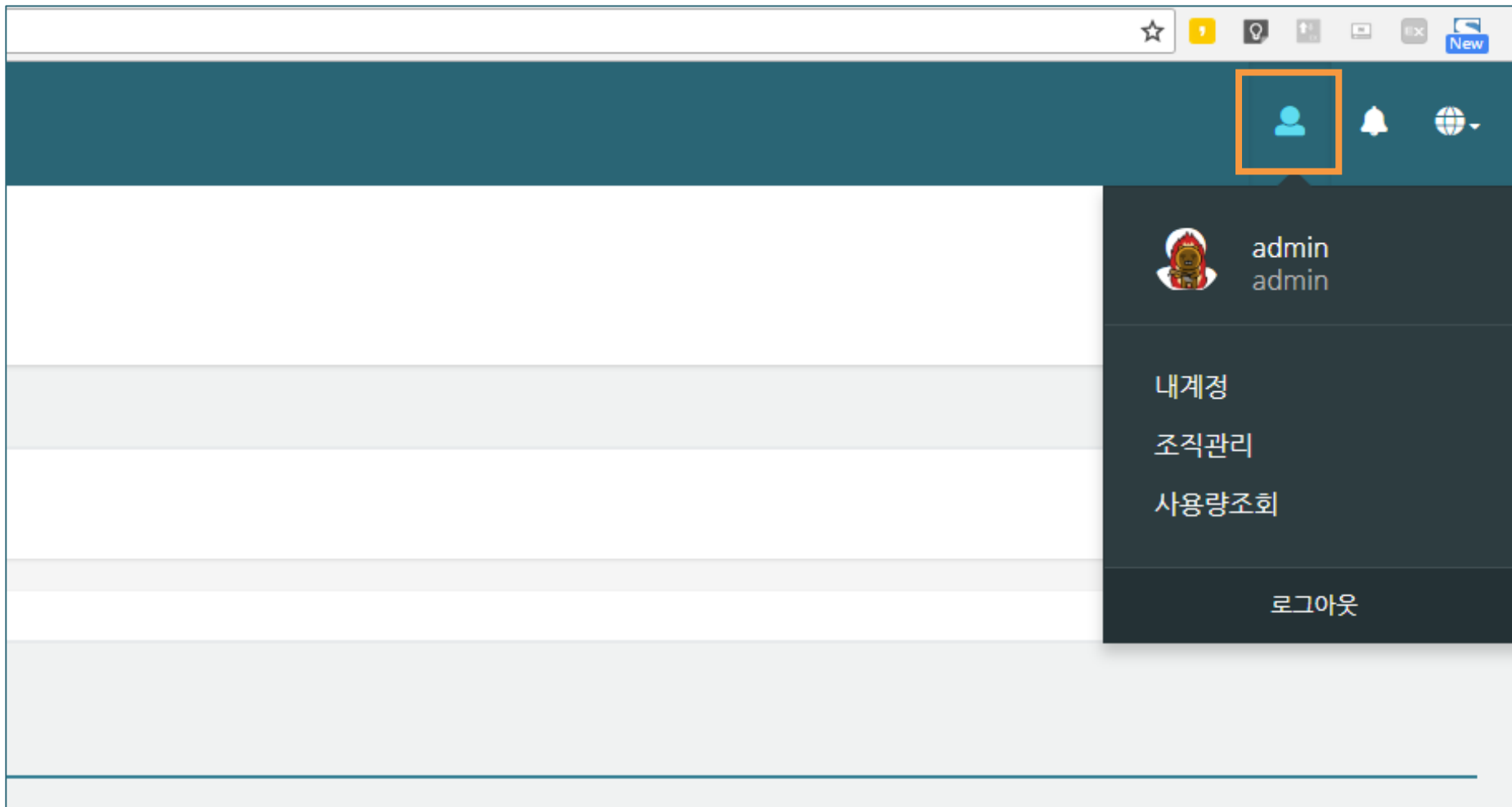
```

256 M / 최대 10G

직접입력 저장

Copyright © 2018 PaaS-TA. All rights reserved

» 계정관리



» 내 계정

계정관리 → 내계정

The screenshot displays the '내 계정' (My Account) page in the PENNE portal. The page is divided into several sections:

- Header:** PENNE logo, navigation links (대시보드, 카탈로그, 문서), and user profile icons.
- Management Menu:** A sidebar menu with '내 계정' (My Account) highlighted, along with '조직관리' (Organization Management) and '로그아웃' (Logout).
- 내 계정 (My Account):**
 - Profile:** A circular profile picture of a cartoon character with a flame-like mane.
 - 로그인 정보 (Login Information):**
 - 이메일 (Email): admin
 - 비밀번호 변경 (Change Password): [Edit]
 - 사용자 정보 (User Information):**
 - 이름 (Name): admin [Edit]
 - 전화번호 (Phone Number): 1231456 [Edit]
 - 우편번호 (Postal Code): 112233 [Edit]
 - 주소 (Address): 서울 특별시 마포구 마포대로 38 2층 202호 [Edit]
 - 나의 조직 (My Organization):**
 - koren [Delete]
 - Kulcloud [Delete]
 - apcc [Delete]

» 조직관리

계정관리 → 조직관리

Management | 대시보드 | 카탈로그 | 문서

내계정 | **조직관리** | 새 조직 추가

공간 1 | + 공간 생성 | 도메인 1 | + 도메인 추가

prd | paasta.koren.kr

사용자 1 | + 사용자 초대

역할설정

admin | OrgManager(조직 관리자) | BillingManager(조직 결제 관리자) | OrgAuditor(조직 참관인) | 멤버 취소

할당량

선택	서비스명	메모리	인스턴스 메모리	라우트	서비스 인스턴스	가격	인스턴스	라우트포트
<input checked="" type="radio"/>	default	4096	2048	10	10	무료	10	10
<input type="radio"/>	10	10	10	10	10	무료	10	10

할당량 수정

» 문서

The image displays two overlapping screenshots. The background screenshot is a web portal interface for 'PENNE'. It features a top navigation bar with '대시보드', '카탈로그', and '문서' (highlighted). A sidebar on the left contains icons for '개요', '이벤트', '상태', '서비스', '환경변수', '라우트', and '로그'. The main content area shows a 'Dashboard' for 'koren > prd' with a 'STARTED' button and a '서비스' (Services) section containing a table:

이름
testmysql

The foreground screenshot is a GitHub repository page for 'PaaS-TA / Guide-3.0-Penne'. It shows repository statistics: 448 commits, 1 branch, 2 releases, and 9 contributors. A commit history table is visible:

Commit Message	Time Ago
leedonghyean Update PaaS-TA_BOSH_CLI_V2_사용자_가이드v1.0.md	2 hours ago
Install-Guide Guide 수정	6 months ago
Service-Guide Update PaaS-TA 형상관리 서비스팩 설치 가이드_v1.0.md	20 days ago
Use-Guide Update PaaS-TA_BOSH_CLI_V2_사용자_가이드v1.0.md	2 hours ago
Download_Page.md Update Download_Page.md	3 months ago
Portal_Page.md Update Portal_Page.md	8 months ago
README.md Update README.md	3 hours ago

Below the table, the README content is partially visible, showing the title 'PaaS-TA3.1 가이드 문서'.

실습

Portal: 샘플 애플리케이션 및 서비스 생성·연동

실습 사이트

playpark.paas-ta.co.kr

실습 소요시간

약 5분

진행 방법

- ① 학생 계정으로 사이트 접속
- ② 서비스를 생성

필요 도구

노트북

실습

Portal: 샘플 애플리케이션 및 서비스 생성·연동

The screenshot shows the PENNE portal interface. At the top, there is a navigation bar with '대시보드', '카탈로그', and '문서' links. Below the navigation bar, there is a 'Catalog' section with a '전체보기' button. The main content area displays a grid of service cards. The cards are organized into two sections: '개발환경' (Development Environment) and '서비스' (Service). The '개발환경' section includes cards for GO, Nodejs, Java8 온라인, 전자정부 프레임워크, Ruby, and Java8 오프라인. The '서비스' section includes cards for 파이프라인, 형상관리, and Mysql DBMS 서비스. The Mysql DBMS 서비스 card is highlighted with a red border. Each card displays the service name, a brief description, and a '파스타' (Pasta) button with a '무료' (Free) label.

Copyright © 2018 PaaS-TA. All rights reserved

설명

카탈로그 페이지에서, Mysql DBMS 서비스를 클릭한다.

실습

Portal: 샘플 애플리케이션 및 서비스 생성·연동

The screenshot shows the PENNE portal interface. The top navigation bar includes '대시보드', '카탈로그', and '문서'. The left sidebar has icons for '전체보기', '앱 템플릿', '앱 개발환경', and '서비스'. The main content area is titled 'Mysql DBMS 서비스' and includes a 'Back' button, a 'MySQL' logo, a '요약' section with 'Mysql 관계형 DBMS 서비스' and a '문서보기' button, and a '설명' section with a paragraph about MySQL. The right sidebar is titled '서비스 생성' and contains a form with dropdown menus for '지역' (대한민국), '조직' (system), and '공간' (dev), a dropdown for '앱' (연결없이 시작), a text input for '서비스 이름' (mysql), a blue notification '현재 이름으로 서비스를 생성할수 있습니다.', a dropdown for '서비스 이용사양 선택' (Mysql-Plan1-10con), and a 'Back' button.

설명

조직 및 공간을 선택한 후, 이름을 입력한다.

실습

Portal: 샘플 애플리케이션 및 서비스 생성·연동

The screenshot displays the PENNE Portal dashboard. At the top, there is a navigation bar with '대시보드', '카탈로그', and '문서' menus, along with user, notification, and globe icons. Below the navigation bar, the 'Dashboard' section shows the current environment: '대한민국', 'system', and 'dev'. A progress bar indicates '0 MB / 할당량 100 GB' with a '100 GB' label. Below the progress bar, there are two tabs: '애플리케이션' and '서비스'. The '서비스' tab is active, showing a 'User Provided' service card for 'Mysql'. The card details are as follows:

- 연결 앱 :
- 서비스 명 : Mysql-DB
- 제공서비스 : Mysql-Plan1-10con

The card also features a large plus sign (+) in the center, indicating a service that can be added or configured.

설명

Mysql 서비스 생성된 것을 확인한다.

실습

Portal: 샘플 애플리케이션 및 서비스 생성·연동

The screenshot displays the PENNE Portal interface. At the top, there is a navigation bar with '대시보드' (Dashboard), '카탈로그' (Catalog), and '문서' (Documents). Below this, a 'Dashboard' section shows '대한민국' (South Korea), 'system', and 'dev' environments. A storage usage bar indicates '0 MB / 할당량 100 GB'. The main content area is divided into '애플리케이션' (Applications) and '서비스' (Services) tabs. Under the '서비스' tab, three service cards are visible: 'mysql', 'pipeline', and 'source-controller'. Each card lists connection details and a '대시보드' (Dashboard) button. A 'User Provided 생성' (Generate User Provided) button is located in the top right of the service area. A fourth empty card with a '+' sign is also present.

Service Name	연결 앱 (Connection App)	서비스 명 (Service Name)	제공서비스 (Provided Service)	대시보드 (Dashboard)
mysql		Mysql-DB	Mysql-Plan1-10con	대시보드
pipeline		delivery-pipeline	delivery-pipeline-shared	대시보드
source-controller		p-paasta-sourcecontrol	Default	대시보드

설명

추가적으로 형상관리, 파이프라인 서비스를 생성한다.

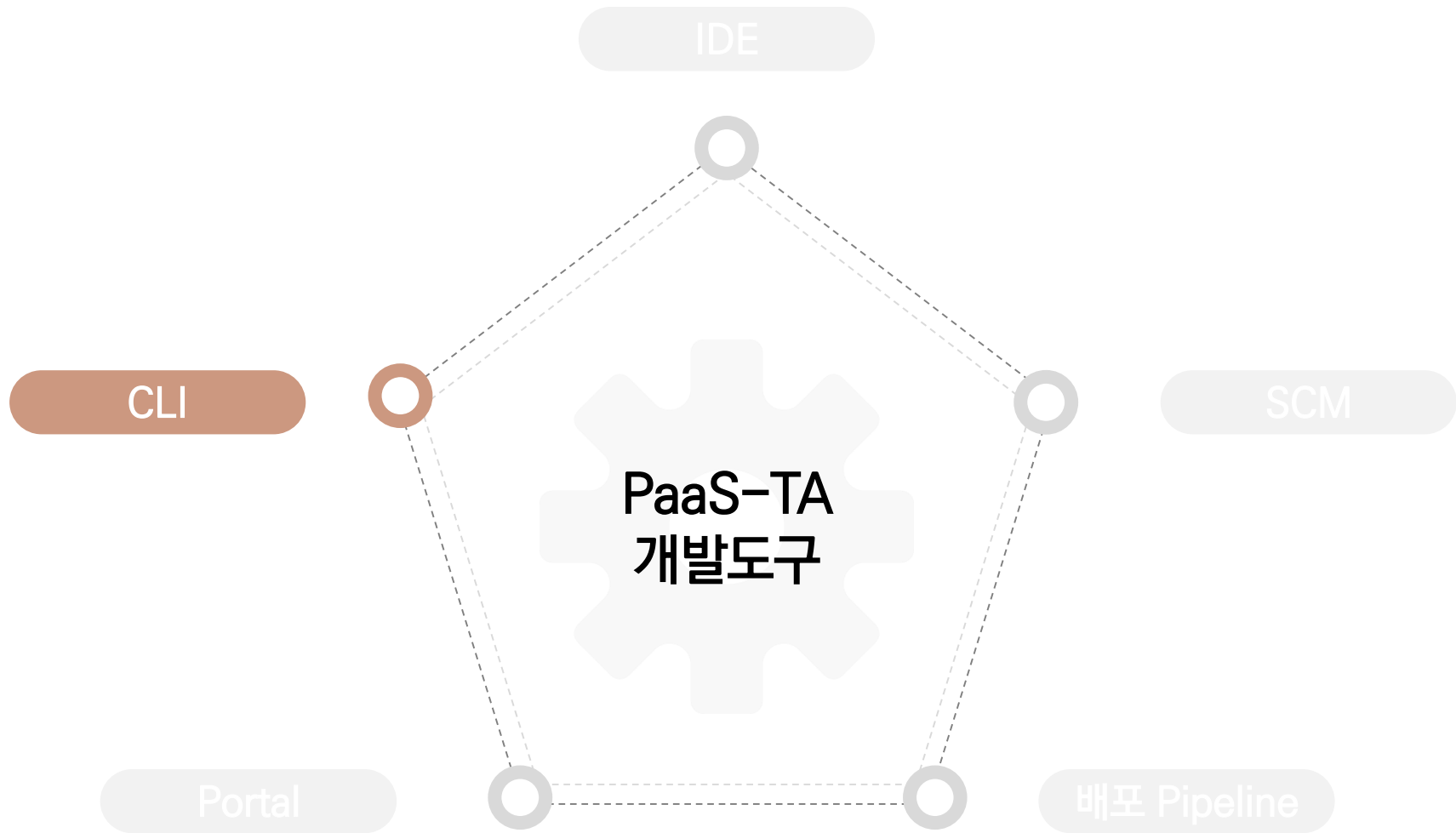
／ 핵심정리 ／

- ✓ Portal에 접속하여 Org & Space의 현황을 확인할 수 있음
- ✓ 대시보드를 통하여 배포된 앱의 현황을 확인 및 관리할 수 있음
- ✓ 카탈로그를 통하여 서비스를 생성, 앱과 연결할 수 있음
- ✓ 기타 앱의 환경변수, 라우트 정보, 로그 등을 확인할 수 있음



MEMO





» 개요

IDE(통합 개발 환경)

OpenPaaS배포와 Release를 관리하기 위해 도움을 주는 커맨드 라인 유틸리티

✓ 기본 Syntax

```
cf [global options] command <arguments...> [command options]
```

command<arguments>
인자는 필수 인자

OpenPaaS 명령어에 대괄호로 묶인
[command options]은 명령어에 따라
선택적으로 사용

» 개요

IDE(통합 개발 환경)

명령어에 따라 **약어**를 제공

예를 들어, App start CLI명령어는 start 이지만 st도 사용 가능

✓ 약어 사용 예시

```
$ cf start  
$ cf st
```

» CLI설치

Browser address bar: <https://cloudfoundry.org/docs-download/>

Navigation: Platform **Get Started** The Foundry Community Case Studies Events Newsroom About Us

Secondary Navigation: Start Here Certified Platforms **Docs & Download** Developer Training Developer Certification Free Community Courses

Docs

Our community maintains docs related to Application Runtime and BOSH – the two **platforms** that comprise Cloud Foundry.

APPLICATION RUNTIME DOCS → BOSH DOCS →

Download

The best way to experience Cloud Foundry is by using a **Certified Provider** offering. If you choose to download Cloud Foundry and run it yourself, here's how.

Installation

» CLI설치

Cloud Foundry Docs

docs.cloudfoundry.org

Cloud Foundry CLI

Direct a CF deployment from your local command line.

[Installing the cf CLI](#)

- [Getting Started with the cf CLI](#)
- [Using the cf CLI with an HTTP Proxy Server](#)
- [Using the cf CLI with a Self-Signed Certificate](#)
- [Using cf CLI Plugins](#)
- [Developing cf CLI Plugins](#)
- [About Starting Applications](#)
- [cf CLI Reference Guide](#)

Develop and Manage Applications

Develop, run, and track applications in the cloud.

- [Considerations for Designing and Running Apps in the Cloud](#)
- [Deploy an Application](#)
- [Deploying a Large Application](#)
- [Deploy an App with Docker](#)
- [Application Container Lifecycle](#)
- [Using Application Health Checks](#)
- [Deploying with Application Manifests](#)

- [Java Buildpack](#)
- [Tips for Java Developers](#)
- [Grails](#)
- [Ratpack](#)
- [Spring](#)
- [Configure Service Connections for Grails](#)
- [Configure Service Connections for Play](#)
- [Configure Service Connections for Spring](#)
- [Cloud Foundry Eclipse Plugin](#)
- [Cloud Foundry Java Client Library](#)

Node.js

- [Node.js Buildpack](#)
- [Tips for Node.js Applications](#)
- [Environment Variables Defined by the Node Buildpack](#)
- [Configure Service Connections for Node.js](#)

Ruby

- [Ruby Buildpack](#)
- [Getting Started Deploying Ruby Apps](#)
- [Getting Started Deploying Ruby on Rails Apps](#)
- [Deploy a Sample Ruby on Rails App](#)
- [Configure Rake Tasks for Deployed Apps](#)
- [Tips for Ruby Developers](#)

» CLI설치

Installing the cf CLI | Cloud Foundry Documentation

docs.cloudfoundry.org/cf-cli/install-go-cli.html

Cloud Foundry Documentation [Get Involved](#)

CLI)

- Installing the cf CLI
- Getting Started with the cf CLI
- Using the cf CLI with an HTTP Proxy Server
- Using the cf CLI with a Self-Signed Certificate
- Using cf CLI Plugins
- Developing cf CLI Plugins
- cf CLI Reference Guide

Information for Operators

- > Deploying Cloud Foundry
- > Administering Cloud Foundry
- > Custom Services
- > Running and Troubleshooting Cloud Foundry
- > Logging and Metrics in Cloud Foundry

Use an Installer

Follow the instructions for your operating system below.

Windows Installation

To use the cf CLI installer for Windows, perform the following steps:

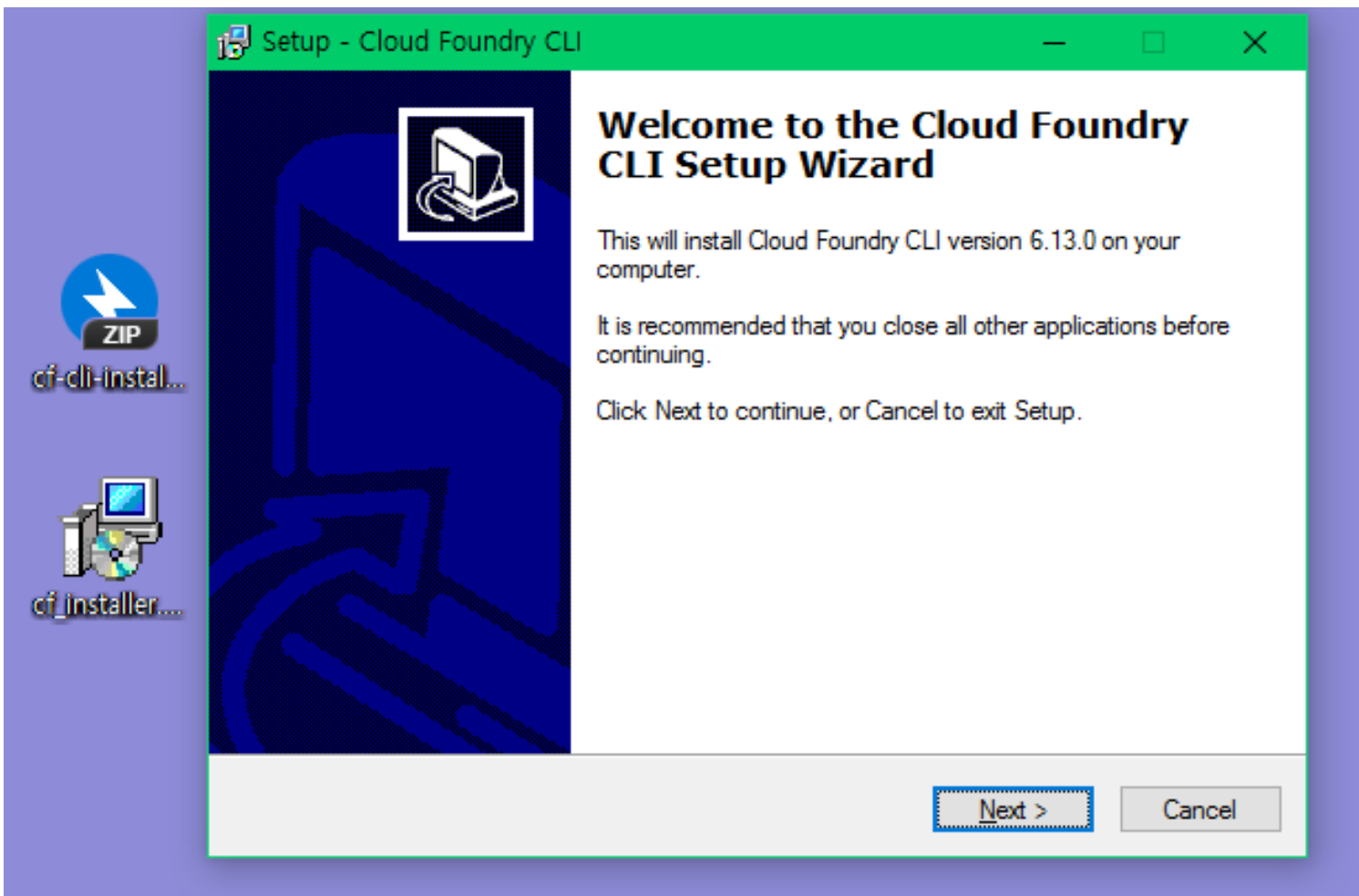
1. Download [the Windows installer](#).
2. Unpack the zip file.
3. Double click the `cf CLI` executable.
4. When prompted, click **Install**, then **Close**.
5. To verify your installation, open a terminal window and type `cf`. If your installation was successful, the cf CLI help listing appears.

Mac OS X Installation

To use the cf CLI installer for Mac OS X, perform the following steps:

1. Download [the OS X installer](#).
2. Open the `.pkg` file.

» CLI설치



» Log into PaaS-TA

Before You Begin

cf CLI 도구는 PaaS-TA API와 통신하는 방법을 알고 있음

cf CLI 도구가 **포맷할 명령을 입력**하고 **PaaS-TA 공급자에게 전송**할 수 있음

PaaS-TA는 처음에는 이러한 명령을 수락하지 않음

사용자는 **개발자권한의 계정으로 앱을 배포 및 관리**할 수 있으며,
이 모든 작업을 위해 **권한을 할당**받아야 함

cf **login 명령**으로 PaaS-TA 접속 정보, 사용자 패스워드 등의 정보로 인증

로그인하면 사용자는 **액세스할 수 있는 org & space에 액세스**할 수 있음

» Log into PaaS-TA

Logging into PaaS-TA from the CLI

CLI 도구는 애플리케이션을 푸시하거나 변경하기 전에 **PaaS-TA에 로그인하고 인증**해야함

이 작업은 **cf login**을 통해 수행되며, 다음과 같은 샘플이 아래에 표시됨

```
$ cf login
API endpoint> https://api.run.pivotal.io
Email> krutten@starkandwayne.com
Password>
Authenticating...
OK
Targeted org starkandwayne
Targeted space development
API endpoint: https://api.run.pivotal.io (API version: 2.75.0)
User:      krutten@starkandwayne.com
Org:      starkandwayne
Space:    development
```

이 출력에서 이메일은 서비스에 등록 할 때 사용한 주소

» Log into PaaS-TA

Logging into PaaS-TA from the CLI

API Endpoint는 PaaS-TA 공급자에 의해 제공되며 일반적으로 <https://api.example.com> 또는 <https://api.subdomain.example.com>의 형태로 제공

- ✓ Pivotal Web Services는 <https://api.run.pivotal.io>를 사용
- ✓ IBM BlueMix는 <https://api.ng.bluemix.net>을 사용
- ✓ SAP Cloud Platform은 <https://api.cf.eu10.hana.ondemand.com> 또는 <https://api.cf.us10.hana.ondemand.com>을 사용
- ✓ Swisscom은 <https://api.lyra-836.appcloud.swisscom.com>을 사용

고급 명령 줄 옵션을 사용하여 로그인하는 방법은 여러 가지가 있지만 이 과정에서는 **cf login**만 필요

» Log into PaaS-TA

More on Your Login

성공적으로 로그인하면 시스템으로 인증되며 사용하는 PaaS-TA 공급자는 액세스할 수 있는 정보와 계정에서 수행할 수 있는 작업을 파악

CLI 도구는 일부 정보, 즉 **PaaS-TA Endpoint API** 및 **인증 시 주어진 토큰(token)**을 저장

로그인할 때 비밀번호를 저장하는 대신 PaaS-TA는 CLI가 저장할 수 있는 **임시 토큰을 생성**

» Log into PaaS-TA

More on Your Login

CLI는 모든 명령에 대해 **이메일과 암호를 다시 묻는 대신 이 토큰을 사용**할 수 있음

토큰은 보통 24 시간 후에 만료되며 CLI는 다시 로그인해야함

이렇게 하면 마지막으로 사용한 API Endpoint가 기억되므로 이제 24 시간 동안 다시 인증하기 위해서는 이메일과 암호만 제공하면 됨

» Get Help from the CLI

The Help Overview

CLI는 명령 줄에서 액세스 할 수 있는 기본 도움말이 함께 제공됨

기본 도움말

- ✓ 일반적으로 필요한 옵션을 찾는 가장 빠르고 쉬운 방법
- ✓ 설치된 버전이 지원하는 것에 대한 확실한 안내서(guid)

`$ cf help` 명령을 통해 볼 수 있음

» Get Help from the CLI

The Help Overview

```
$ cf help
cf version 6.25.0+787326d95.2017-02-28, Cloud Foundry command line tool
Usage: cf [global options] command [arguments...] [command options]

Before getting started:
  config      login,l      target,t
  help,h      logout,lo

Application lifecycle:
  apps,a      run-task,rt  events
  push,p      logs         set-env,se
  start,st    ssh          create-app-manifest
  stop,sp     app
  restart,rs  env,e
  restage,rg  scale

Services integration:
  marketplace,m      create-user-provided-service,cups
  services,s         update-user-provided-service,uups
  create-service,cs  create-service-key,csk
  update-service     delete-service-key,dsk
  delete-service,ds  service-keys,sk
  service           service-key
  bind-service,bs    bind-route-service,brs
  unbind-service,us  unbind-route-service,urs
```

» Get Help from the CLI

More Detailed Help

`help` 명령을 사용하여 필요에 따라 보다 자세한 도움말을 얻을 수 있음

예를 들어, 이미 푸시 한 앱에 사소한 변경을가하는 명령 `cf restage`가 있음

 Note : 이 명령은 `cf rg` 로 축약 될 수 있음

명령 줄에서 다음을 사용할 수 있음

```
$ cf help restage
$ cf restage --help
```

두 명령 모두 `restage` 명령에 대한 도움말과 동일한 결과를 표시

» Get Help from the CLI

More Detailed Help

```
$ cf help restage
NAME:
  restage - Recreate the app's executable artifact using the latest pushed app files and the latest environment (variables, service bindings, buildpack, stack, etc.)

USAGE:
  cf restage APP_NAME

ALIAS:
  rg

ENVIRONMENT:
  CF_STAGING_TIMEOUT=15      Max wait time for buildpack staging, in minutes
  CF_STARTUP_TIMEOUT=5      Max wait time for app instance startup, in minutes

SEE ALSO:
  restart
$
```

환경 변수 쌍을 설정하여 작업을 사용자 정의 할 수 있음

대부분의 경우 도움말은 고려하지 않았을 수 있는 명령에 대한 뉘앙스를 보여줌

» Get Help from the CLI

All Help Available

이 도구에는 초기 cf 도움말 화면에서 제외 되는 추가 명령이 있음

example

```
$ cf help -a
```

-a의 옵션은 CF CLI가 실행할 수 있는 모든 명령을 나열

» 주요 CLI 명령어 이해

분류	명령어	역할
GETTING STARTED	login	OpenPaaS에 로그인 하기 위한 명령어
	logout	cf에 logout
APPS	push	App을 OpenPaaS에 배포하고 app을 Start
	delete	App을 삭제
	start	App을 기동
	stop	App을 중지
	restart	App을 재기동
	restage	App을 restage
SERVICES	marketplace	cf 마켓플레이스에서 제공하는 서비스 목록을 조회
	create-service	마켓플레이스에서 제공하는 서비스로 서비스 인스턴스를 만들
	delete-service	서비스 인스턴스를 삭제
	bind-service	App과 서비스 인스턴스를 바인딩
	unbind-service	App과 서비스 인스턴스를 언바인딩
ROUTES	routes	현재 조직/스페이스에 존재하는 라우트 정보목록을 조회
	delete-orphaned-routes	App에 매핑되지 않은 라우트 정보를 모두 삭제

» 주요 CLI 명령어 이해

login(1/3)

기본 Syntax

```
$ cf login [-a API_URL] [-u USERNAME] [-p PASSWORD] [-o ORG]
[-s SPACE]
```

설명

OpenPaaS에 로그인 하기 위한 명령어

파라미터

파라미터명	설명	필수
-a API_URL	CLI가 접속 하려는 OpenPaaS URL Ex) https://api.10.244.0.34.xip.io	X
-u USERNAME	OpenPaaS에 접속하는 사용자 id	X
-p PASSWORD	OpenPaaS에 접속하는 사용자 password	X
-o ORG	OpenPaaS에 접속하는 사용자 소속조직 명	X
-s SPACE	OpenPaaS에 접속하는 사용자 소속조직 스페이스명	X

» 주요 CLI 명령어 이해

login(2/3)

사용예시

```
# 파라미터 지정한 경우
```

```
$ cf login --skip-ssl-validation -a https://api.10.244.0.34.xip.io -u  
admin -p admin -o crossent -s development
```

```
# 파라미터 지정하지 않을 경우
```

```
$ cf login
```

```
API endpoint: https://api.10.244.0.34.xip.io
```

```
Email> admin
```

```
Password>
```

```
Authenticating...
```

```
OK
```

```
Targeted org crossent
```

» 주요 CLI 명령어 이해

```
login(3/3)
```

사용예시

```
Select a space (or press enter to skip):
```

1. development
2. staged
3. oper

```
Space> 1
```

```
Targeted space development
```

```
API endpoint: https://api.10.244.0.34.xip.io (API version: 2.29.0)
```

```
User: admin
```

```
Org: crossent
```

```
Space: development
```

» 주요 CLI 명령어 이해

login(3/3)

```
D:\W>cf login -a https://api.paas-ta.co.kr --skip-ssl-validation
```

```
API endpoint: https://api.paas-ta.co.kr
```

```
Email> sang.han@infranicmail.com
```

```
Password>
```

```
Authenticating...
```

```
OK
```

```
Select an org (or press enter to skip):
```

```
1. sangyeon
```

```
2. test
```

```
Org> 1
```

```
Targeted org sangyeon
```

```
Targeted space dev
```

```
API endpoint: https://api.paas-ta.co.kr (API version: 2.94.0)
```

```
User: sang.han@infranicmail.com
```

```
Org: sangyeon
```

```
Space: dev
```

로그인 명령어

```
$ cf login -a
https://api.paas-
ta.co.kr --skip-ssl-
validation
```

로그인할 때 인증키가 아닌
아이디와 패스워드를 이용해
접근하기 위해서는 이 옵션을
넣어야 함

» 주요 CLI 명령어 이해

```
logout
```

기본 Syntax

```
$ cf logout
```

설명

```
cf에 logout
```

사용예시

```
$ cf logout
```

» 주요 CLI 명령어 이해

Apps

기본 Syntax

\$ cf apps

설명

앱을 확인

사용예시



```

C:\WINDOWS\system32\cmd.exe
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>cf apps
Getting apps in org sangyeon / space dev as sang.han@infranicmail.com...
OK
No apps found

```

» 주요 CLI 명령어 이해

Services

기본 Syntax

\$ cf services

설명

서비스를 확인

사용예시

```
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
D:\W>
```

```
D:\W>cf services_
```


» 주요 CLI 명령어 이해

push(1/3)

기본 Syntax

```
$ cf push <APP_NAME> [-b BUILDPACK_NAME] [-c COMMAND] [-d DOMAIN] [-f MANIFEST_PATH] [-i NUM_INSTANCES] [-k DISK] [-m MEMORY] [-n HOST] [-p PATH] [-s STACK] [-t TIMEOUT] [--no-hostname] [--no-manifest] [--no-route] [--no-start
```

설명

App을 OpenPaaS에 배포하고 app을 Start

파라미터

파라미터명	설명	필수
APP_NAME	push하는 App명(지정하지 않으면 디렉토리명)	O
-b BUILDPACK	custom빌드팩 URL ex) https://github.com/OpenPaaS-RnD/egov-java-buildpack	X
-c COMMAND	App start command	X
-d DOMAIN	App 도메인	X

» 주요 CLI 명령어 이해

push(2/3)

파라미터

파라미터명	설명	필수
-f MANIFEST_PATH	Manifest 파일 경로	X
-i NUM_INSTANCES	App 인스턴스 갯수	X
-m MEMORY	인스턴스 메모리 용량	X
-k DISK	디스크 사용 용량	X
-n HOST	호스트명 ex) my-subdomain)	X
-p PATH	App의 디렉토리 경로 또는 App file(zip,war등)경로	X
-s STACK	App이 실행되는 운영체제 파일시스템(default: cflinuxfs2)	X
-t TIMEOUT	App이 실행되는동안 CLI가 대기하는 timeout시간	X
--no-hostname	App에 root 도메인을 매핑	X

» 주요 CLI 명령어 이해

push(3/3)

파라미터

파라미터명	설명	필수
--no-manifest	Manifest 파일을 무시	X
--no-route	Push된 앱에 라우트 정보를 삭제하고 App에 라우트 정보를 매핑하지 않음	X
--no-start	App을 push하고 Start하지 않음	X
--random-route	App에게 라우트 정보를 랜덤하게 생성	X

사용예시

```
$ cf push spring-music
```

» 주요 CLI 명령어 이해

delete

기본 Syntax

```
$ cf delete <APP_NAME> [--f] [--r]
```

설명

App을 삭제

파라미터

파라미터명	설명	필수
APP_NAME	APP명	O
--f	확인 없이 App 삭제	X
--r	App에 매핑된 라우트 정보 삭제	X

사용예시

```
$ cf delete spring-music
```

» 주요 CLI 명령어 이해

```
start
```

기본 Syntax

```
$ cf start <APP_NAME>
```

설명

App을 기동

파라미터

파라미터명	설명	필수
APP_NAME	APP명	0

사용예시

```
$ cf start spring-music
```

» 주요 CLI 명령어 이해

stop

기본 Syntax

```
$ cf stop <APP_NAME>
```

설명

App을 중지

파라미터

파라미터명	설명	필수
APP_NAME	APP명	O

사용예시

```
$ cf stop spring-music
```

» 주요 CLI 명령어 이해

restart

기본 Syntax

```
$ cf restart <APP_NAME>
```

설명

App을 재기동

파라미터

파라미터명	설명	필수
APP_NAME	APP명	O

사용예시

```
$ cf restart spring-music
```

» 주요 CLI 명령어 이해

restage

기본 Syntax

```
$ cf restage <APP_NAME>
```

설명

App을 restage함 (환경변수 설정 또는 서비스 바인딩시 사용)

파라미터

파라미터명	설명	필수
APP_NAME	APP명	0

사용예시

```
$ cf restage spring-music
```


» 주요 CLI 명령어 이해

marketplace

기본 Syntax

```
$ cf marketplace [-s SERVICE_NAME]
```

설명

cf 마켓플레이스에서 제공하는 서비스 목록을 조회

파라미터

파라미터명	설명	필수
-s SERVICE_NAME	서비스의 plan이 조회됨	X

사용예시

```
$ cf create-app-manifest spring-music -p ./spring-music-manifest.yml
```

» 주요 CLI 명령어 이해

create-service

기본 Syntax

```
$ cf create-service <SERVICE> <PLAN> <SERVICE_INSTANCE>
[-c PARAMETERS_AS_JSON] [-t TAGS]
```

설명

마켓플레이스에서 제공하는 서비스로 서비스 인스턴스를 생성

파라미터

파라미터명	설명	필수
SERVICE	마켓플레이스에 있는 서비스명	O
PLAN	서비스 플랜명	O
SERVICE_INSTANCE	만들 서비스 인스턴스명	O
-c PARAMETERS_AS_JSON	서비스 설정정보를 json 형태로 입력 Ex) -c '{"ram_gb":4}'	X
-t TAGS	서비스 인스턴스 태그	X

사용예시

```
$ cf create-service spring-music-db silver p-mysql
```

» 주요 CLI 명령어 이해

delete-service

기본 Syntax

```
$ cf delete-service SERVICE_INSTANCE [-f]
```

설명

서비스 인스턴스를 삭제

파라미터

파라미터명	설명	필수
SERVICE_INSTANCE	서비스 인스턴스 명	O
-f	삭제 확인 메시지 없이 서비스 인스턴스 삭제합니다.	X

사용예시

```
$ cf delete-service spring-music-db
```

» 주요 CLI 명령어 이해

bind-service

기본 Syntax

```
$ cf bind-service <APP_NAME> <SERVICE_INSTANCE> [-c
PARAMETERS_AS_JSON]
```

설명

App과 서비스 인스턴스를 바인딩

파라미터

파라미터명	설명	필수
APP_NAME	APP명	O
SERVICE_INSTANCE	서비스 인스턴스 명	O
-c PARAMETERS_AS_JSON	바인딩 설정 파라미터(json형태)	X

사용예시

```
$ cf bind-service spring-music spring-music-db -c '{"permissions":"read-only"}'
$ cf bind-service spring-music spring-music-db -c
~/workspace/tmp/instance_config.json
```

» 주요 CLI 명령어 이해

unbind-service

기본 Syntax

```
$ cf unbind-service <APP_NAME> <SERVICE_INSTANCE>
```

설명

App과 서비스 인스턴스를 언바인딩

파라미터

파라미터명	설명	필수
APP_NAME	APP명	O
SERVICE_INSTANCE	서비스 인스턴스 명	O

사용예시

```
$ cf bind-service spring-music spring-music-db
```

» 주요 CLI 명령어 이해

```
routes
```

기본 Syntax

```
$ cf routes
```

설명

현재 조직/스페이스에 존재하는 라우트 정보 목록을 조회

사용예시

```
$ cf routes
```

» 주요 CLI 명령어 이해

delete-orphaned-routes

기본 Syntax

```
$ cf delete-orphaned-routes [-f]
```

설명

App에 매핑되지 않은 라우트 정보를 모두 삭제

파라미터

파라미터명	설명	필수
-f	삭제 확인메시지 없이 라우트 정보를 삭제	X

사용예시

```
$ cf delete-orphaned-routes
```

» CLI로 배포 하기

Floating domain 앞에
들어갈 구분자 도메인

앱이 사용 할 instance 개수

배포하고자 하는 파일의 위치를 명시
배포를 위한 cf 명령어는 “cf push”
이때 manifest.yml 파일이 있는
위치에서 실행

```
---
applications:
- name: test01
  host: test01
  memory: 1024M
  instances: 1
  random-route: false
  path: ./test01.war
```

배포 된 앱의 이름

앱의 최대 사용가능 메모리

true값을 주면 URL을 랜덤으로 생성 여부

CF 명령어로 배포를 하기 위해 **manifest.yml**를 작성

» CLI로 배포 하기

```

C:\Users\spring\Desktop\JJ\test01 >cf push
Using manifest file C:\Users\spring\Desktop\JJ\test01\manifest.yml

Creating app test01 in org cloudpack / space devpack04 as devpack04...
OK

Using route test01.devpack.co.kr
Binding test01.devpack.co.kr to test01...
OK

Uploading test01...
Uploading app files from: C:\Users\spring\Desktop\JJ\test01\test01.war
Uploading 1.2K, 4 files
Done uploading
OK

```

배포 명령어

\$ cf push

✓ 명세서를 자동으로 읽어서 호스트 이름으로 해당 파일이 업로드 됨

```

requested state: started
instances: 1/1
usage: 1G x 1 instances
urls: test01.devpack.co.kr
last uploaded: Sat May 14 08:10:50 UTC 2016
stack: cflinuxfs2
buildpack: java-buildpack=v3.6-https://github.com/cloudfoundry/java-buildpack.git#5194155 open-jdk-like-jre=1.8.0_91-unlimited-crypto open-jdk-like-memory-calculator=2.0.2_RELEASE tomcat-access-logging-support=2.5.0_RELEASE tomcat-instance=8.0.33 tomcat-lifecycle...

```

state	since	cpu	memory	disk	de
tails					
#0	running	2016-05-14 05:11:41 PM	13.8%	141.5M of 1G	128.8M of 1G

배포 결과 **running**이 확인되면 배포 완료!

» CLI로 서비스 생성 및 연동

```

C:\#>cf marketplace
Getting services from marketplace in org cloudpack / space systeer-dev as systeer-dev...
OK

service      plans      description
mysql        5gb*, 1gb*, 10gb*  MySQL databases on demand
rabbitmq     standard*  RabbitMQ is a robust and scalable high-performance multi-protocol
messaging broker.
redis-100mb  shared-vm*  Redis service to provide a key-value store
redis-250mb  shared-vm*  Redis service to provide a key-value store
redis-30mb   shared-vm*  Redis service to provide a key-value store

* 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.
C:\#>

```

Service 리스트
확인 명령어

\$ cf marketplace

» CLI로 서비스 생성 및 연동

```

C:\#>cf create-service mysql 1gb test_mysql_db
Creating service instance test_mysql_db in org cloudpack / space systeer-dev as systeer-dev...
OK

Attention: The plan `1gb` of service `mysql` is not free. The instance `test_mysql_db` will incur a
cost. Contact your administrator if you think this is in error.

C:\#>cf services
Getting services in org cloudpack / space systeer-dev as systeer-dev...
OK

name          service  plan  bound apps  last operation
test_mysql_db  mysql    1gb   create succeeded

C:\#>

```

Service 생성 및
확인 명령어

\$ cf create-service
서비스명 플랜명
인스턴스명

\$ cf services

» CLI로 서비스 생성 및 연동

```

관리자-명령 프롬프트
C:\#>cf bind-service welcome_cf test_mysql_db
Binding service test_mysql_db to app welcome_cf in org cloudpack / space systeer-dev as systeer-dev...
OK
TIP: Use 'cf restage welcome_cf' to ensure your env variable changes take effect

C:\#>cf services
Getting services in org cloudpack / space systeer-dev as systeer-dev...
OK

name          service  plan  bound apps  last operation
test_mysql_db  mysql    1gb   welcome_cf  create succeeded

C:\#>

```

App 연결 및
연결 확인 명령어

\$ cf bind-service
[애플리케이션 명]
[서비스 인스턴스 명]

\$ cf services

» CLI로 서비스 생성 및 연동

System-Provided:

```

{
  "VCAP_SERVICES": {
    "Mysql-DB": [
      {
        "credentials": {
          "hostname": "133.186.153.136",
          "name": "op_52a8f81f_e6e7_4cd4_bb99_08467434e65f",
          "password": "69ff076fb8899e6b",
          "port": "3306",
          "uri": "mysql://4c04c02bb83c66b6:69ff076fb8899e6b@133.186.153.136:3306/op_52a8f81f_e6e7_4cd4_bb99_08467434e65f",
          "username": "4c04c02bb83c66b6"
        },
        "label": "Mysql-DB",
        "name": "mydbtest",
        "plan": "Mysql-Plan1-10con",
        "provider": null,
        "syslog_drain_url": null,
        "tags": [
          "mysql",
          "document"
        ]
      }
    ]
  }
}

```

환경변수에서
서비스 확인 가능 명령어

`$ cf env <APP_NAME>`

» CLI로 서비스 생성 및 연동

```

C:\WINDOWS\system32\cmd.exe - cf restage welcome-cf

D:\WcloudWPaaS\Wcloudfoundry\Wcf_test2\welcome-cf>cf restage welcome-cf
Restaging app welcome-cf in org sangyeon / space dev as sang.han@infranicsmail.com...

Staging app and tracing logs...
-----> Java Buildpack Version: v3.19 | https://github.com/cloudfoundry/java-buildpack.git#727297d
-----> Downloading Open Jdk JRE 1.8.0_181 from https://java-buildpack.cloudfoundry.org/openjdk/trusty/x86_64/openjdk-1.8.0_181.tar.gz (found in cache)
Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (0.9s)
-----> Downloading Open JDK Like Memory Calculator 2.0.2_RELEASE from https://java-buildpack.cloudfoundry.org/memory-calculator/trusty/x86_64/memory-calculator-2.0.2_RELEASE.tar.gz (found in cache)
Memory Settings: -Xss349K -Xmx681574K -XX:MaxMetaspaceSize=104857K -Xms681574K -XX:MetaspaceSize=104857K
  
```

필요 App restage 명령어
(환경변수 설정 또는 서비스 바인딩시 사용)

\$ cf restage <APP_NAME>

» CLI로 서비스 생성 및 연동

```

C:\Windows\system32\cmd.exe
JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR
-XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh
$CALCULATED_MEMORY
-Djava.ext.dirs=$PWD/.java-buildpack/container_security_provider:$PWD/.java-buildpack/
open_jdk_jre/lib/ext
-Djava.security.properties=$PWD/.java-buildpack/security_providers/java.security
-Djava.security.egd=file:///dev/urandom
-Daccess.logging.enabled=false
-Djava.endorsed.dirs=$PWD/.java-buildpack/tomcat/endorsed
-Dhttp.port=$PORT" exec
$PWD/.java-buildpack/tomcat/bin/catalina.sh
run

#0 state since cpu memory disk details
running 2018-08-16T07:05:53Z 0.1% 218.4M of 1G 131.3M of 1G

D:\Cloud\WPaaS\Cloudfoundry\cf_test2\welcome-cf>

```

필요 App restage 명령어
(환경변수 설정 또는 서비스 바인딩시 사용)

\$ cf restage <APP_NAME>

» CLI로 서비스 생성 및 연동

주의사항

해당 서비스를 생성하고 바인딩하면 환경변수를 통해
해당 접속정보를 확인할 수 있음

이 접속정보를 틀을 이용해 데이터를 컨트롤 할 수 있음

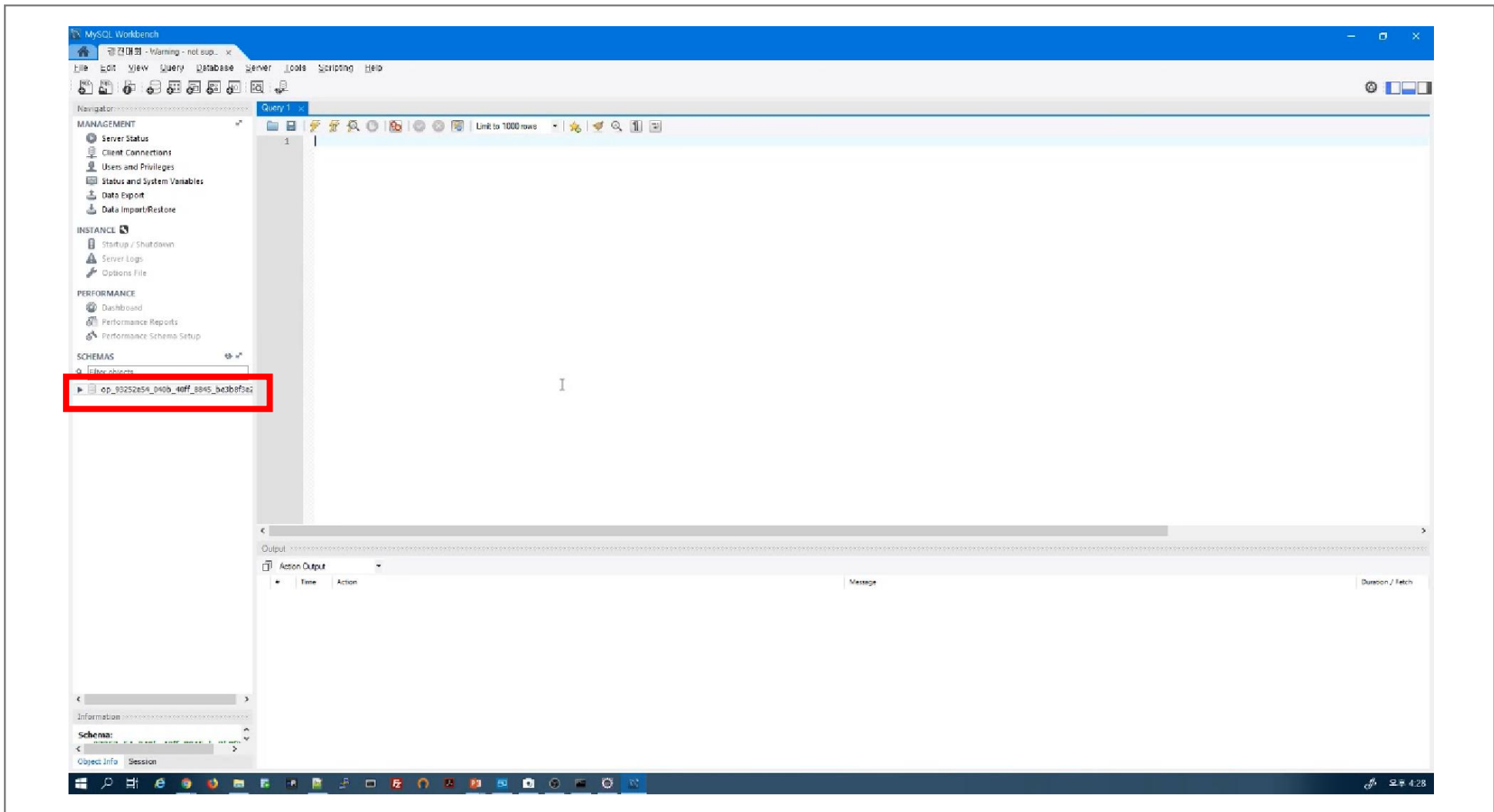
바인딩을 새로 할 때마다 정보가 새로 생성되므로 한 번 바인딩한
이후에는 끊지 않는 것이 좋음

» CLI로 서비스 생성 및 연동

```
C:\WINDOWS\system32\cmd.exe
OK

System-Provided:
{
  "VCAP_SERVICES": {
    "Mysql-DB": [
      {
        "credentials": {
          "hostname": "133.186.153.136",
          "name": "op_93252e54_040b_40ff_8845_be3b8f3e2357",
          "password": "b88e966616b4b2f1",
          "port": "3306",
          "uri": "mysql://6fd89178eb7c288b:b88e966616b4b2f1@133.186.153.136:3306/op_93252e54_040b_40ff_8845_be3b8f3e2357",
          "username": "6fd89178eb7c288b"
        },
        "label": "Mysql-DB",
        "name": "mydbtest",
        "plan": "Mysql-Plan1-10con",
        "provider": null,
        "syslog_drain_url": null,
        "tags": [
          "mysql",
          "document"
        ],
        "volume_mounts": []
      }
    ]
  }
}
```

» CLI로 서비스 생성 및 연동



» CLI로 서비스 생성 및 연동

```

C:\WINDOWS\system32\cmd.exe
OK

name           requested state   instances  memory  disk  urls
mytestapp     stopped          0/1        512M    1G    mytestapp.paas-ta.co.kr
welcome-cf    started          1/1        1G      1G    welcome-cf09.paas-ta.co.kr

D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>cf unbind-service welcome-cf mydbtest
Unbinding app welcome-cf from service mydbtest in org sangyeon / space dev as sang.han@infrancismail.com.
..
OK

D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\cloud\WPaaS\cloudfoundry\wcf_test2\welcome-cf>

```

App과 서비스 인스턴스
언바인딩 명령어

\$ cf unbind-service
 <APP_NAME>
 <SERVICE_INSTANCE>

» CLI로 서비스 생성 및 연동

```

C:\Windows\system32\cmd.exe
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>cf services
Getting services in org sangyeon / space dev as sang.han@infrancismail.com...
OK

name      service  plan          bound apps  last operation
mydbtest  Mysql-DB Mysql-Plan1-10con  create succeeded

D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>cf delete-service mydbtest

Really delete the service mydbtest?> y
Deleting service mydbtest in org sangyeon / space dev as sang.han@infrancismail.com...
OK

D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>
D:\CloudWPaaS\cloudfoundry\wcf_test2\welcome-cf>

```

서비스 인스턴스
삭제 명령어

\$ cf delete-service
SERVICE_INSTANCE [-f]

» CLI를 이용한 앱 관리

```

C:\Users\spring>cf start test01
Starting app test01 in org cloudpack / space devpack04 as devpack04...

0 of 1 instances running, 1 starting
1 of 1 instances running

App started

OK

App test01 was started using this command `CALCULATED_MEMORY=$(($PWD/.java-buildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-2.0.2_RELEASE -memorySizes=metaspace:64m.. -memoryWeights=heap:75,metaspace:10,native:10,stack:5 -memoryInitials=heap:100%,metaspace:100% -totMemory=$MEMORY_LIMIT) && JAVA_HOME=$PWD/.java-buildpack/open_jdk_jre JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh $CALCULATED_MEMORY -Daccess.logging.enabled=false -Dhttp.port=$PORT" exec $PWD/.java-buildpack/tomcat/bin/catalina.sh run`

Showing health and status for app test01 in org cloudpack / space devpack04 as devpack04...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: test01.devpack.co.kr
last uploaded: Sat May 14 05:32:06 UTC 2016
stack: cflinuxfs2
buildpack: java-buildpack=v3.6-https://github.com/cloudfoundry/java-buildpack.git#5194155 maria-db-jdbc=1.4.4 open-jdk-like-jre=1.8.0_91-unlimited-crypto open-jdk-like-memory-calculator=2.0.2_RELEASE tomcat-access-logging-support=2.5.0_RELEASE tomcat-instance=8.0...

state      since                cpu    memory          disk
-----
#0  running  2016-05-14 02:48:36 PM  0.0%   142.8M of 512M  129.1M of 1G

```

CF에 로그인되어있다는
전제하에 이 과정을 시행

시작 명령어

\$ cf start 앱이름

중지 명령어

\$ cf stop 앱이름

재시작 명령어

\$ cf restart 앱이름

App 상태 확인 명령어

\$ cf app 앱이름

\$ cf apps

App 삭제 명령어

\$ cf delete 앱이름

실습

CLI를 활용한 cf명령 실시

접속 api Endpoint

<https://api.paas-ta.co.kr>

CLI 설치 URL

<https://docs.cloudfoundry.org/cf-cli/install-go-cli.html#installer>

실습 소요시간

약 1시간

진행 방법

cf push를 이용하여, 애플리케이션 등록
cf bind-service를 이용하여, 애플리케이션과 서비스 바인딩

필요 도구

노트북, CLI 설치 프로그램

실습

CLI를 활용한 cf 명령 실시

```
applications:  
- name: api-server  
  memory: 700M  
  instances: 1  
  path: api_server.war  
  buildpack: java_buildpack  
  env:  
    spring_application_name: apiserver  
    spring_cloud_config_uri: http://config-server.115.68.46.188.xip.io  
    eureka_client_serviceUrl_defaultZone: http://eureka-server.115.68.46.188.xip.io:8761/eureka/  
    eureka_client_instance_hostname: localhost  
    server_port: 8090
```

설명

제공된 manifest.yml 파일을 확인한다.
api_server.war 파일의 위치가 맞는지 확인한다.

실습

CLI를 활용한 cf 명령 실시

```
C:\Users\박철한\Desktop\교육실습\config>cf push -f manifest.yml
Pushing from manifest to org system / space dev as admin...
Using manifest file manifest.yml
Deprecation warning: Use of 'buildpack' attribute in manifest is deprecated in favor of 'buildpacks'. Please see http://docs.cloudfoundry.org/devguides/deploy-apps/manifest.html#deprecated for alternatives and other app manifest deprecations. This feature will be removed in the future.

Getting app info...
Creating app with these attributes...
+ name: config-Server
+ path: C:\Users\박철한\Desktop\교육실습\config\Config_Server.war
+ buildpacks:
+   java_buildpack
+ instances: 1
+ memory: 768M
+ env:
+ management_security_enabled
+ server_port
+ spring_application_name
+ spring_cloud_config_server_git_password
+ spring_cloud_config_server_git_uri
+ spring_cloud_config_server_git_username
+ routes:
+ config-server.115.68.46.188.xip.io

Creating app config-Server...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
422.58 KiB / 422.58 KiB [=====] 100.00% 1s

Waiting for API to complete processing files...
```

설명

cf push -f manifest.yml 명령어로 앱을 업로드한다.

실습

CLI를 활용한 cf 명령 실시

```
C:\Users\박철한\Desktop\교육실습\config>cf apps
Getting apps in org system / space dev as admin...
OK
```

name	requested state	instances	memory	disk	urls
UI	started	1/1	1G	1G	ui.115.68.46.188.xip.io
api-server	started	1/1	700M	1G	api-server.115.68.46.188.xip.io
config-Server	started	1/1	768M	1G	config-server.115.68.46.188.xip.io
eureka-server	started	1/1	700M	1G	eureka-server.115.68.46.188.xip.io

설명

cf apps 를 입력하여, 정상적으로 배포가 완료되었는지 확인한다.

실습

CLI를 활용한 cf 명령 실시

```
C:\Users\박철한\Desktop\교육실습\config>cf services  
Getting services in org system / space dev as admin...
```

name	service	plan	bound apps	last operation	broker
mysql	Mysql-DB	Mysql-Plan1-10con		create succeeded	mysql
pipeline	delivery-pipeline	delivery-pipeline-shared		create succeeded	delivery-pipeline
source-controller	p-paasta-sourcecontrol	Default		create succeeded	source-controller

설명

cf service를 입력하여, 바인딩할 서비스에 대해서 정보를 취득한다.

실습

CLI를 활용한 cf명령 실시

```
C:\Users\박철한>cf bind-service api-server mysql
Binding service mysql to app api-server in org system / space dev as admin...
OK
```

설명

cf bind-service [바인드할 앱명] [바인드할 서비스명]
를 입력하여, 배포한 앱과 서비스를 바인딩한다.

실습

CLI를 활용한 cf 명령 실시

```
C:\Users\박철한\Desktop\교육실습\api-server>cf services
Getting services in org system / space dev as admin...
```

name	service	plan	bound apps	last operation	broker
mysql	Mysql-DB	Mysql-Plan1-10con	api-server	create succeeded	mysql
pipeline	delivery-pipeline	delivery-pipeline-shared		create succeeded	delivery-pipeline
source-controller	p-paasta-sourcecontrol	Default		create succeeded	source-controller

설명

cf service를 입력하여, 정상적으로 바인딩이 되었는지 확인한다.

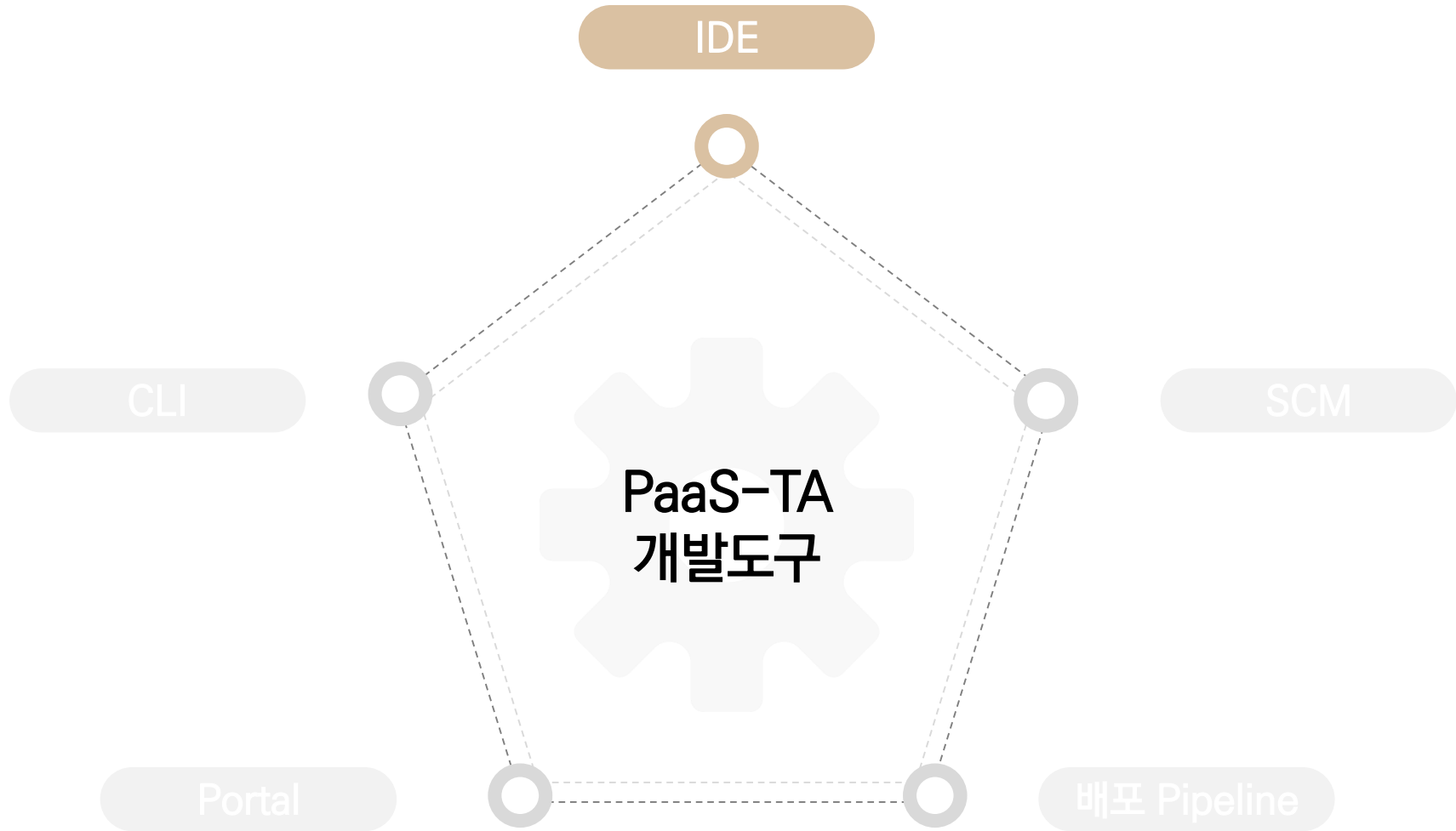
／ 핵심정리 ／

- ✓ CLI를 통해 **api endpoint에 접근하여 PaaS-TA를 사용할 수 있음**
- ✓ **CLI기반으로 앱을 배포할 수 있음**
- ✓ **CLI기반으로 서비스를 생성 및 앱과 연동할 수 있음**



MEMO





» 개요

IDE(통합 개발 환경)

프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서
처리하는 환경을 제공하는 소프트웨어

코딩, 디버그, 컴파일, 배포 등의 관련 모든 작업을 포함

종래의 소프트웨어 개발에서 따로 사용했던 컴파일러, 텍스트 편집기, 디버거 등을
하나로 묶어 **대화형 인터페이스를 제공**

최근의 IDE는 그래픽 사용자 인터페이스(GUI) 애플리케이션 개발용 고속 개발
도구가 많음



» 개요

클라우드 IDE

기존 IDE의 불편함을 줄이기 위해 등장

기존 설치형 IDE와는 다른 **새로운 기능을 제공**하며, 기존 IDE를 대체하는 목적보다 **새로운 시장을 형성**하기 위해 등장함

» 개요

클라우드 IDE

장점

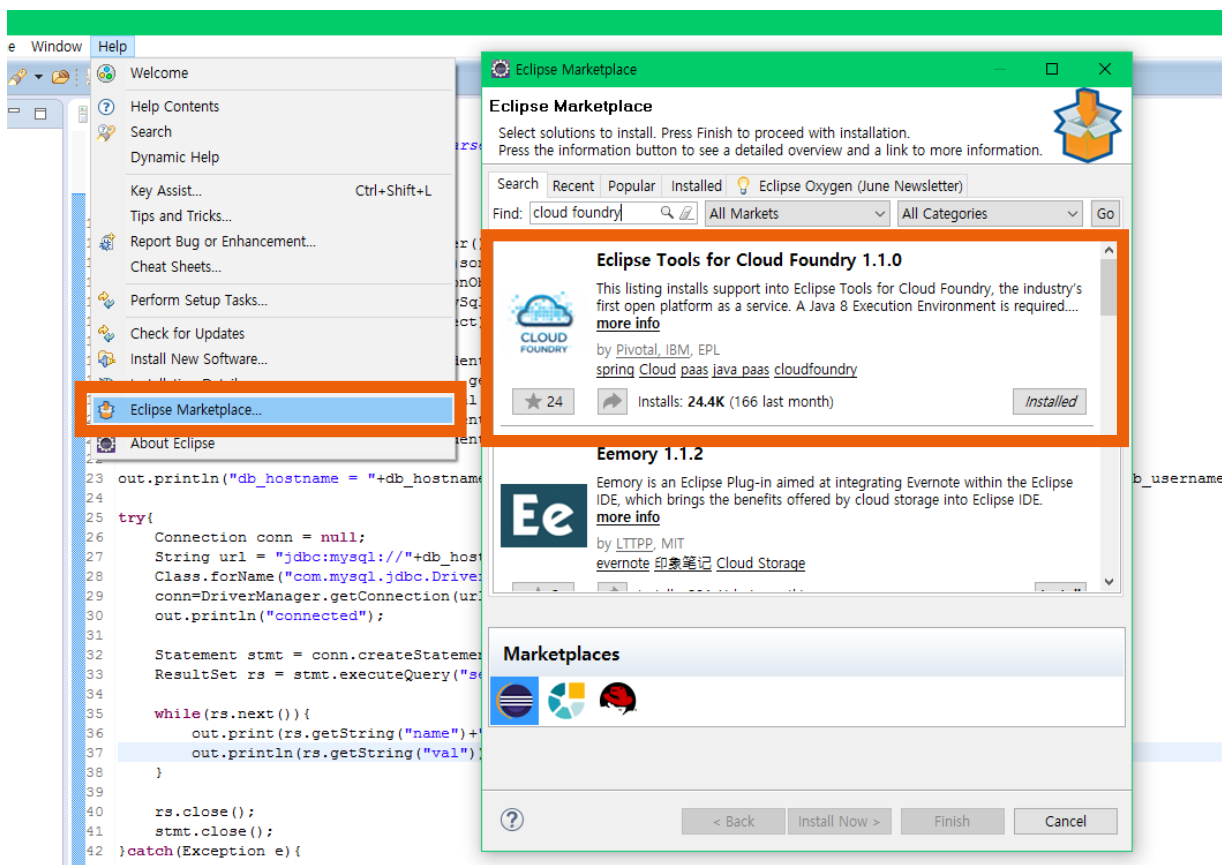
- ✓ 물리적인 컴퓨터에 종속되지 않음
: 웹 브라우저를 실행하고 로그인만 하면 사용 가능하므로 인터넷에 연결된 컴퓨터만 있으면 어디서나 개발 작업 가능
- ✓ 여러 사람이 동시에 작성 가능해 '페어 프로그래밍'을 하기 편함
: 댓글과 공유 기능을 제공해 다양한 피드백을 쉽게 받을 수 있음
- ✓ 접근 권한을 중앙에서 관리해 보안성을 높일 수 있음
- ✓ 수정 기록이 웹에 실시간으로 남기 때문에 향후 코드 내용 추적 용이

단점

- ✓ 기존 설치형 IDE보다 성능이 낮고 기능이 부족

» Eclipse plugin 설치

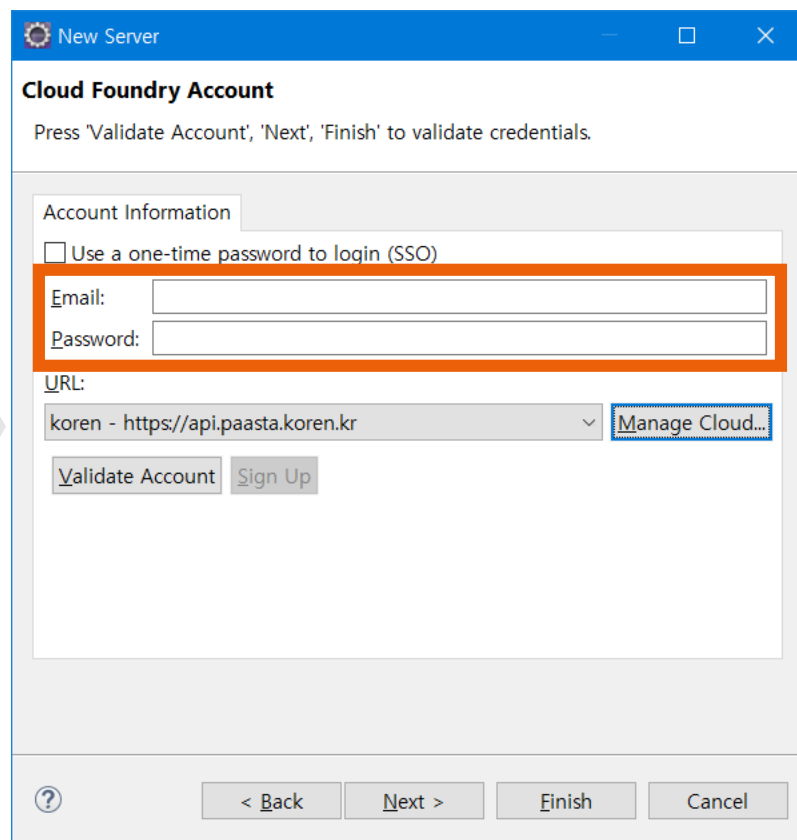
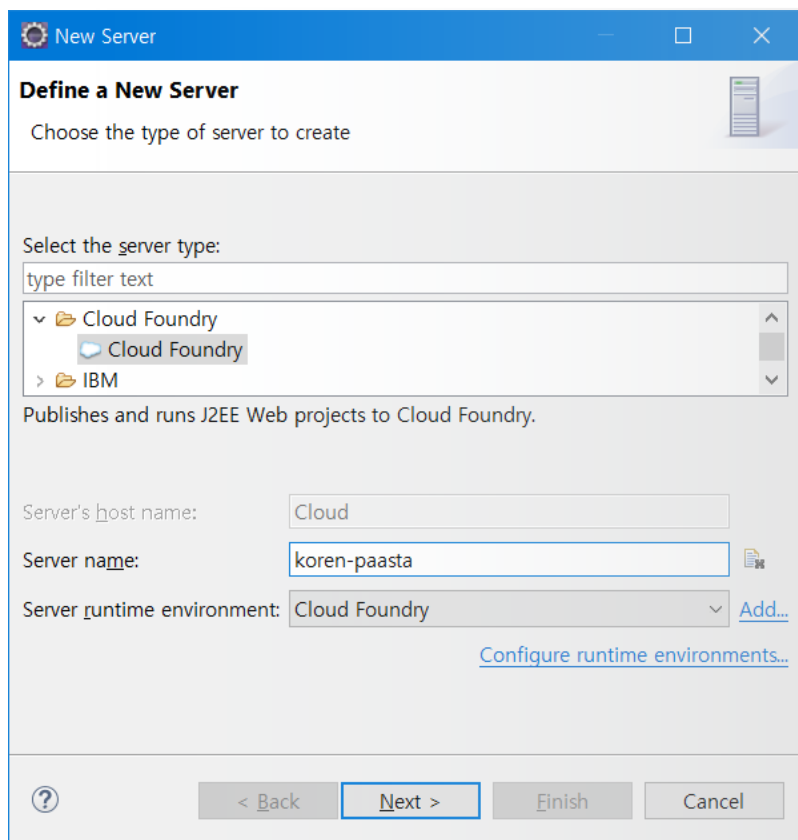
Application Instance 관리



* 플러그인은 Eclipse v3.8 및 v4.3 (Java EE 버전 권장) 및 STS 3.0.0 이상을 지원

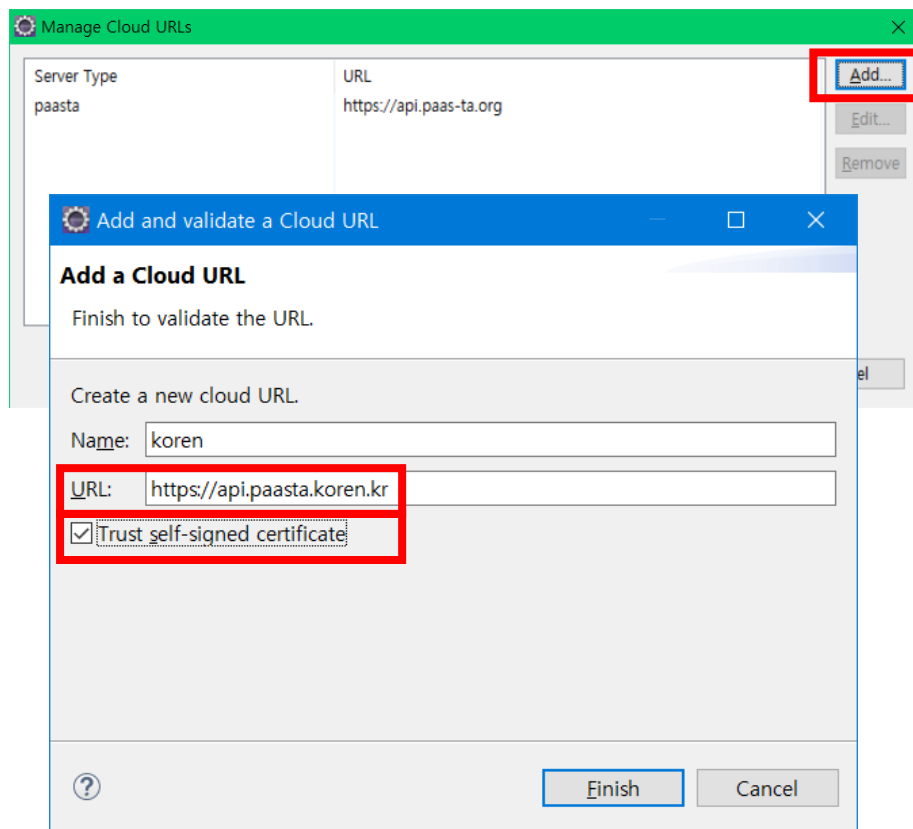
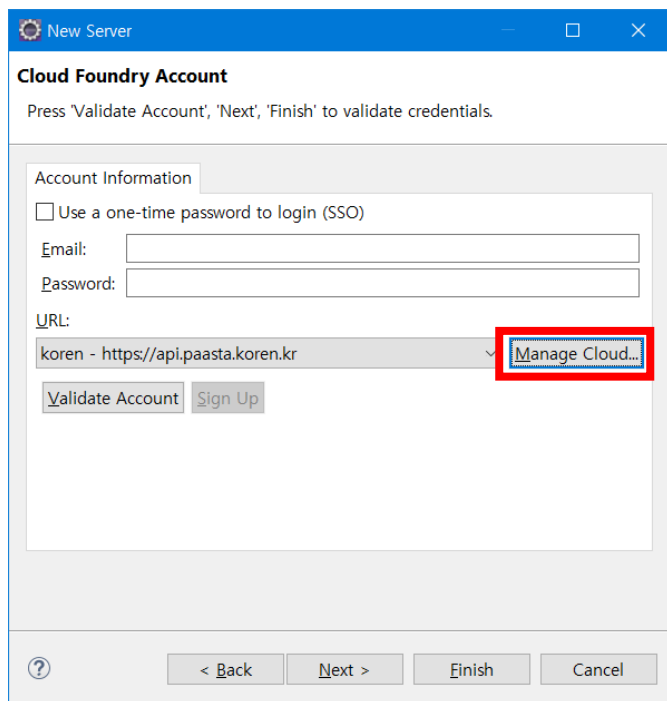
» Eclipse plugin으로 PaaS-TA연동하기

Cloud Foundry 선택



» Eclipse plugin으로 PaaS-TA연동하기

Cloud Foundry 선택



» 프로젝트 배포하기

New Server

Cloud Foundry Account

Press 'Validate Account', 'Next', 'Finish' to validate credentials.

Account Information

Use a one-time password to login (SSO)

Email: lect01

Password: ●●●●●●

URL: koren - https://api.paasta.koren.kr [Manage Cloud...](#)

Validate Account [Sign Up](#)

? < Back Next > Finish Cancel

validate account



계정 인증



Finish

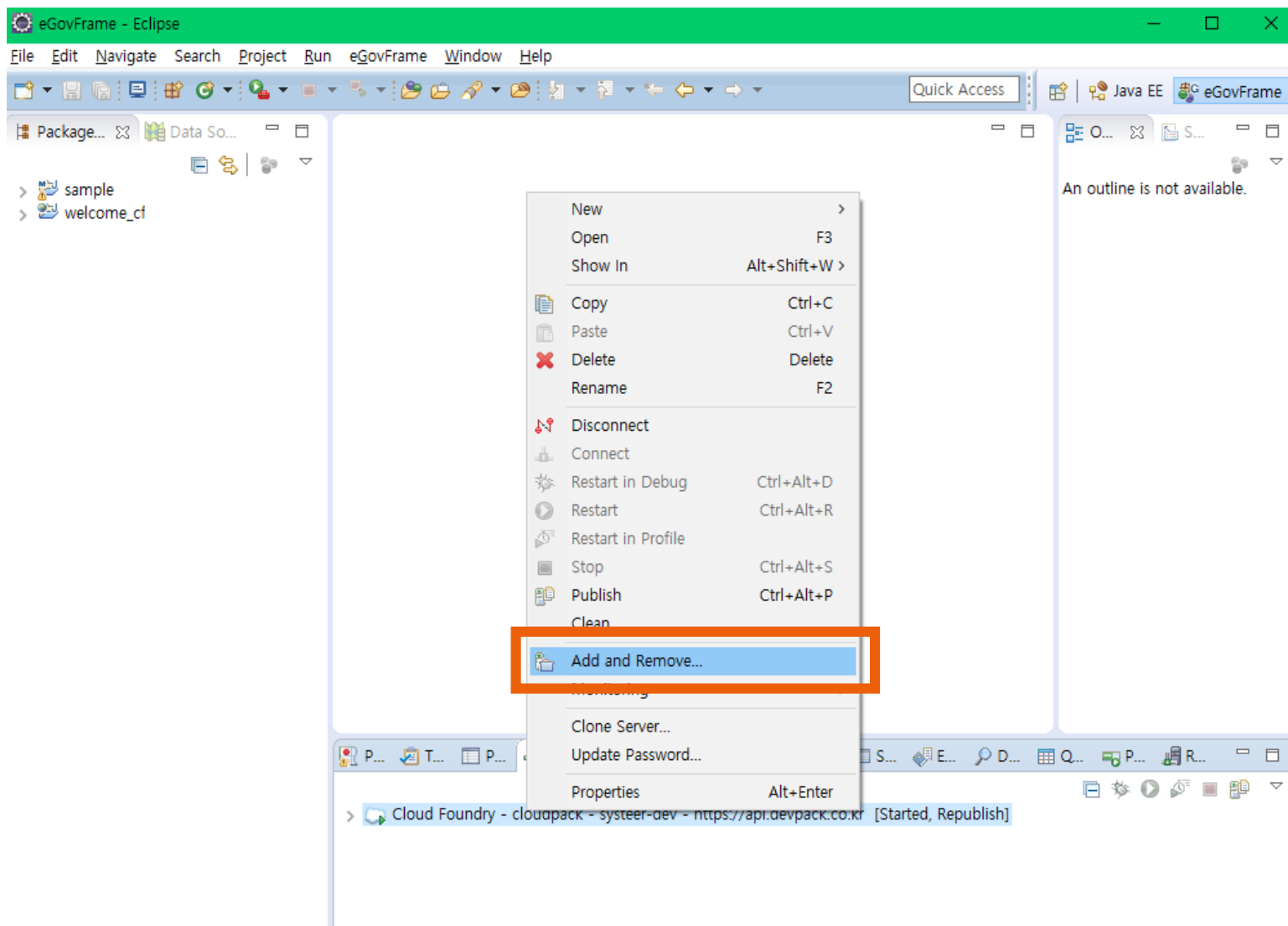


server 생성 완료

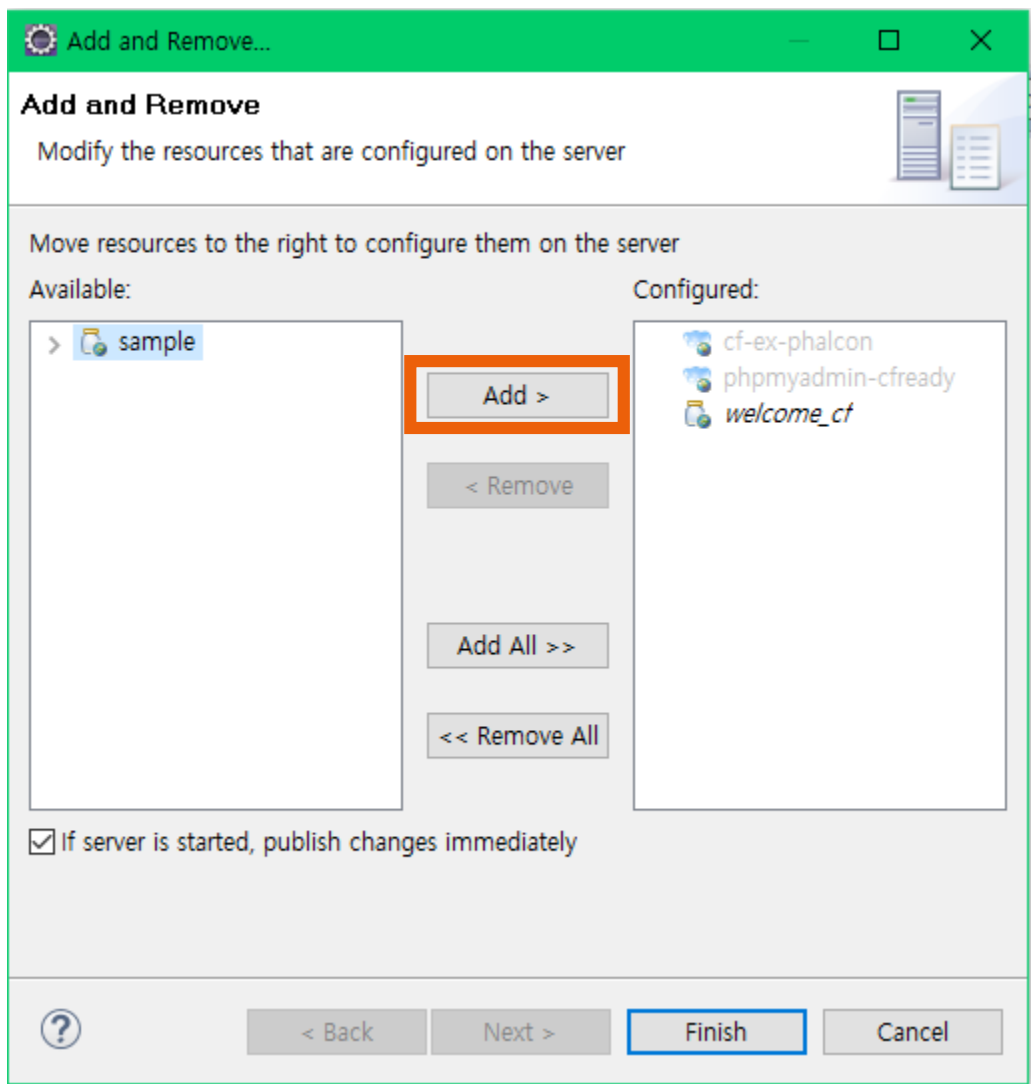


개발 프로젝트 배포

» 서비스 생성 및 연동하기



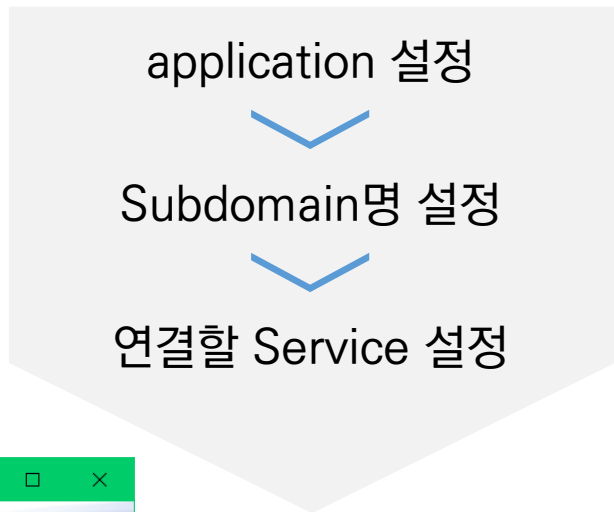
» 서비스 생성 및 연동하기



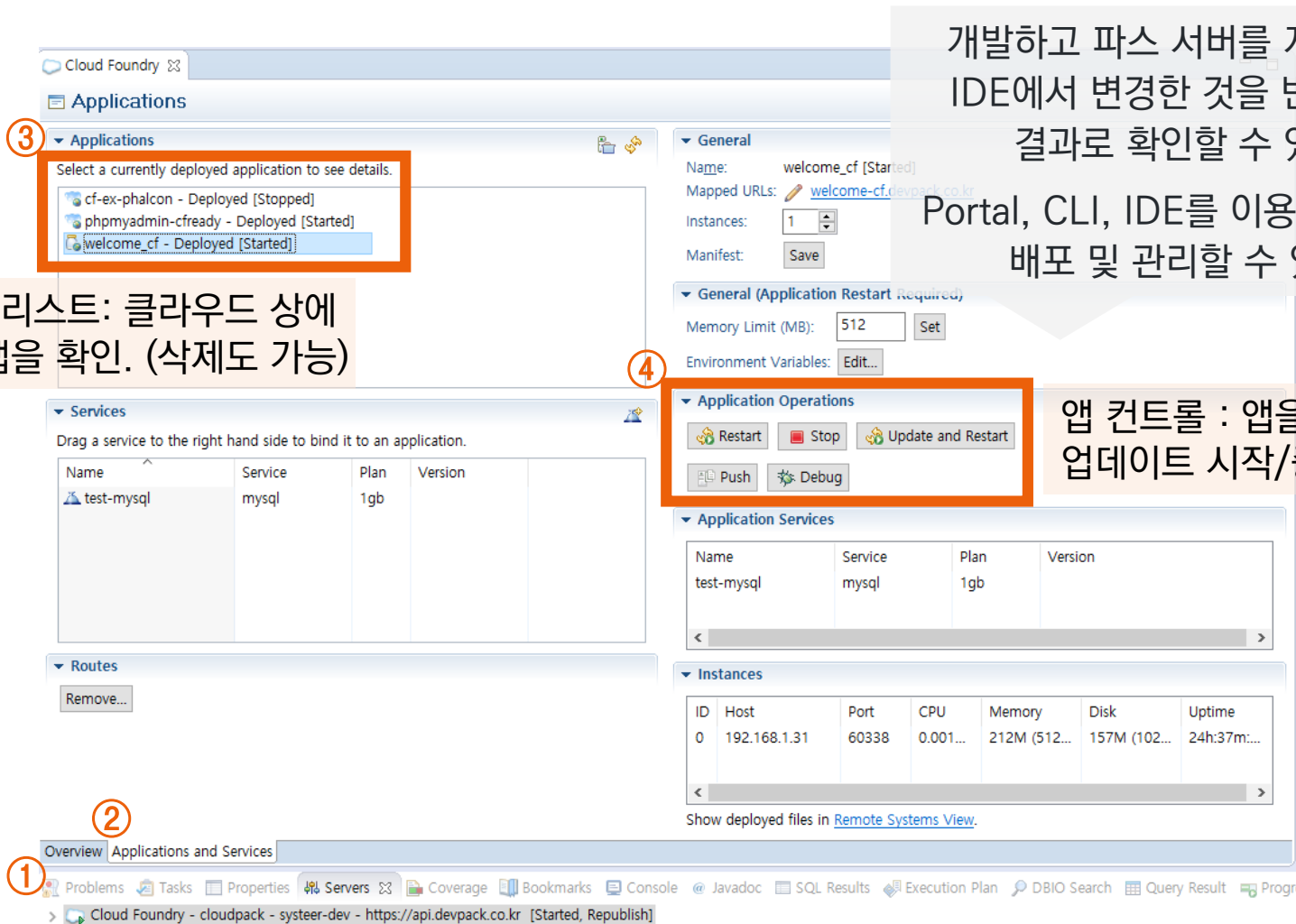
» 서비스 생성 및 연동하기

The screenshots illustrate the configuration steps for a PaaS application:

- Application details:** The application name is set to `mariner`.
- Launch deployment:** The subdomain is set to `mariner`, the domain is `paas-ta.org`, and the memory limit is set to `512` MB.
- Services selection:** The `mariner` service is selected for binding to the application.



» 대시보드를 이용한 앱 관리



개발하고 파스 서버를 지정하고 IDE에서 변경한 것을 반영해서 결과로 확인할 수 있음

Portal, CLI, IDE를 이용해서 앱을 배포 및 관리할 수 있음

애플리케이션 리스트: 클라우드 상에 올라가 있는 앱을 확인. (삭제도 가능)

앱 컨트롤 : 앱을 배포 업데이트 시작/중지/재시작

① Overview Applications and Services
 Problems Tasks Properties Servers Coverage Bookmarks Console Javadoc SQL Results Execution Plan DBIO Search Query Result Progr
 > Cloud Foundry - cloudpack - systeer-dev - https://api.devpack.co.kr [Started, Republish]

*기타 연결 서비스 상태 메모리 등을 확인 관리 가능

» IntelliJ를 이용한 PaaS-TA연동



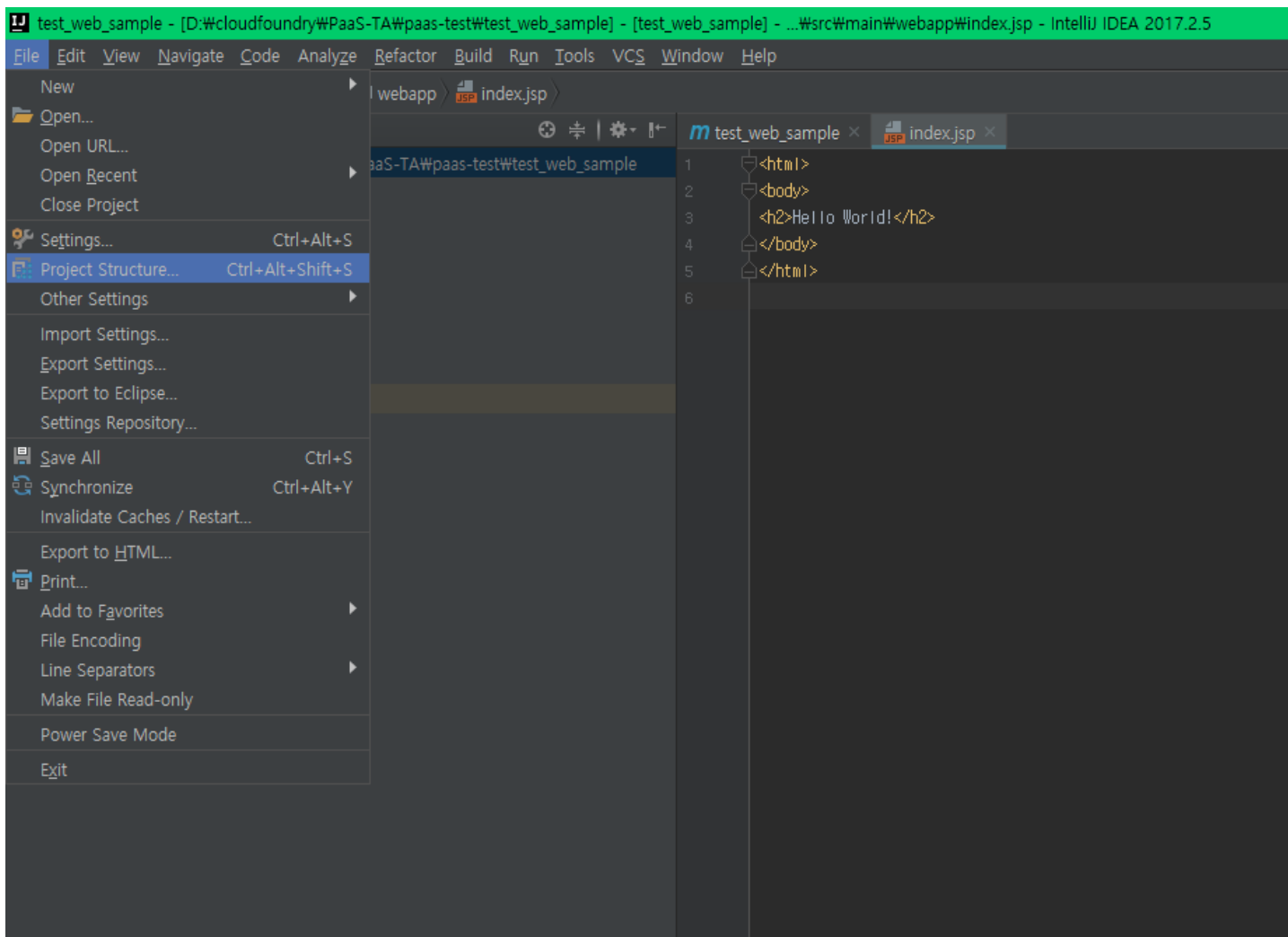
정의

JetBrains사에서 제작한
상용 자바 통합 개발 환경

특징

- 이클립스 재단 의 이클립스와 썬 마이크로시스템즈의 넷빈즈로 대표되는 무료 자바 통합개발환경에서 볼랜드(/코드기어)의 제이빌더(JBuilder)와 함께 얼마 안 되는 상용 개발 도구 가운데 하나
- 유료프로그램

» IntelliJ를 이용한 PaaS-TA연동



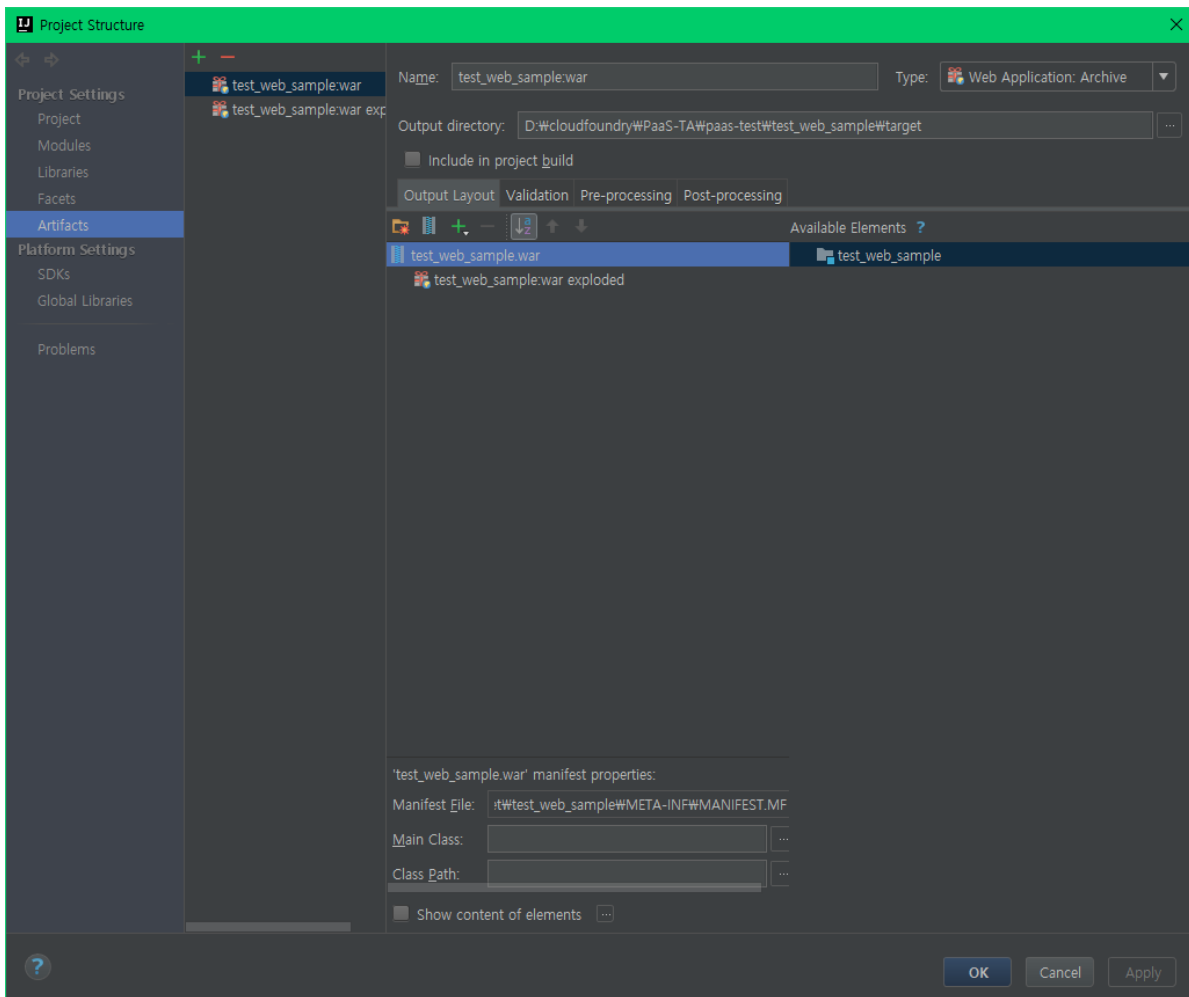
Project Structure
Artifact 설정



Cloud Foundry 사용

» IntelliJ를 이용한 PaaS-TA연동

Project Structure 에서 Artifact를 war로 설정한 예제



» IntelliJ를 이용한 PaaS-TA연동

Project Structure 에서 Artifact를 war로 설정한 예제

프로젝트 설정 시 artifact를 설정하지 않으면 배포하는 PaaS-TA 형식이 보이지 않음

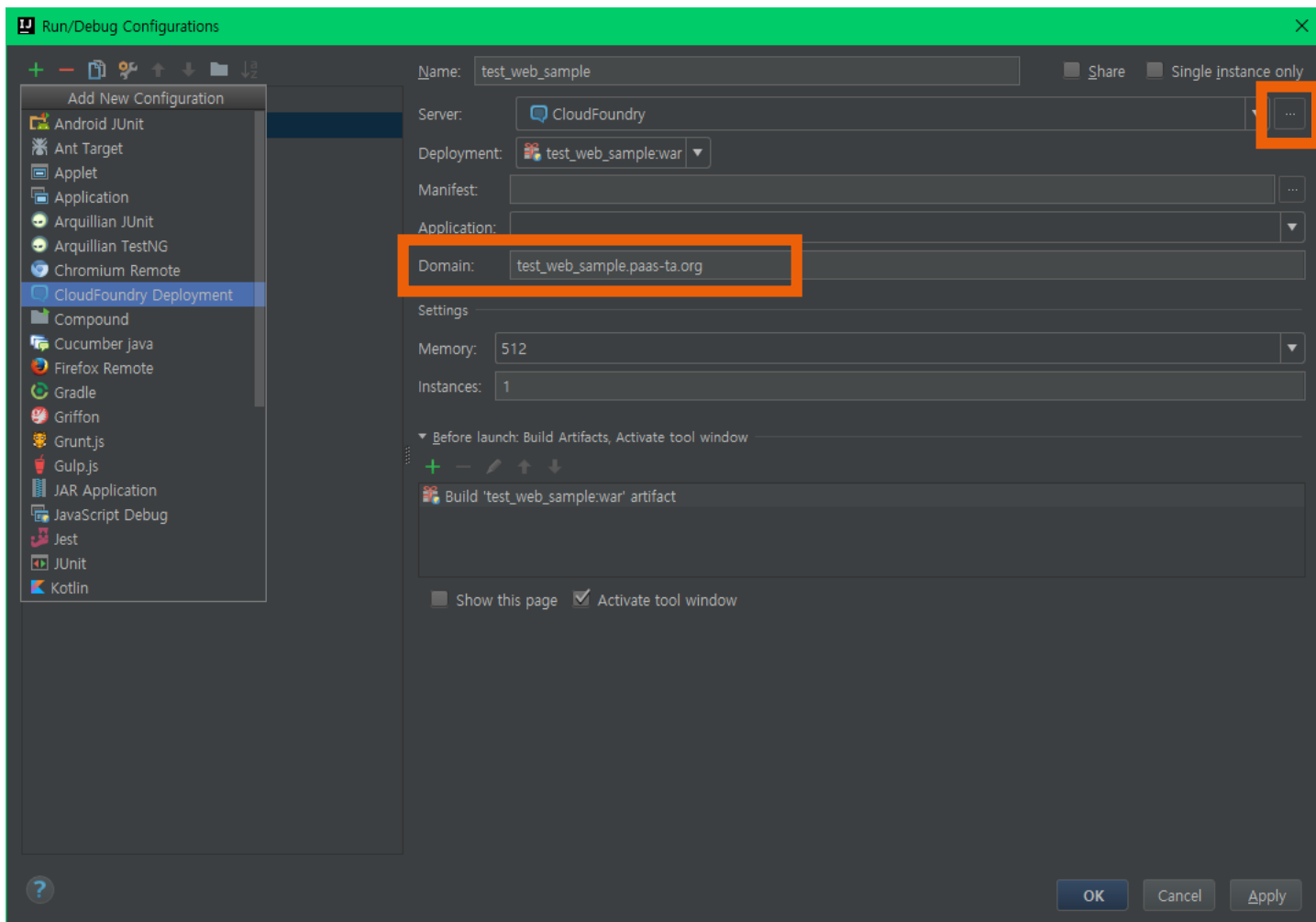
프로젝트 스트럭처에서 artifact를 어떻게 할지 세팅 PaaS-TA 설정을 할 수 있음

이 부분만 다르고 **이클립스와 동일하게 개발과 동시에 배포**할 수 있음
(이클립스보다 간단하게 구성)

앱을 정지하는 등의 기능은 가능하나 세밀한 기능은 구현하기 어려움

서비스를 생성하는 인터페이스를 제공하고 있지 않음
(앱 배포한 것을 컨트롤할 수 있는 수준)

» IntelliJ를 이용한 PaaS-TA연동

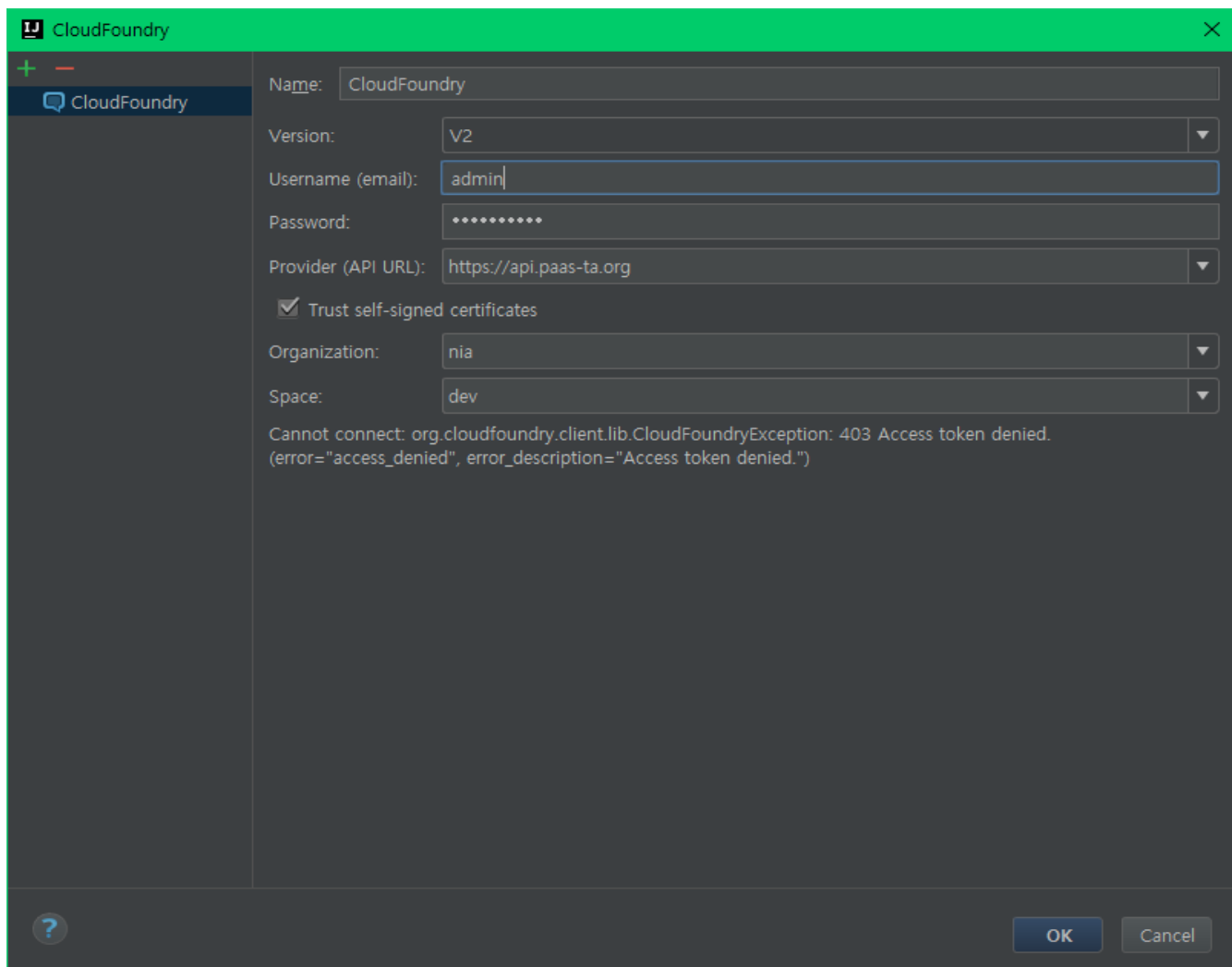


Cloudfoundry 선택

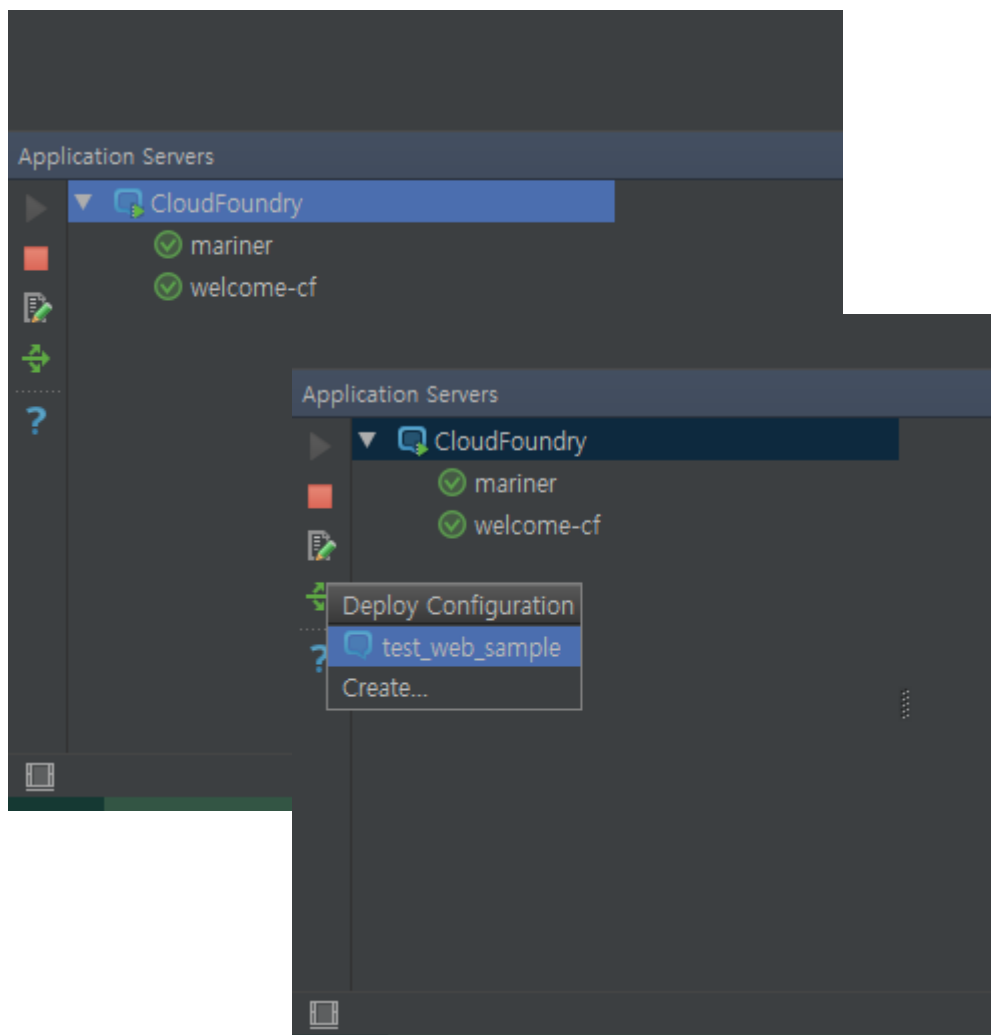
서버 선택

Full Domain 입력

» IntelliJ를 이용한 PaaS-TA연동



» IntelliJ를 이용한 PaaS-TA연동



Application Server 리스트에서
프로젝트 배포 및 배포된
Application 정보 확인 가능
(배포 및 삭제만 됨)

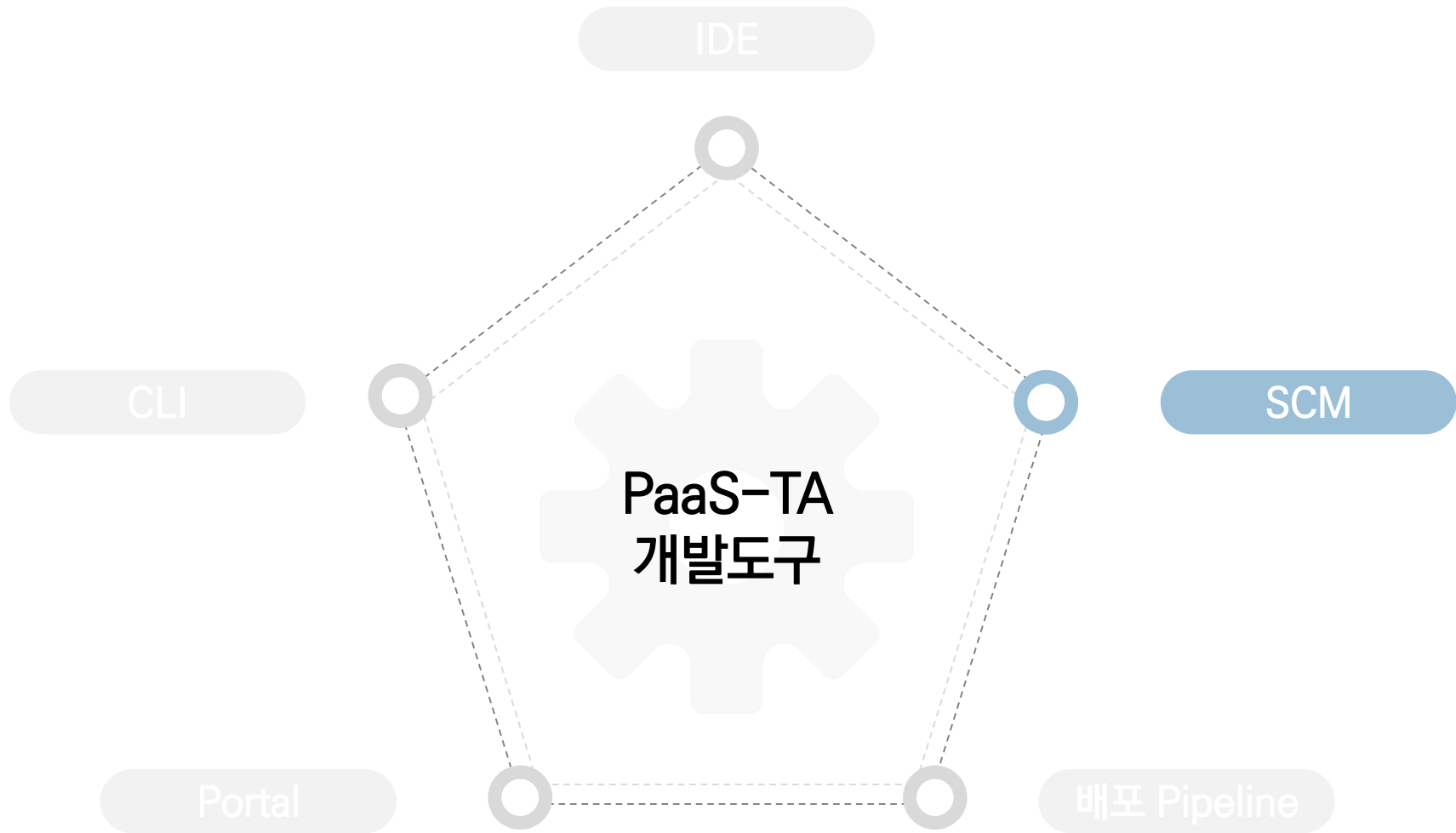
／ 핵심정리 ／

- ✓ Eclipse에서 **cloudfoundry plugin**을 통하여 **api**에 접속할 수 있음
- ✓ **프로젝트를 배포**할 수 있음
- ✓ 접속한 **api** 데시보드에서 **app**을 **관리**할 수 있음
- ✓ **서비스를 생성**하여 **앱과 연동**할 수 있음



MEMO





» 개요



SCM

시스템 형상 요소의 기능적 특성이나 물리적 특성을 문서화하고 그러한 특성의 변경을 관리하며, 변경의 과정이나 실현 상황을 기록 및 보고하여 지정된 요건이 충족되었다는 사실을 검증하는 것, 또는 그 과정

형상 관리는 프로젝트를 개발하는 동안 생산성과 안전성을 높여 **좋은 품질의 소프트웨어를 생산**하고 **유지보수도 용이**하게 하는데 목적이 있음

» 개요

SCM 기능

언제라도 특정 시간대에 가장 안정적인 버전의 소프트웨어를 유지할 수 있도록 소프트웨어 제품이 변경되어가는 상태에 대한 가시성을 확보

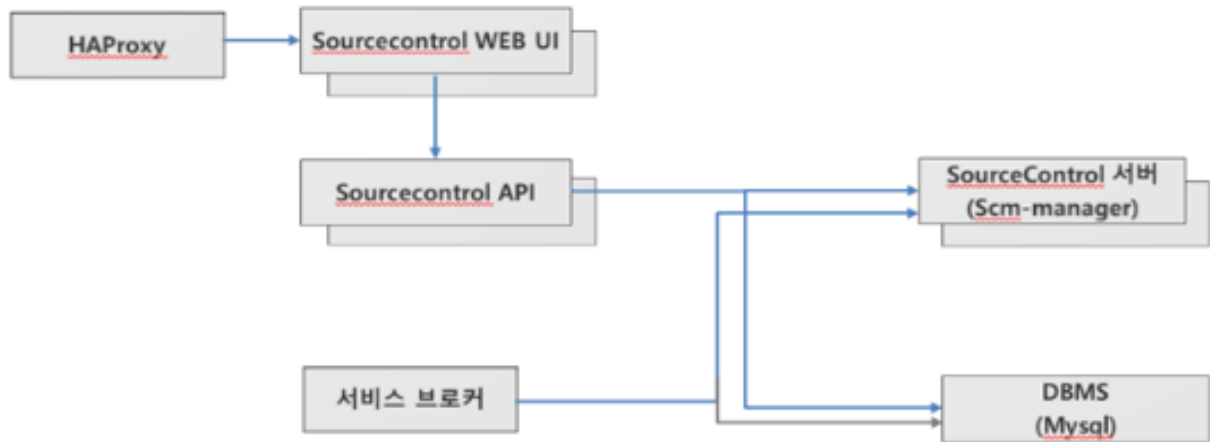
누가 변경했는지, 변경된 것은 무엇인지, 언제 변경되었는지, 왜 변경했는지 알림

적절한 변경 관리를 통하여 **무제한 변경을 사전에 예방**하고 **변경에 따른 부작용을 최소화**함

프로젝트를 적절히 통제하여 체계적이고 효율적으로 관리할 수 있으며 가시성과 추적성을 보장함으로써 소프트웨어의 생산성과 품질을 높일 수 있음

» SCM서비스 소개

시스템 구성도



VM 명	인스턴스 수	vCPU 수	메모리(GB)	디스크(GB)
scm-manager	1	1	4	Root 8G + 영구디스크 30G
paasta-sourcecontrol-webUI	N	1	2	Root 8G + 영구디스크 2G
paasta-sourcecontrol-broker	1	1	1	Root 8G + 영구디스크 2G
paasta-sourcecontrol-api	N	1	2	Root 8G + 영구디스크 2G
DBMS(mariadb)	1	2	2	Root 8G + 영구디스크 2G
HAProxy	1	1	1	Root 8G + 영구디스크 2G

실 습

SCM 구성 실습

실습 사이트

playpark.paas-ta.org

실습 소요시간

약 1시간

진행 방법

SCM 서비스를 생성 후 대시보드에서 repository를 생성하여 로컬환경과 동기화해본다.

필요 도구

노트북, 각자의 계정

실습

SCM 구성 실습

The screenshot shows the PaaS-TA Portal dashboard. At the top, there is a navigation bar with the PENNE logo and menu items: 대시보드, 카탈로그, 문서. Below this is a breadcrumb trail: Dashboard > 대한민국 > system > dev. A storage usage bar shows 0 MB / 할당량 100 GB. There are two tabs: 애플리케이션 and 서비스. A 'User Provided 생성' button is visible. Three service cards are shown:

- mysql**
 - 연결 앱 :
 - 서비스 명 : Mysql-DB
 - 제공서비스 : Mysql-Plan1-10con
- pipeline**
 - 연결 앱 :
 - 서비스 명 : delivery-pipeline
 - 제공서비스 : delivery-pipeline-shared
 - 대시보드 : **대시보드**
- source-controller**
 - 연결 앱 :
 - 서비스 명 : p-paasta-sourcecontrol
 - 제공서비스 : Default
 - 대시보드 : **대시보드**

A plus sign (+) is visible in the bottom right corner of the service cards area.

설명

PaaS-TA Portal에서 서비스 탭으로 이동한다.
형상관리의 대시보드를 클릭한다.

실습

SCM 구성 실습

The screenshot shows the SCM system interface. At the top, there is a navigation bar with the logo, '형상관리' (Shape Management), '레파지토리 목록' (Repository List), and 'admin'. Below the navigation bar, the breadcrumb is '레파지토리 목록'. A search bar is present with the text '레파지토리 명' and a search icon. To the right of the search bar are three filter dropdowns: '형상관리 전체', '전체 레파지토리', and '최신 업데이트 순'. A '신규생성' (New Generation) button is highlighted with a red box. Below the search and filter area, the text '조회된 데이터가 없습니다.' (No data found) is displayed. At the bottom of the page, there is a copyright notice: 'Copyright © 2017 PaaS-TA. All rights reserved.'

설명

형상관리에서 신규생성을 클릭한다.

실습

SCM 구성 실습

형상관리 0 레파지토리 목록 admin

레파지토리 신규생성

레파지토리 명 (+필수) UI *영문만 허용

유형 (+필수) Git

레파지토리 설명 (선택) Paas-TA UI

생성 취소

Copyright © 2017 PaaS-TA. All rights reserved.

설명

필수 정보를 입력한 후 생성을 클릭한다.

실습

SCM 구성 실습

형상관리 1 레파지토리 목록 admin

레파지토리 상세보기 (UI) 정보보기/수정 신규생성

파일(file) 커밋(Commit) 0 참여자(Contributor) 1

브랜치: Tag: 레파지토리 클론

폴더/파일명 최종 변경사항 http://admin@115.68.47.175:8084/ 복사

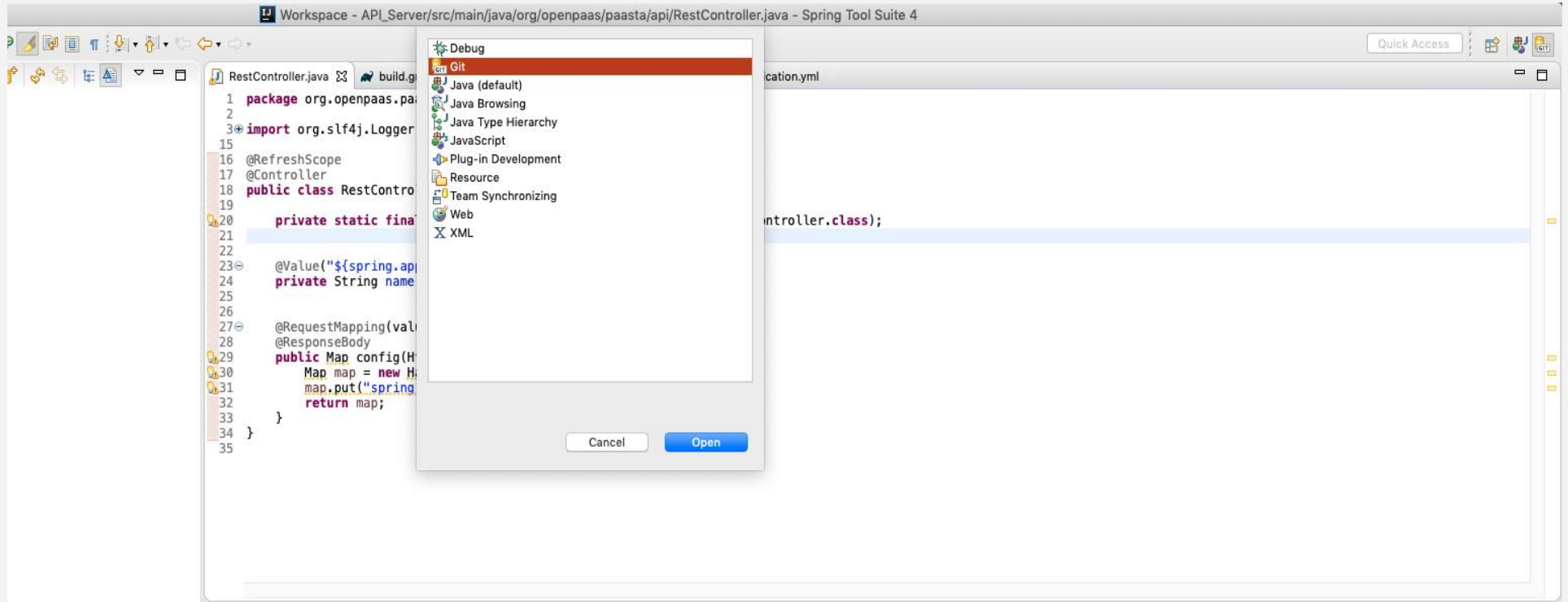
조회된 데이터가 없습니다.

Copyright © 2017 PaaS-TA. All rights reserved.

설명

레파지토리 생성을 확인한다.

실습

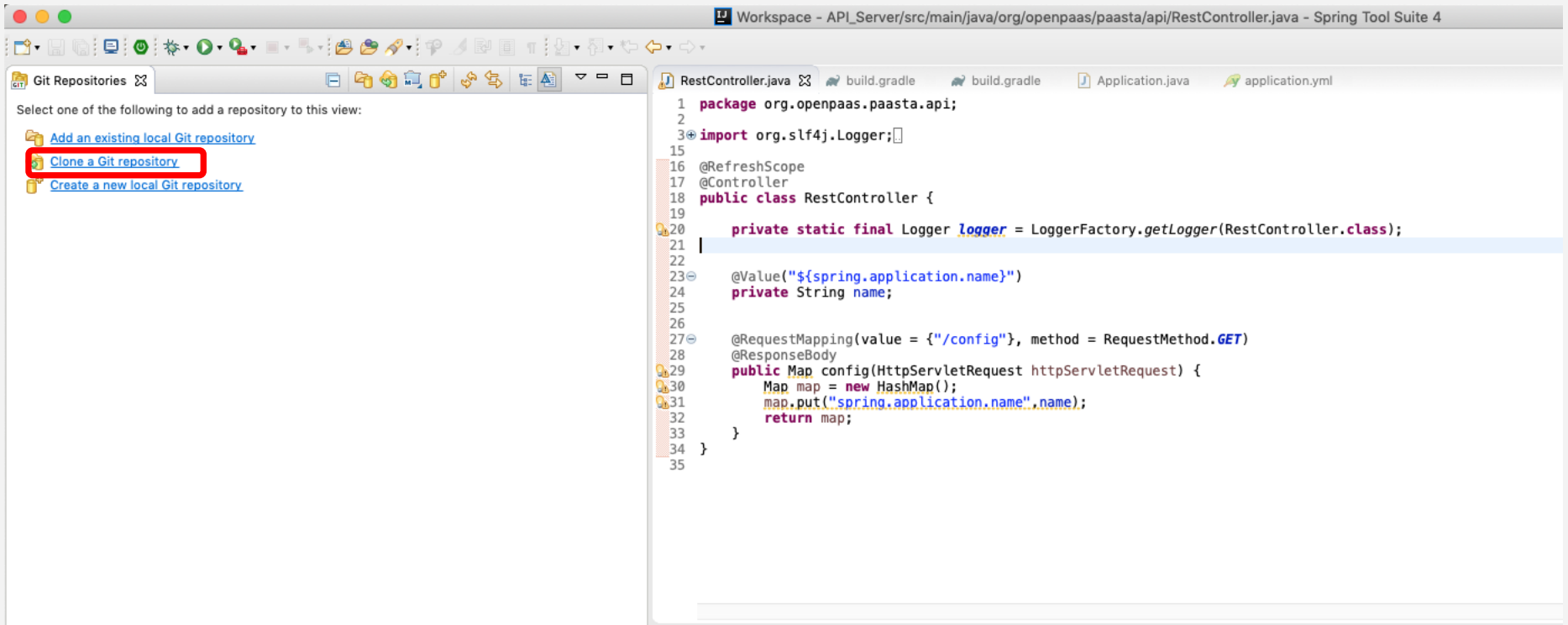


설명

개발툴(STS기준)에서 GIT을 선택한다.

실습

SCM 구성 실습



The screenshot shows the Spring Tool Suite 4 interface. On the left, the 'Git Repositories' view is open, displaying three options: 'Add an existing local Git repository', 'Clone a Git repository' (highlighted with a red box), and 'Create a new local Git repository'. On the right, the 'RestController.java' file is open in the code editor, showing the following code:

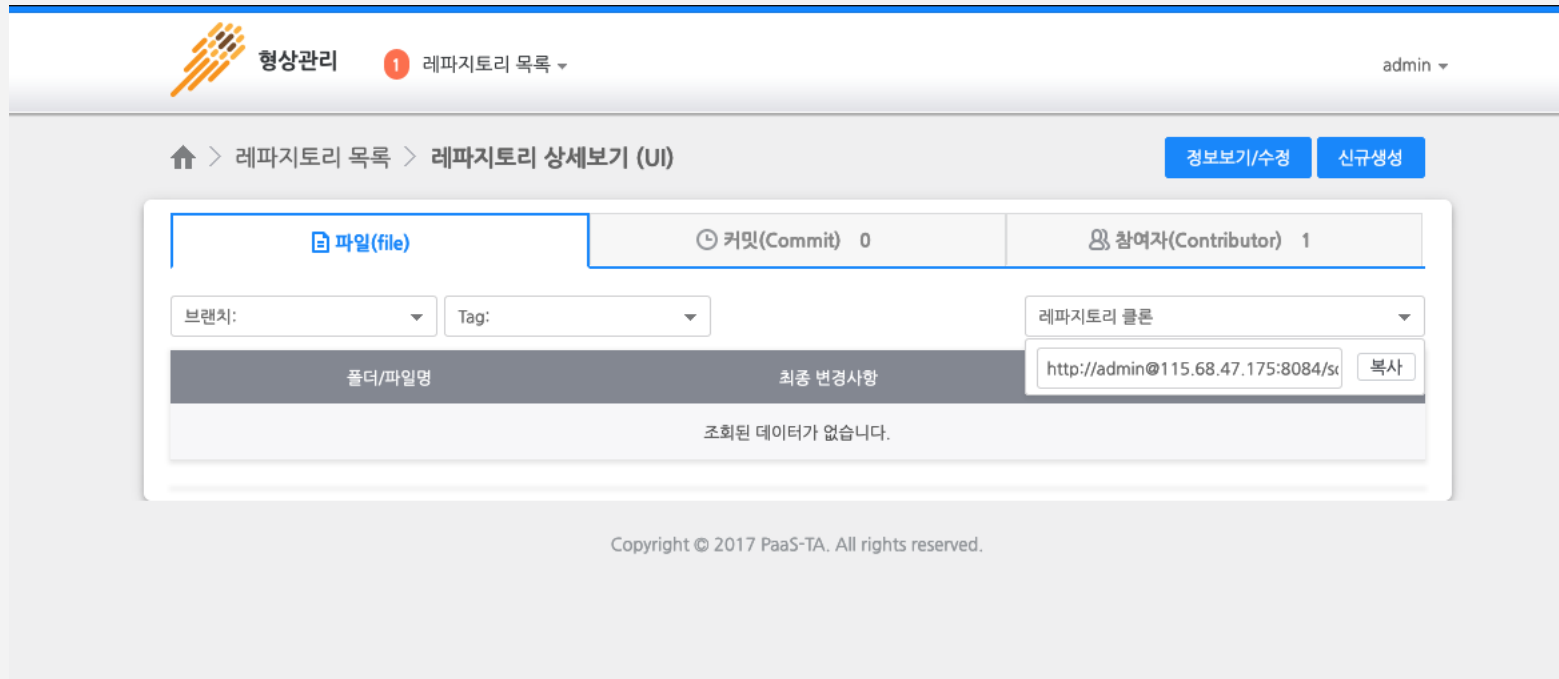
```
1 package org.openpaas.paasta.api;
2
3 import org.slf4j.Logger;
4
5
6
7
8
9
10
11
12
13
14
15
16 @RefreshScope
17 @Controller
18 public class RestController {
19
20     private static final Logger logger = LoggerFactory.getLogger(RestController.class);
21
22
23     @Value("${spring.application.name}")
24     private String name;
25
26
27     @RequestMapping(value = {"/config"}, method = RequestMethod.GET)
28     @ResponseBody
29     public Map config(HttpServletRequest httpServletRequest) {
30         Map map = new HashMap();
31         map.put("spring.application.name", name);
32         return map;
33     }
34 }
35
```

설명

Clone a Git repository를 선택한다.

실습

SCM 구성 실습



형상관리 1 레파지토리 목록 admin

레파지토리 상세보기 (UI) 정보보기/수정 신규생성

파일(file) 커밋(Commit) 0 참여자(Contributor) 1

브랜치: Tag: 레파지토리 클론

폴더/파일명 최종 변경사항 http://admin@115.68.47.175:8084/sr 복사

조회된 데이터가 없습니다.

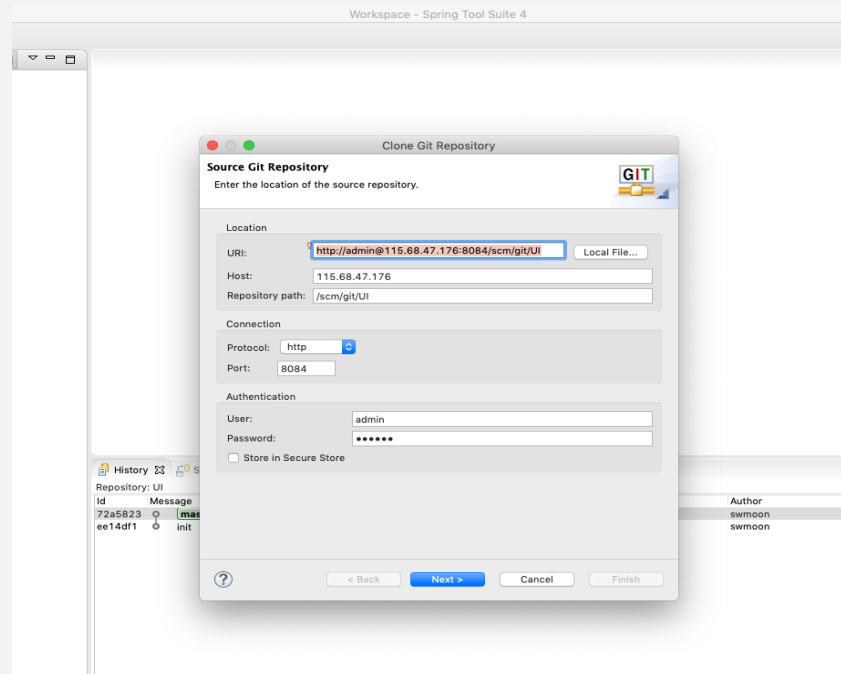
Copyright © 2017 PaaS-TA. All rights reserved.

설명

형상 관리로 이동하여, 레파지토리 클론을 선택 후 URL를 복사한다.

실습

SCM 구성 실습

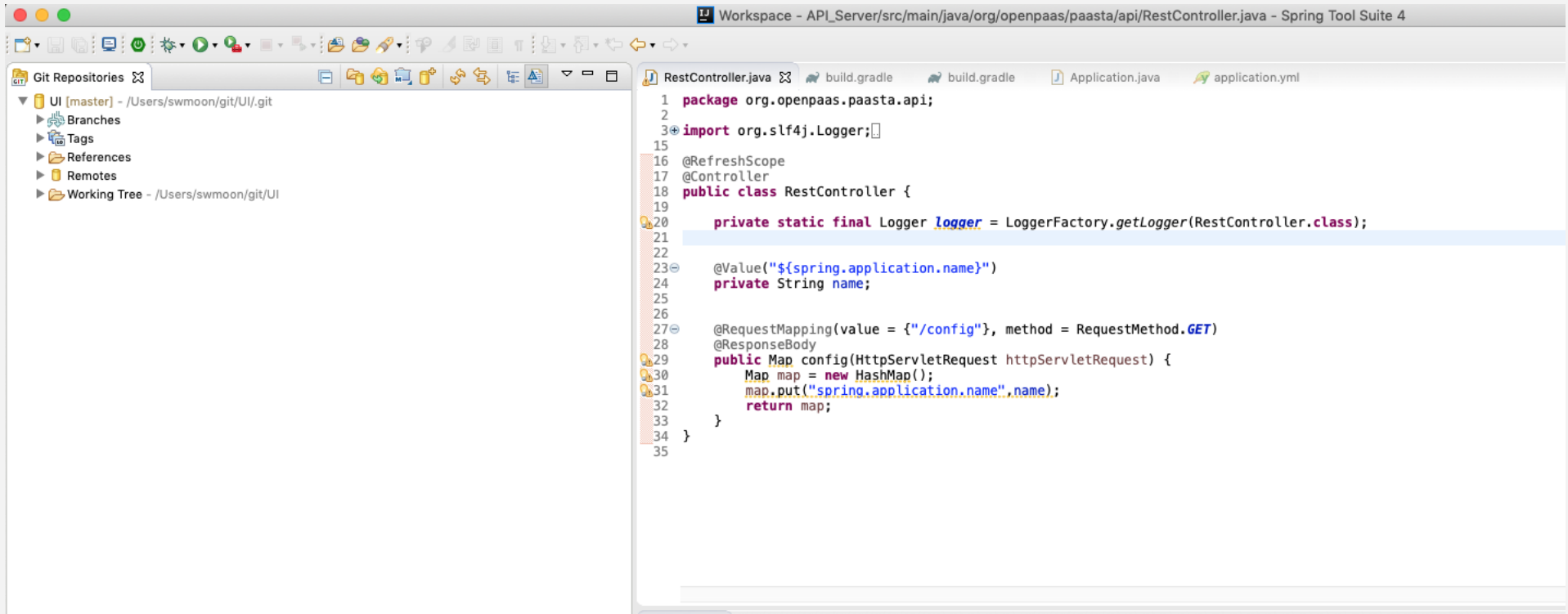


설명

URI에 복사한 주소를 입력한다.

실습

SCM 구성 실습



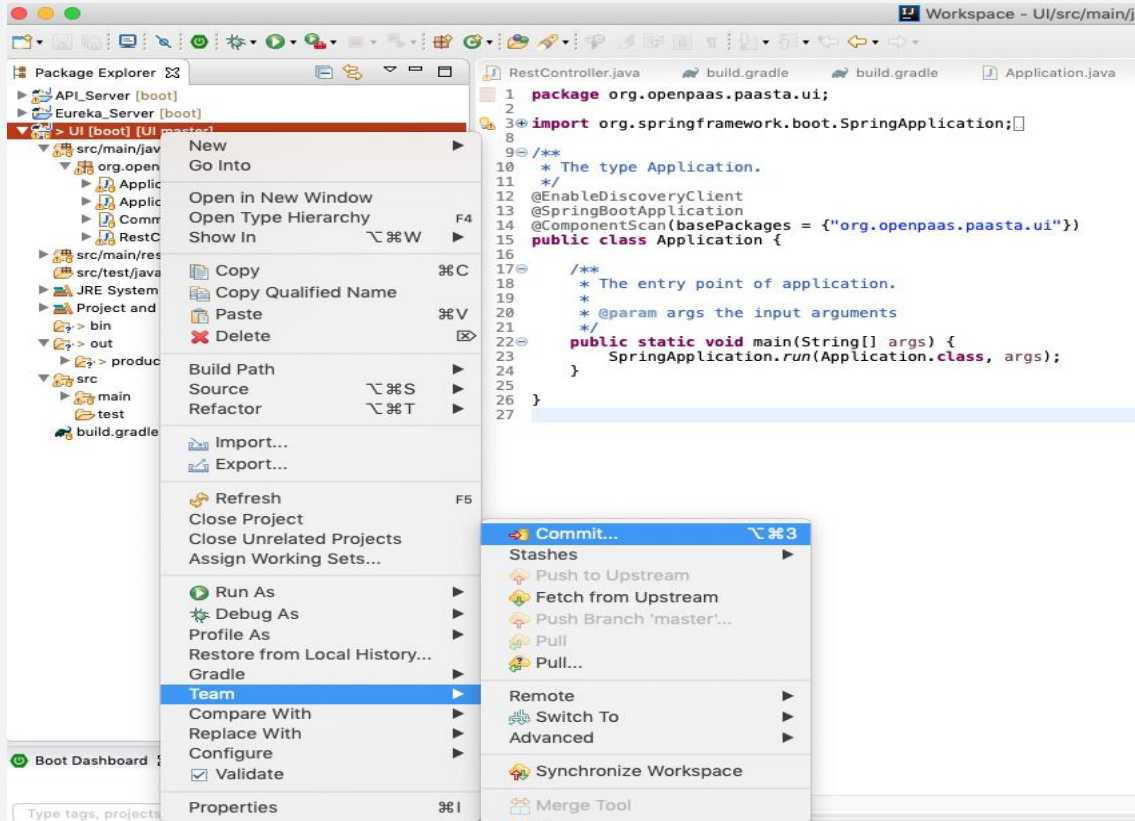
The screenshot shows the Spring Tool Suite 4 IDE. The top toolbar contains various icons for development and testing. The left sidebar shows the 'Git Repositories' view with a tree structure for a repository named 'UI [master]'. The main editor area displays the code for 'RestController.java'.

```
1 package org.openpaas.paasta.api;
2
3 import org.slf4j.Logger;
4
5
6
7
8
9
10
11
12
13
14
15
16 @RefreshScope
17 @Controller
18 public class RestController {
19
20     private static final Logger logger = LoggerFactory.getLogger(RestController.class);
21
22
23     @Value("${spring.application.name}")
24     private String name;
25
26
27     @RequestMapping(value = {"/config"}, method = RequestMethod.GET)
28     @ResponseBody
29     public Map config(HttpServletRequest httpServletRequest) {
30         Map map = new HashMap();
31         map.put("spring.application.name", name);
32         return map;
33     }
34 }
35
```

설명

형상관리에 있는 레파지토리가 추가된것을 확인한다.

실습

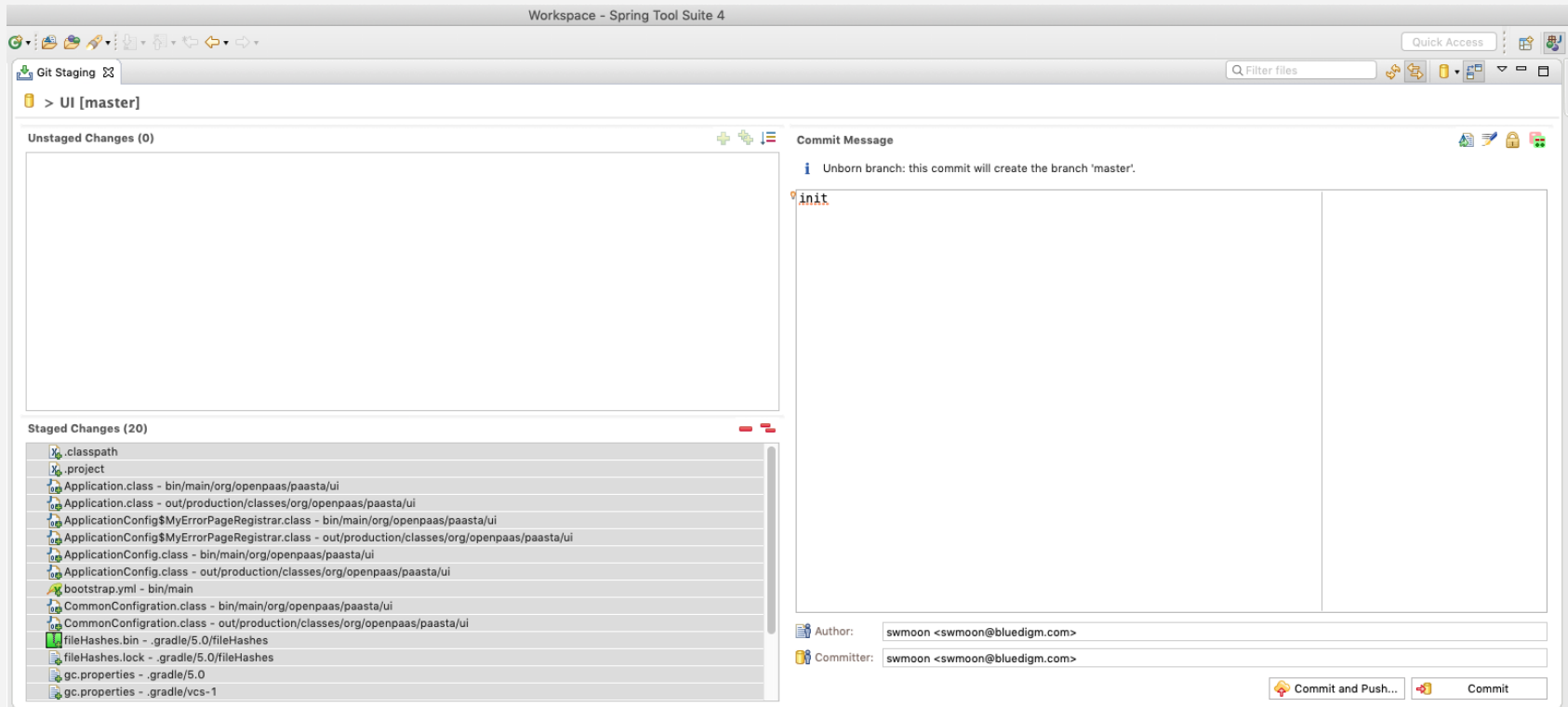


설명

제공된 예제소스를 형상관리로 받은 폴더에 복사한다.
Team -> Commit을 진행한다.

실습

SCM 구성 실습

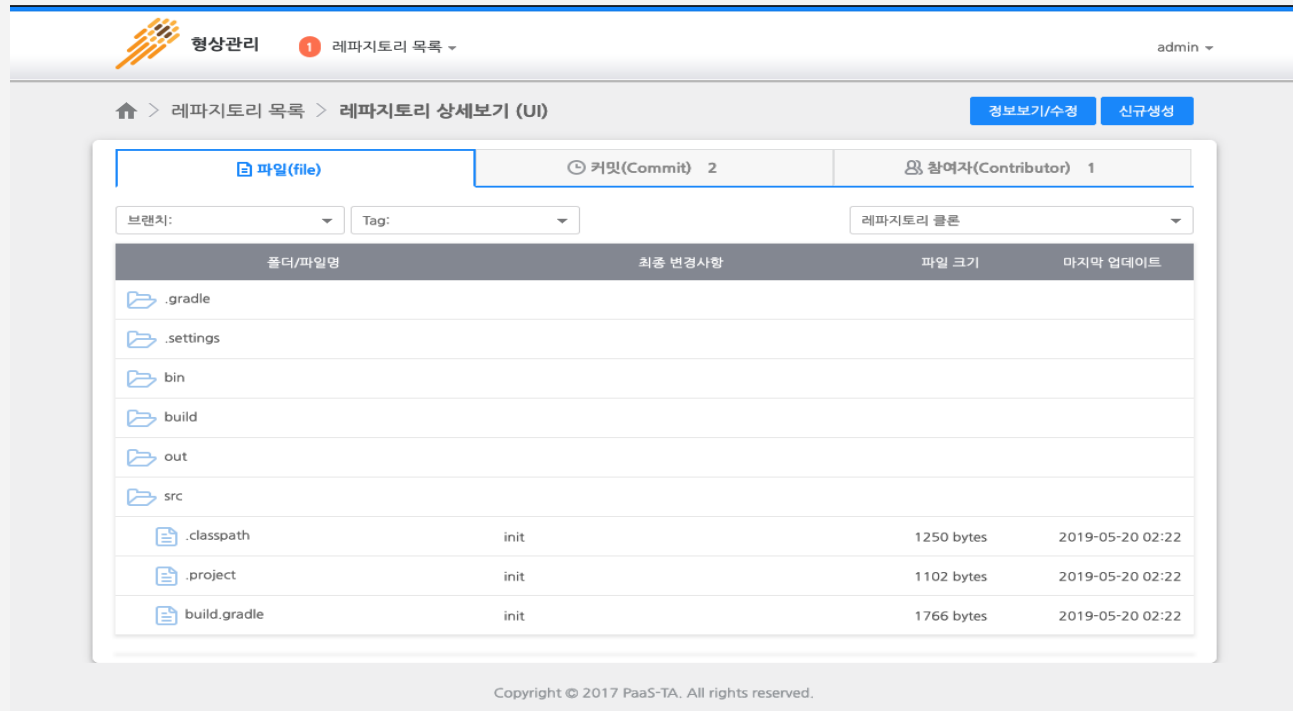


설명

추가할 내용을 선택 후 Commit and Push를 진행한다.

실습

SCM 구성 실습



The screenshot shows a web interface for repository management. At the top, there is a navigation bar with the text '형상관리' (Configuration Management) and a notification icon with the number '1'. The user is logged in as 'admin'. The main content area shows the repository path '레파지토리 상세보기 (UI)' and two buttons: '정보보기/수정' (View Info/Modify) and '신규생성' (New Creation). Below this, there are tabs for '파일(file)', '커밋(Commit) 2', and '참여자(Contributor) 1'. There are also dropdown menus for '브랜치:' (Branch), 'Tag:', and '레파지토리 클론' (Repository Clone). The main part of the interface is a table listing files and folders. The table has columns for '폴더/파일명' (Folder/File Name), '최종 변경사항' (Last Change), '파일 크기' (File Size), and '마지막 업데이트' (Last Update). The files listed are .gradle, .settings, bin, build, out, src, .classpath, .project, and build.gradle.

폴더/파일명	최종 변경사항	파일 크기	마지막 업데이트
.gradle			
.settings			
bin			
build			
out			
src			
.classpath	init	1250 bytes	2019-05-20 02:22
.project	init	1102 bytes	2019-05-20 02:22
build.gradle	init	1766 bytes	2019-05-20 02:22

Copyright © 2017 PaaS-TA. All rights reserved.

설명

형상관리에 정상적으로 소스가 업로드 된 것을 확인한다.





실습

SCM 구성 실습

형상관리 2 레파지토리 목록 admin

레파지토리목록 신규생성

레파지토리 명 레파지토리 형상관리 전체 전체 레파지토리 최신 업데이트 순 총 2 건

	config config 🕒 마지막 업데이트 : 2019-05-20 04:33:11 🕒 생성일 : 2019-05-20 04:33:11	 소유자
	UI Paas-TA UI 🕒 마지막 업데이트 : 2019-05-20 02:22:46 🕒 생성일 : 2019-05-20 01:42:44	 소유자

Copyright © 2017 PaaS-TA. All rights reserved.

설명

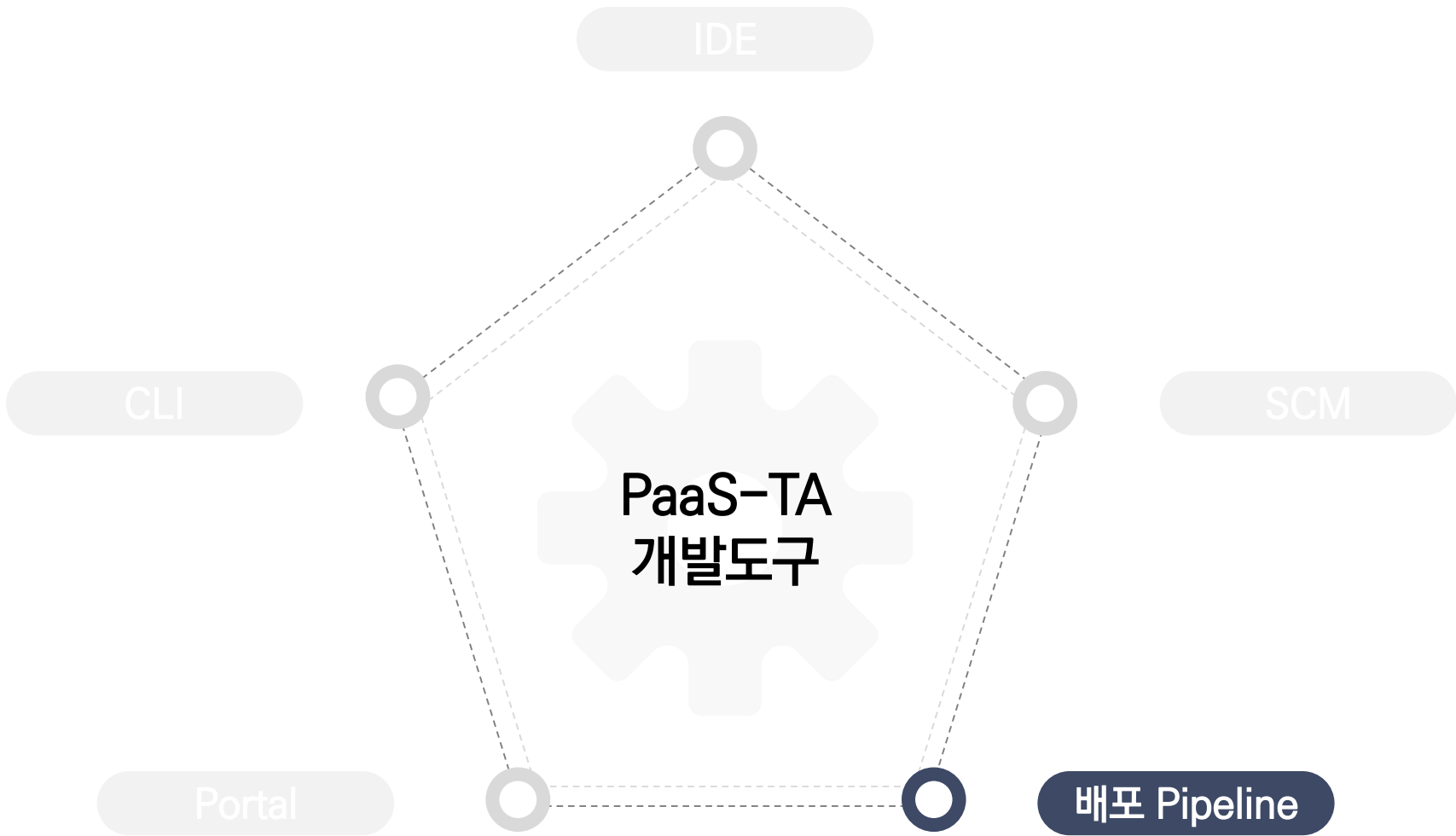
샘플로 제공된 Config를 형상관리에 추가한다.



MEMO



배포 Pipeline



» Pipeline 소개



Pipeline

생산 라인 등과 같이 여러 공정별로 생산 라인이 나열되어 있고
동시에 공정별 프로세서가 가능하게 하는 것

시스템의 효율을 높이기 위해 명령문을 수행하면서
몇 가지의 특수한 작업들을 병렬 처리하도록 설계된 하드웨어 기법

배포 Pipeline

» Pipeline 소개

배포 Pipeline 기능 및 역할

각 프로세스들을 저장한 파일 이름들 사이를 수직선으로 분리시키면 셀(cell)이 인식함

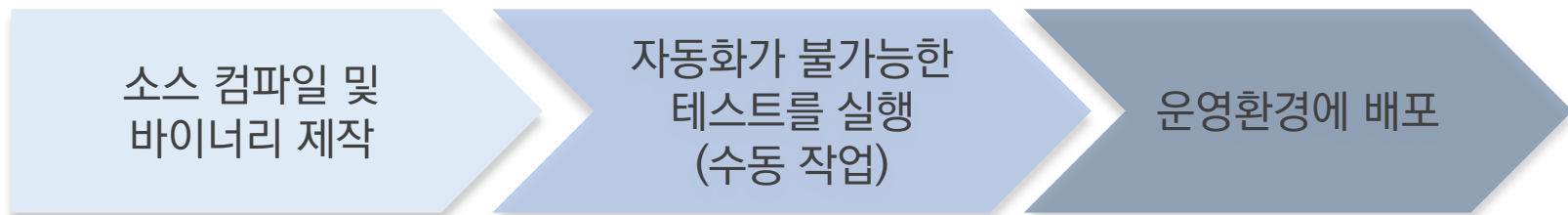
즉, 수직선 왼쪽 파일의 출력이 오른쪽 파일의 입력으로 연결됨



» Pipeline 소개

배포 Pipeline 기능 및 역할

배포 파이프라인은 지속적 배포(Continuous Delivery)의 핵심 요소



자동화된 빌드 및 테스트 환경을 구축할 때 빠른 피드백을 받을 수 있도록
빌드를 여러 단계로 쪼갬

각 단계를 거칠수록 시간이 더 오래 걸리지만 배포본에 대한 확신이 증가함

각 단계들은 자동으로 실행될 수도 있고, 실행 전에 관리자의 승인이 필요할 수도 있음

빌드 시간 단축을 위해 여러 서버에서 동시에 수행될 수도 있음

» Pipeline 소개

배포 Pipeline 기능 및 역할

Pipeline은 **운영환경에 문제를 일으키는 변경사항을 감지**해 내는 일을 함

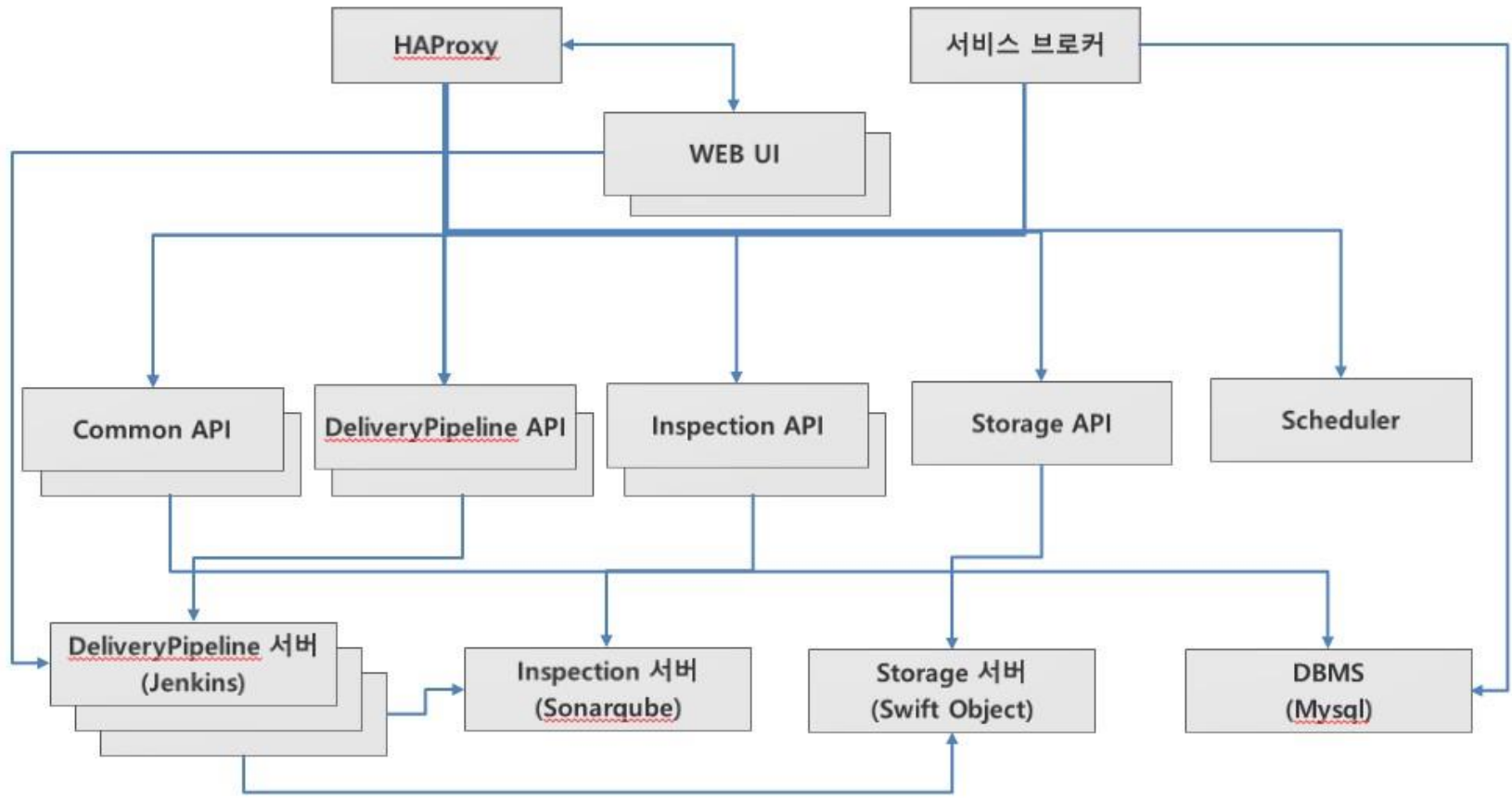
문제는 성능, 보안 혹은 사용성 등 여러 가지 이슈가 될 수 있음

소프트웨어 배포에 관여하는 다양한 그룹들이 서로 협력하도록 만들고, 모든 사람이 시스템 변경 사항의 흐름을 철저하게 추적할 수 있는 환경을 제공해야 함

배포 Pipeline

» SCM을 이용한 Pipeline 구성

시스템 구성도



» SCM을 이용한 Pipeline 구성

시스템 구성도

VM 명	인스턴스 수	vCPU 수	메모리(GB)	디스크(GB)
HAProxy	1	1	2	Root 4G
WEB UI	N	1	2	Root 4G
Service broker	1	1	2	Root 4G
Common API	N	1	2	Root 4G
DeliveryPipeline API	N	1	2	Root 4G
Inspection API	N	1	2	Root 4G
Storage API	1	1	2	Root 4G
Scheduler	1	1	2	Root 4G
DeliveryPipeline	N	1	2	Root 8G + 영구디스크 10G
Inspection	1	1	2	Root 4G
Storage	1	1	4	Root 4G + 영구디스크 50G
DBMS(mariadb)	1	1	4	Root 6G + 영구디스크 4G
Postgres	1	1	2	Root 6G + 영구디스크 4G

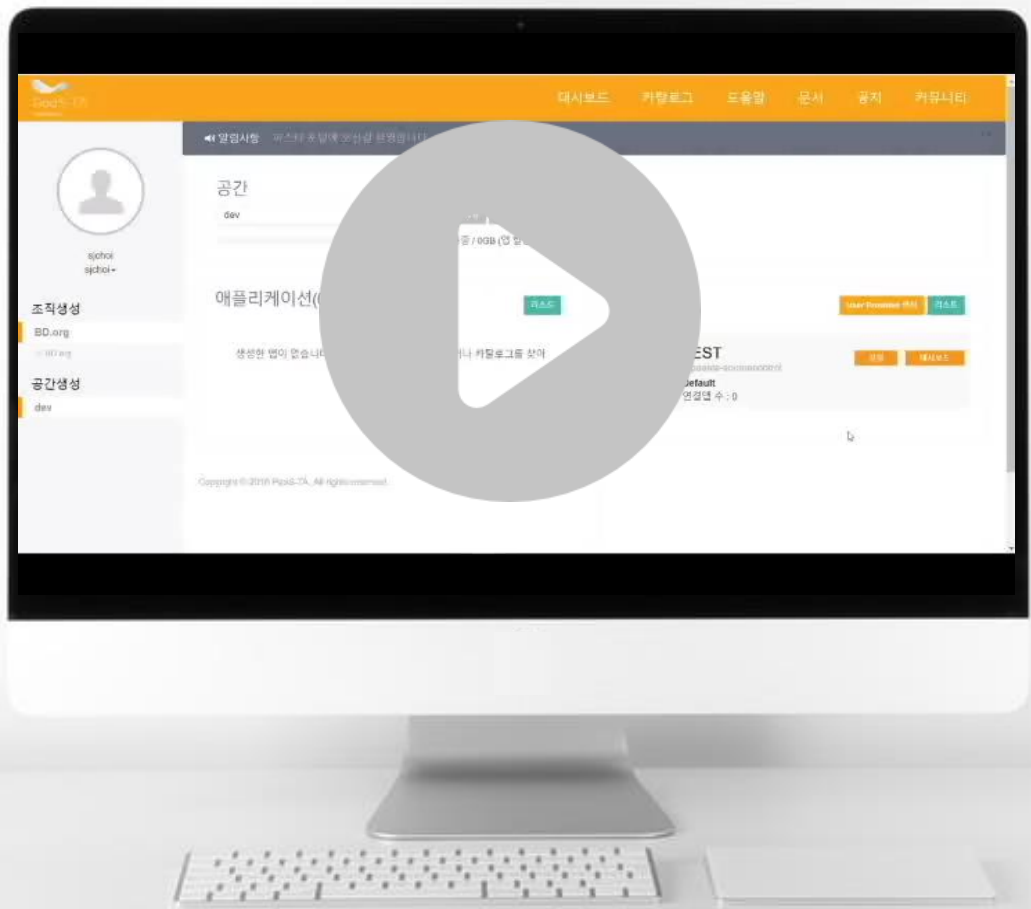
» 개발 및 배포 예시

Scheduler	<ul style="list-style-type: none"> 배포파이프라인에서 소스 변경시 빌드를 진행하도록 설정한 Job을 주기적으로 체크하는 애플리케이션
배포파이프라인 서비스 브로커	<ul style="list-style-type: none"> PaaS-TA와 배포파이프라인 서비스 간에 중계 역할을 하는 애플리케이션
WEB UI	<ul style="list-style-type: none"> 배포파이프라인 WEB UI 대시보드는 웹 인터페이스를 통해 배포파이프라인을 구성 할 수 있도록 하는 웹 애플리케이션 Common/DeliveryPipeline/Inspection/Storage API 애플리케이션의 REST API 를 호출하여 그 결과를 화면에 제공한다. 화면을 분리하여 UI 프레임워크 변경 시 API 서버와 영향도가 없이 구성
Inspection API	<ul style="list-style-type: none"> 품질 관리 및 결과에 필요한 REST API를 제공
Common API	<ul style="list-style-type: none"> 배포파이프라인에서 필요한 공통 기능과 DBMS 메타 데이터 제어에 필요한 REST API를 제공
DeliveryPipeline API	<ul style="list-style-type: none"> 지속적인 통합 및 배포에 필요한 REST API를 제공
Storage API	<ul style="list-style-type: none"> 배포파이프라인에서 빌드에 아티팩트 파일을 관리하는 REST API를 제공

배포 Pipeline

» 개발 및 배포 예시

SCM에 등록된 소스를 Pipeline을 이용해 배포



실 습

배포 파이프라인 구성 실습

실습 사이트

playpark.paas-ta.org

실습 소요시간

약 30분

진행 방법

배포 파이프라인 서비스를 생성하여 SCM소스를 배포할 수 있도록 설정 후 배포 해본다.

피드백 포인트

각자의 서비스를 생성했는지 확인해본다.

필요 도구

각자의 계정

실습

배포 파이프라인 구성 실습

The screenshot shows the PaaS-TA Portal dashboard. At the top, there is a navigation bar with '대시보드', '카탈로그', and '문서' links. Below the navigation bar, the dashboard displays '0 MB / 할당량 100 GB' and a progress bar. The main content area is divided into '애플리케이션' and '서비스' tabs. The '서비스' tab is active, showing a list of services. The 'pipeline' service is highlighted, and its details are shown in a card. The '대시보드' link in the service details is circled in red. Other services shown include 'mysql' and 'source-controller'.

서비스명	연결 앱	서비스명	제공서비스	대시보드
mysql	mysql	Mysql-DB	Mysql-Plan1-10con	
pipeline	delivery-pipeline	delivery-pipeline-shared		대시보드
source-controller	p-paasta-sourcecontrol	Default		대시보드

설명

PaaS-TA Portal에서 서비스 탭으로 이동한다.
파이프라인의 대시보드를 클릭한다.

실습

배포 파이프라인 구성 실습

The screenshot shows a web application interface for pipeline management. At the top left is the logo '배포파이프라인' (Deployment Pipeline). The navigation menu includes '파이프라인 목록' (Pipeline List), '파이프라인 관리' (Pipeline Management), and '품질 관리' (Quality Management). The user is logged in as 'admin 님, 환영합니다.' with a '사용자 관리' (User Management) link. The breadcrumb shows '모든 파이프라인' (All Pipelines). A search bar contains '파이프라인 명 검색' and a dropdown menu is set to '최신 생성일 순' (Latest creation date). A red box highlights the '신규생성' (New Generation) button. The main content area displays '조회 된 데이터가 없습니다.' (No data found). The footer contains the copyright notice 'Copyright © 2017 PaaS-TA. All rights reserved.'

설명

신규생성 버튼을 클릭한다.

실습

배포 파이프라인 구성 실습

0 파이프라인 목록 ▾ 파이프라인 관리 ▾ 품질 관리 ▾

파이프라인 신규생성

파이프라인 명 **(+필수)**

UI

파이프라인 설명 (선택)

UI

생성

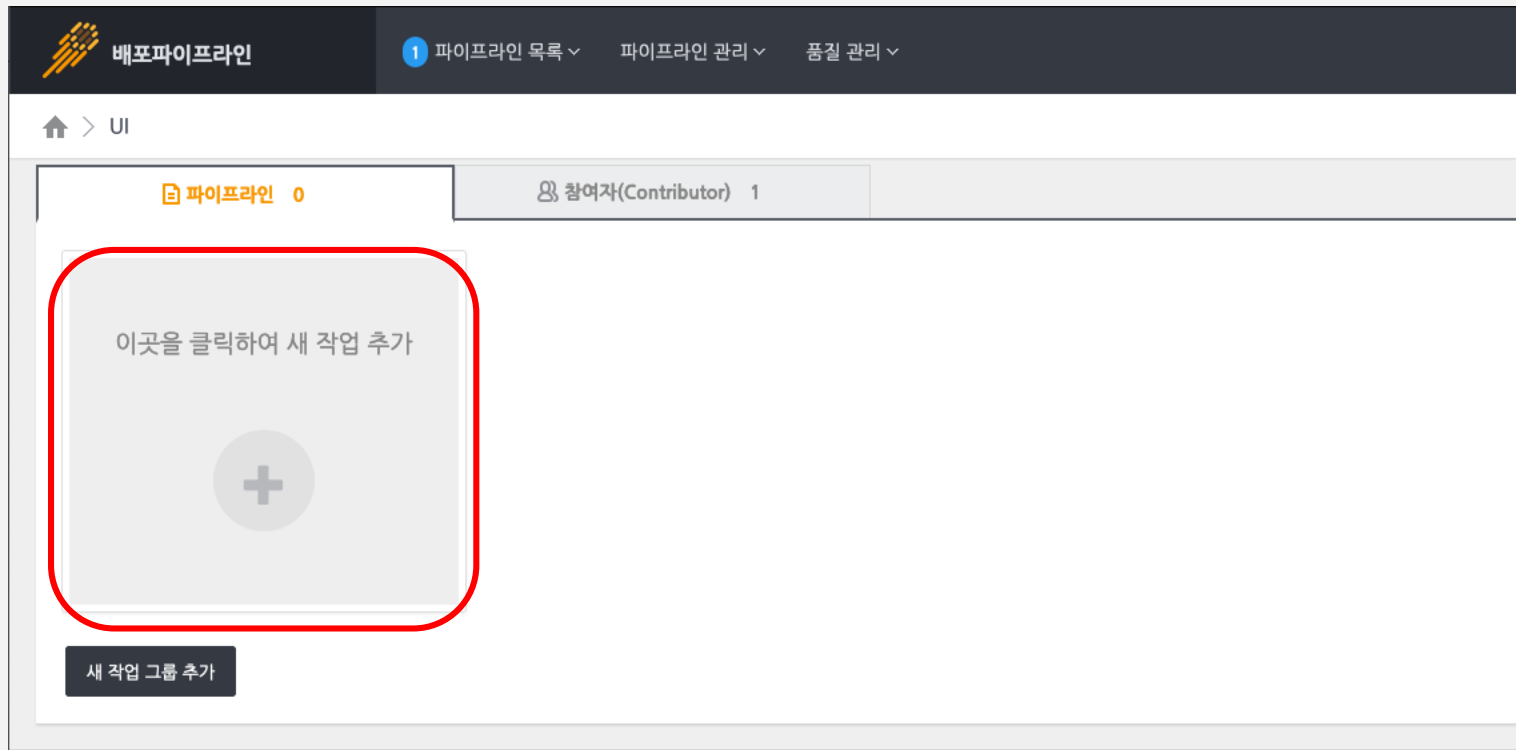
취소

설명

필수 정보를 입력한 후에 생성을 진행한다.

실습

배포 파이프라인 구성 실습



설명

생성된 파이프라인을 선택후 새 작업 추가를 진행한다.

실습

배포 파이프라인 구성 실습

구성상세

내 작업	<input type="text" value="빌드 job"/>
작업 유형 (Job)	<input type="text" value="빌드 (Build)"/>
• 빌더유형 (type)	
	<input type="text" value="Gradle"/>
<input type="checkbox"/> 이 작업에 실패하는 경우 연결 작업 실행 중단	

입력 유형	<input type="text" value="SCM (Git)"/>
	<input type="text" value="admin"/>
	<input type="text" value="....."/>
	<input type="text" value="http://admin@115.68.47.176:8084/scm/git/UI"/>
<input type="button" value="조회"/>	
• 브랜치 (Branch)	
	<input type="text" value="master"/>

• 작업 트리거 (Job Trigger)

- 이전 작업(job)을 완료할 때 작업 실행
- 변경사항을 푸시할 때마다 작업 실행
- 이 작업(job)을 수동으로 실행할 때만 작업 실행
- 이 작업(job)을 새 작업 그룹으로 구성

설명

작업 유형 빌드 선택, 빌드유형 Gradle 선택, SCM(GIT) 선택 후 이전 실습에서 만든 Respository 주소와 계정과 패스워드를 입력 후 조회한다. 정상적으로 조회가 될 경우 저장한다.

실습

배포 파이프라인 구성 실습

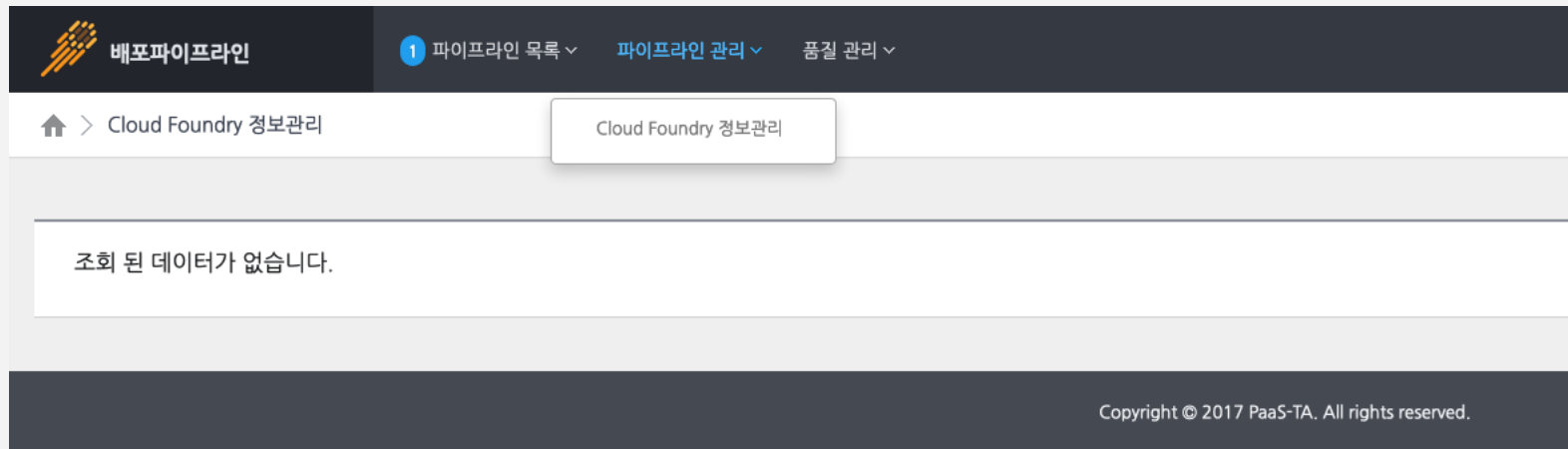
The screenshot shows a web interface for managing CI/CD pipelines. At the top, there is a dark navigation bar with the logo and name '배포파이프라인' (Deployment Pipeline) on the left, and a menu with items '1 파이프라인 목록' (1 Pipeline List), '파이프라인 관리' (Pipeline Management), and '품질 관리' (Quality Management). Below the navigation bar is a breadcrumb trail '홈 > UI'. The main content area is titled '파이프라인 1' (Pipeline 1) and shows '참여자(Contributor) 1' (1 Contributor). A job titled '1. 빌드 job' (1. Build job) is displayed with a '마지막 커밋(Recent Commit)' (Recent Commit) section containing the URL 'http://admin@115.68.47.176:8084/scm/git/UI' and a commit hash '72a5823960f5bfe021c7...' with a timestamp '2019-05-20 04:38:32'. Below this is a '작업(Job)' (Job) section with a 'Build (실행 전)' (Build (Before Execution)) job listed as '실행 전' (Before Execution). At the bottom of the job list are buttons for '+ 추가' (Add), '복제' (Copy), and '삭제' (Delete). A '새 작업 그룹 추가' (Add New Job Group) button is located at the bottom left of the main content area.

설명

정상적으로 등록이 되었을 경우, 다음과 같은 화면이 나온다.

실습

배포 파이프라인 구성 실습

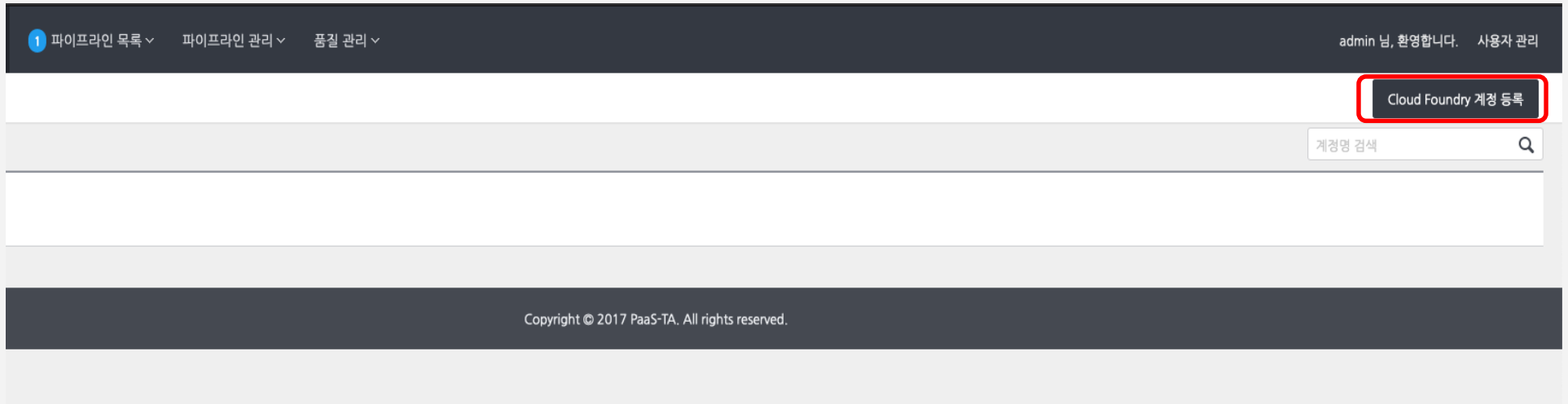


설명

파이프라인 관리 -> Cloud Foundry 정보관리 페이지로 이동한다.

실습

배포 파이프라인 구성 실습



설명

Cloud Foundry 계정 등록을 클릭한다.

실습

배포 파이프라인 구성 실습

1 파이프라인 목록 ▾ 파이프라인 관리 ▾ 품질 관리 ▾

계정 등록

계정명 admin

아이디 admin

비밀번호

URL https://api.115.68.46.188.xip.io URL 관리

설명 (선택)

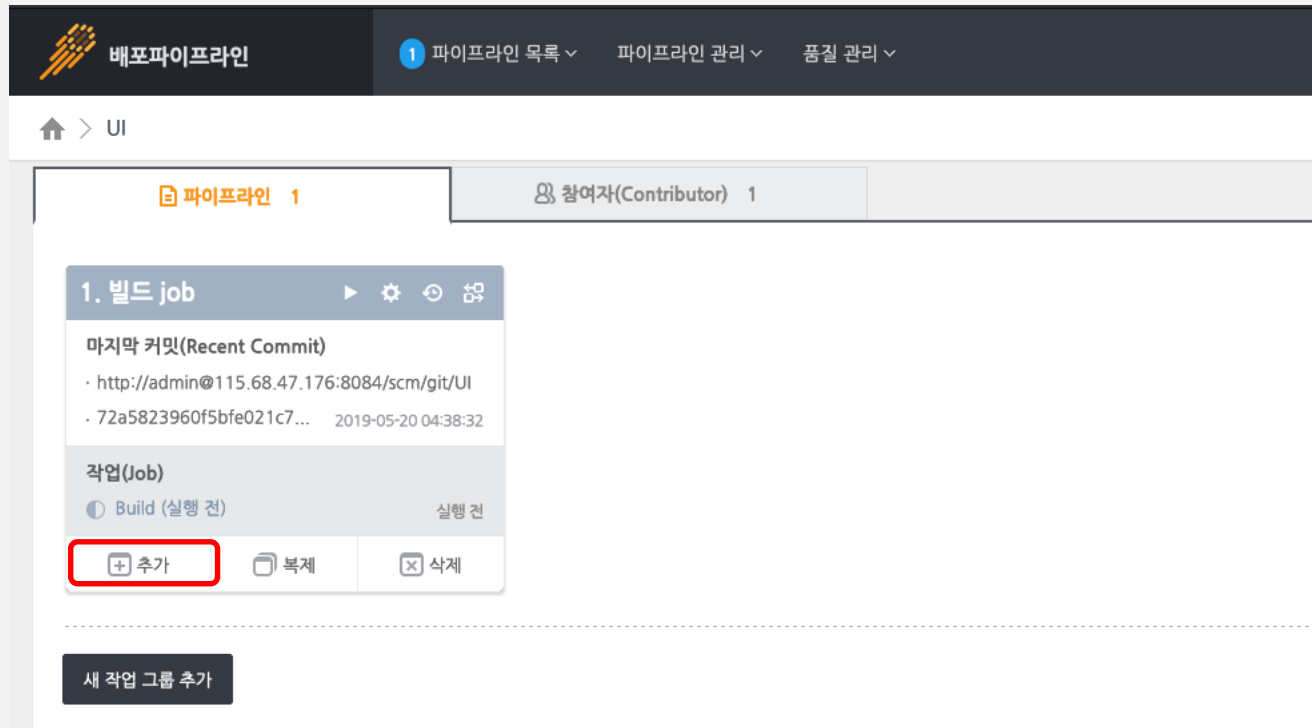
등록 취소

설명

계정정보를 입력 후 URL 관리 선택하여, CF API주소를 추가한다.
추가 후 등록한다.

실습

배포 파이프라인 구성 실습



배포파이프라인

1 파이프라인 목록 ▾ 파이프라인 관리 ▾ 품질 관리 ▾

홈 > UI

파이프라인 1

참여자(Contributor) 1

1. 빌드 job

▶ ⚙️ ↻

마지막 커밋(Recent Commit)

- http://admin@115.68.47.176:8084/scm/git/UI
- 72a5823960f5bfe021c7... 2019-05-20 04:38:32

작업(Job)

- Build (실행 전) 실행 전

+ 추가 복제 삭제

새 작업 그룹 추가

설명

파이프라인 대시보드로 이동하여, 추가버튼을 클릭하여, 배포 잡을 생성한다.

실습

배포 파이프라인 구성 실습

구성상세

내 작업: 배포 job

작업 유형 (Job): 배포 (Deploy)

유형 (type): 개발배포

Cloud Foundry 정보: admin

조직/공간: 조직명: system, dev, UI

MANIFEST 사용 여부: N

MANIFEST.YML: ---\napplications:\n- name: %APP_NAME%\nmemory: 512M

입력 유형: 1. 빌드 job

설명

작업 유형 -> 배포, 유형 -> 개발배포, 조직 및 공간 선택 후 저장한다.

실습

배포 파이프라인 구성 실습

The screenshot shows the Bamboo web interface. At the top, there's a dark header with the '배포파이프라인' logo and navigation links: '1 파이프라인 목록', '파이프라인 관리', and '품질 관리'. Below the header, a breadcrumb trail shows 'UI'. The main content area is titled '파이프라인 2' and shows '참여자(Contributor) 1'. Two job cards are visible: '1. 빌드 job' and '2. 배포 job'. The '1. 빌드 job' card has a play button icon highlighted with a red box. Below each job card, there are details for '마지막 커밋(Recent Commit)' and '마지막 배포(Recent Deploy)', along with a '작업(Job)' section listing 'Build (실행완료)' and 'Deploy (실행완료)'. At the bottom, there's a '새 작업 그룹 추가' button.

설명

빌드 잡에 시작버튼을 클릭하여, 빌드를 진행한다.
정상적으로 완료 된 경우 다음 그림과 같이 초록색으로 변한다.

실습

배포 파이프라인 구성 실습

The screenshot displays the PaaS-TA Portal dashboard. At the top, there is a navigation bar with the PENNE logo and menu items: 대시보드 (Dashboard), 카탈로그 (Catalog), and 문서 (Documents). On the right side of the navigation bar, there are icons for user profile, notifications, and a globe. Below the navigation bar, the main content area is titled 'Dashboard' and shows the current environment as '대한민국' (South Korea), 'system', and 'dev'. A progress bar indicates '1 GB / 할당량 100 GB' (1 GB / 100 GB allocated) with a 1% completion mark. Below this, there are two tabs: '애플리케이션' (Applications) and '서비스' (Services). The '애플리케이션' tab is active, showing a card for an application named 'UI'. The card includes a list of resources: '인스턴스 : 1' (Instances: 1), '디스크 : 1 GB' (Disk: 1 GB), and '메모리 : 1 GB' (Memory: 1 GB). A green 'STARTED' button is visible. Below the resource list, there are two progress bars: 'DISK' at 15% and 'Memory' at 22%. To the right of the application card, there is a large empty box with a plus sign (+).

설명

PaaS-TA Portal 대시보드로 이동하여, 배포된 앱이 정상적으로 시작되었는지 확인한다.

실 습

Eureka, Config서버 구성 실습

실습 사이트

playpark.paas-ta.org

실습 소요시간

약 3시간

진행 방법

이전에 진행한 실습을 통합적으로 이용하여, Eureka는 CFCLI로, Config서버는 형상관리, 배포파이프라인을 통하여, 배포를 진행한다.

피드백 포인트

각자의 서비스를 생성했는지 확인해본다.

필요 도구

각자의 계정

／ 핵심정리 ／

- ✓ SCM은 codebase 프로젝트 관리를 위한 툴로서 PaaS-TA에서 서비스로 제공하고 있어 쉽게 생성하여 사용할 수 있다.
- ✓ 배포 파이프라인은 CI/CD tool로서 배포 관리를 손쉽게 해주는 서비스이다.



MEMO





MEMO

