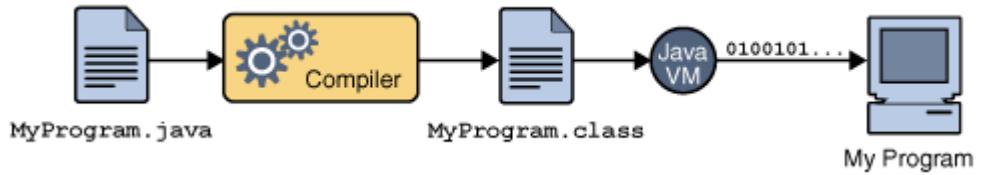
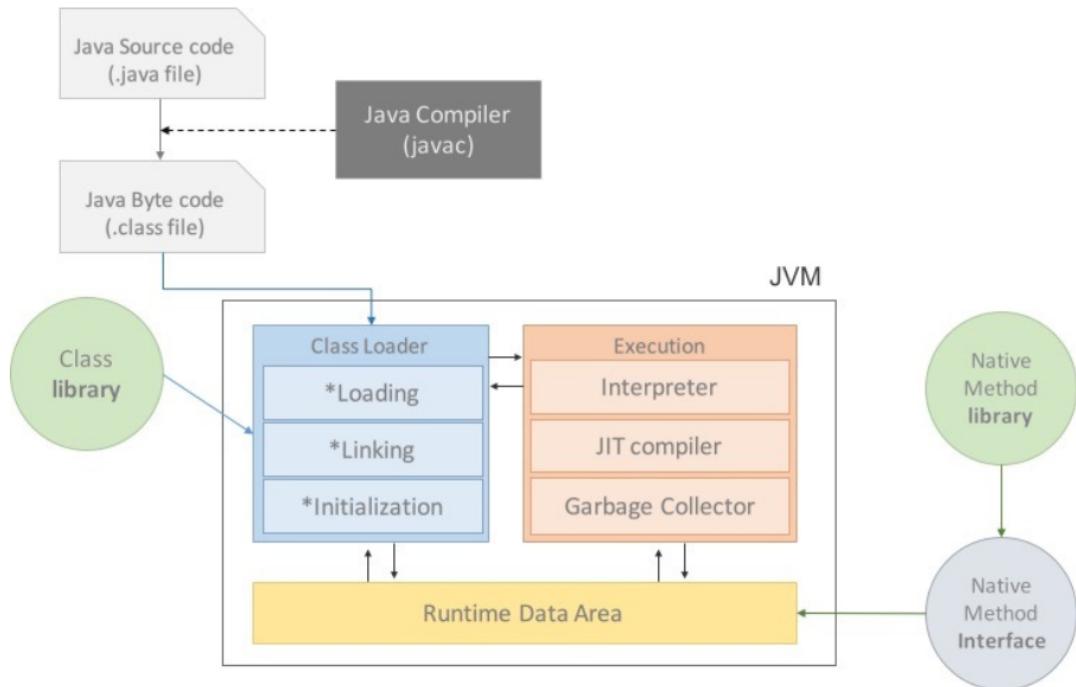


# Caching Configuration Guide - 10.11

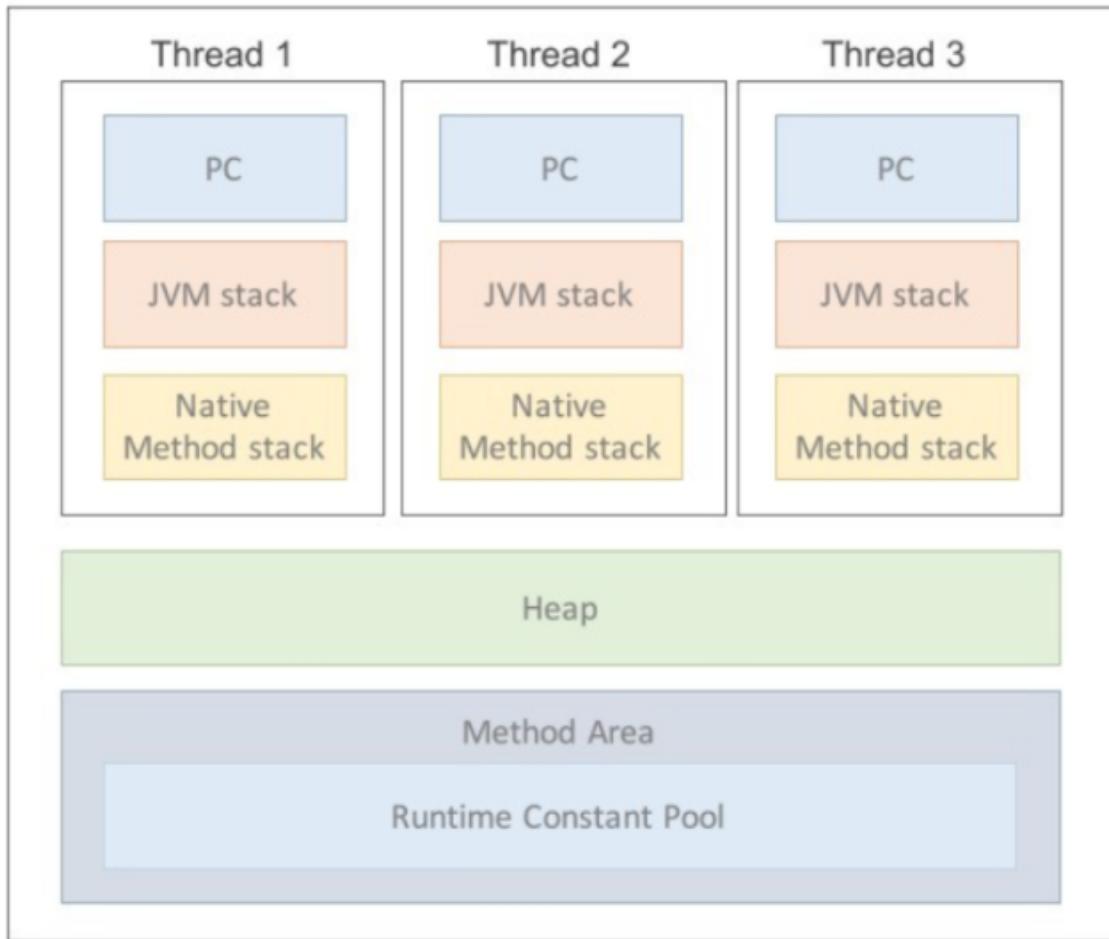
- References
  - [JVM 개념정리](#)
  - [JVM Heap / Off-Heap 영역](#)
  - [캐시, 캐싱이란](#)
  - [아키텍처 설계 : 분산캐시 - EHCache + TSA](#)
  - [webMethods Clustering Guide - 10.11 \(API + EAI\(IS\)\) 참고 \(Notion\)](#)
  - [webMethods Clustering \(Blog\)](#)
  - [Documentation - webMethods Integration Server 10.11](#)
- JVM (Java Virture Machine)



- 자바를 실행시키기 위한 가상 컴퓨터
- OS에 종속적이지 않고 CPU가 자바를 인식하여 실행 할 수 있게 하는 가상 컴퓨터



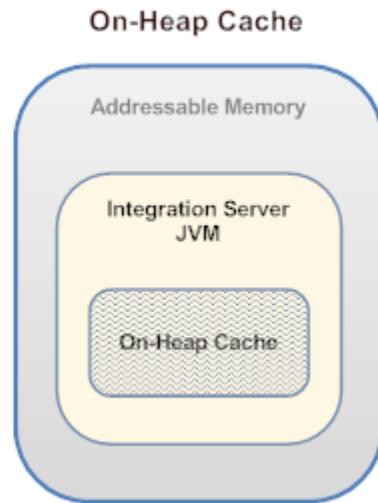
- JVM 런타임 데이터 영역 (Runtime Data Area)



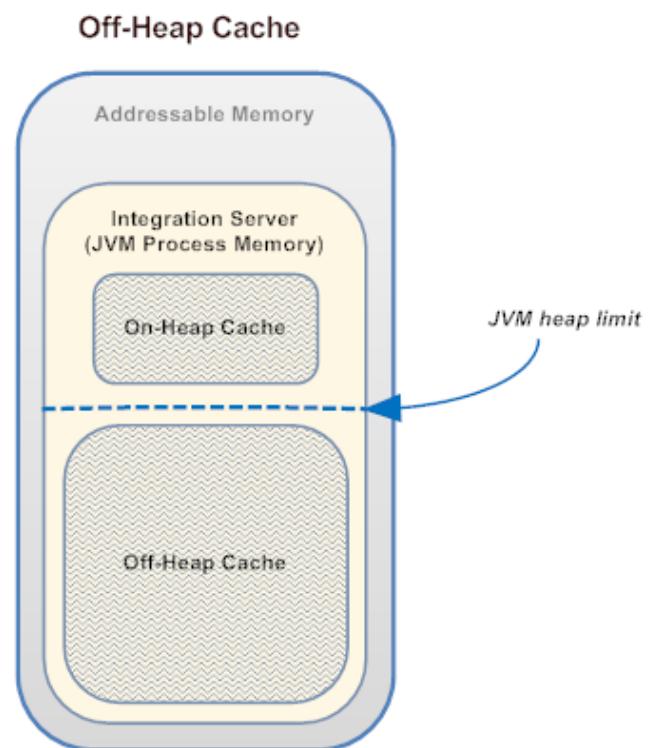
- 프로그램을 실행시키기 위해 메인 OS로 부터 할당받은 메모리 공간
- Stack 영역
  - 프로그램 실행 과정에서 임시로 할당 되었다가 메소드를 빠져나가며 바로 소멸 되는 특성의 데이터를 저장하기 위한 영역
  - 매개변수, 지역변수, 리턴값, 연산시 일어나는 값 등등
- Heap 영역
  - 객체를 저장하는 가상 메모리 공간
  - new 연산자로 생성되는 객체와 배열을 저장
  - Class Area(Static Area)에 올라온 클래스 들만 객체로 생성 가능
- 가비지 컬렉터 (GC)
  - 더이상 사용하지 않는 인스턴스를 찾아 메모리에서 삭제

- On-Heap vs Off-Heap 메모리 영역
  - 자바 프로그램에서 생성된 객체는 Heap 메모리 영역에 생성된다.
  - 힙 영역에 생성된 객체는 GC(Garbage Collector)의 관리를 받으며 사용되지 않는 경우 GC에 의해 회수
  - On-Heap 메모리 영역
    - 일반적인 Heap 메모리 영역
    - GC에 의해 회수 되는 메모리 영역
  - Off-Heap 메모리 영역
    - Heap 밖에 저장한다는 의미
    - GC에 영향을 받지 않는 메모리 영역
    - Off-Heap 영역에 객체를 저장하기 위해서는 객체를 Serialize(직렬화) 해야 한다
    - EHCache, Terracotta BigMemory 는 대표적인 Off-Heap 스토어 캐시 라이브러리로 라이브러리 단에서 메모리를 관리해 준다.
- 캐싱 이란
  - 동일한 데이터에 반복해서 접근해야 하거나 많은 연산이 필요한 일일때, 결과를 빠르게 이용하고자 성능이 좋은 가까운 곳에 저장하는 것
  - 캐시는 컴퓨터의 성능을 향상 시키기 위해 사용되는 메모리를 말한다
- Terracotta Ehcache 란
  - Integration Server 에서 사용하는 표준 기반 캐싱 API
  - 데이터가 필요할 때 마다 데이터베이스 또는 다른 백엔드 시스템에서 데이터를 검색할 필요 없이 애플리케이션이 메모리에서 자주 사용하는 데이터를 가져올 수 있다.
  - webMethods Integration Server 는 Terracotta EHCeache 를 사용하여 자체 내부 작업과 관련된 데이터를 캐시한다.
  - Built-in Service 로 pub.cache 에서 공용 서비스 제공하고 있음

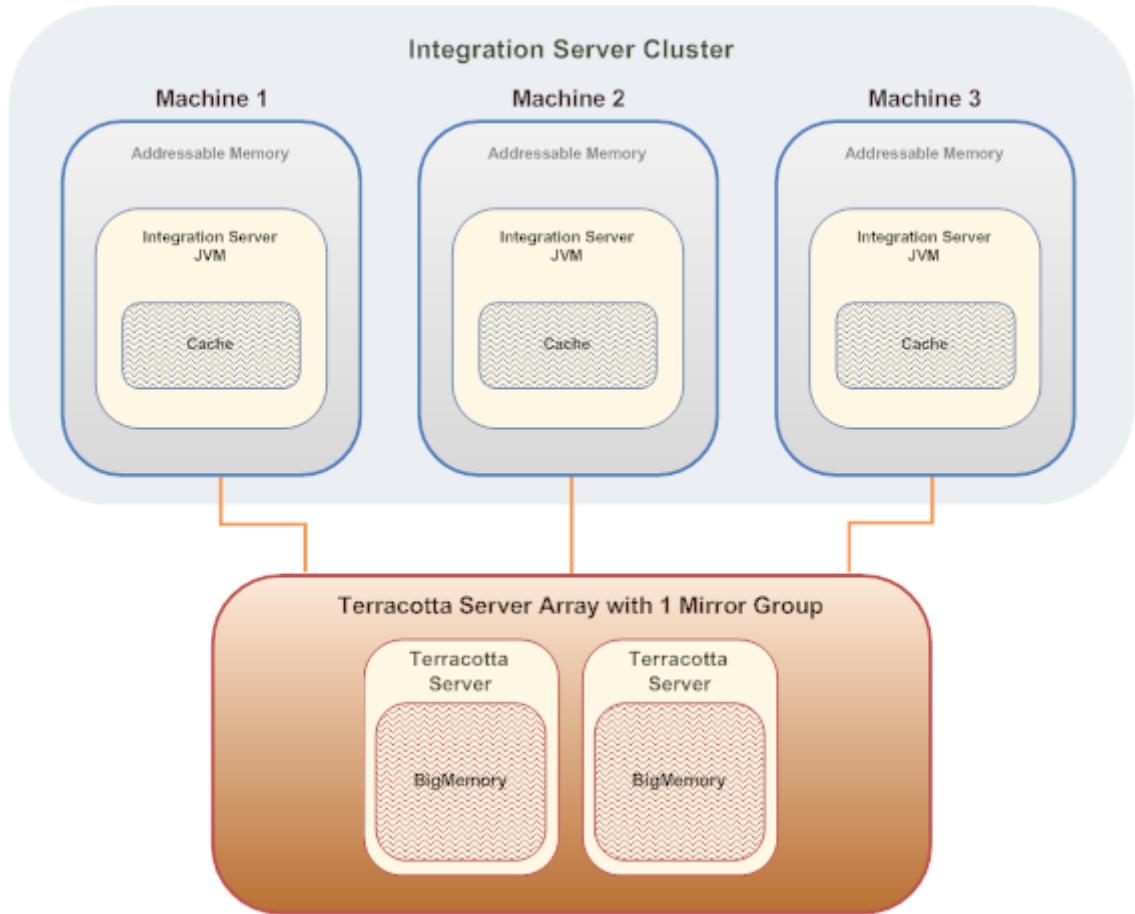
- EHCACHE 캐싱 구성은 On-Heap, 로컬 디스크 저장소, 빅메모리, TSA 에 구성 가능하다.
- On-Heap 캐시



- 빅메모리 캐시 (Off-Heap Cache)



- webMethods 의 캐시와 캐시 관리자 (Cache, Cache Manager)
  - webMethods 는 Terracotta EHCache를 사용하며, 캐시에 Key - Value 쌍을 저장 가능
  - Key와 Value 값은 모두 JAVA 객체
  - IData 객체는 Serialize 가 가능, Serialize하여 캐시에 저장해야 한다.
    - 단, 캐시할 IData에 직렬화 불가능한 스트림, XML, Node 또는 사용자 지정 개체와 같은 개체가 포함되어 있는 경우 예외를 Throw
  - 캐시 되어있는 요소는 키로 식별
  - 캐시에는 연결 된 캐시 관리자가 있으며, 캐시 관리자는 캐시 세트를 함께 시작하고 종료할 수 있는 하나의 관리 단위 그룹으로 그룹화를 위한 컨테이너 역할
    - 즉, 캐시를 생성하기 위해서는 캐시 관리자를 생성 후 해당 캐시 관리자에서 캐시를 생성
  - Integration Server 및 기타 SAG 제품은 자체 내부 프로세스에서도 Terracotta EHCache 를 사용 (시스템 캐시 라고 함)
    - SoftwareAG. 으로 시작하는 시스템 캐시 관리자에 속하는 캐시들
- Terracotta Server Array
  - TSA (Terracotta Server Array) 를 사용하면 분산 캐시를 만들 수 있다.
  - 분산 캐시는 여러 Integration Server 에서 공유 가능



- 캐시의 전체 복사본이 TSA에 존재
- TSA는 하나 이상의 테라코타 서버로 구성되며(N중화), 방대한 양의 데이터(수 테라바이트 단위)를 캐시 가능하다.
- 캐시데이터는 스트라이핑 기술을 사용하여 테라코타 서버에 분산되며, Integration Server의 테라코타 클라이언트는 IS와 TSA 간 캐시 상호작용을 관리한다.

#### ※ 분산 캐시 테스트

- 테스트 Setting
  - Clustering (Terracotta Server Array) 되어 있는 Integration Server 두 대에서 테스트
    - 192.168.1.167:5555(TIS01), 192.168.1.168:5555(TIS02)

## Clustering

General	
Clustering Status	 Enabled
Cluster Name	skbs_test
Session Timeout	60 minutes
Action On Startup Error	Start as Stand-Alone Integration Server
Terracotta Server Array URLs	192.168.1.167:9510,192.168.1.168:9510

Cluster Hosts												
Name	Address	Server Start Time	Server Up Time	Total Memory (KB)	Free Memory (KB)	Used Memory (KB)	Current Server Threads	Max Server Threads	Current System Threads	Max System Threads	Total Requests	Average Duration (ms)
TIS01:5555	192.168.1.167:5555	2022-10-26 10:10:58 KST	1 day 0h:29m:11s	417792	106689	311103	0	1	558	560	0	0
TIS2:5555	192.168.1.168:5555	2022-10-26 10:10:44 KST	1 day 0h:28m:49s	472064	78477	393587	0	1	555	559	0	0

- **STEP 1.** Cache Manager & Cache 생성
  - 클러스터링 된 각 서버(TS01, TS02)에 Cache Manager와 Cache 생성 (이름은 동일하게 구성)
- Cache Manager 생성
  - Settings > Caching 화면

Administration > Caching

## Caching



System Cache Managers	
Name	
<a href="#">SoftwareAG.IS.ART.skbs_test</a>	
<a href="#">SoftwareAG.IS.Core.skbs_test</a>	
<a href="#">SoftwareAG.IS.IData</a>	
<a href="#">SoftwareAG.IS.Services</a>	
<a href="#">SoftwareAG.IS.WMN.skbs_test</a>	

## Add cache manager

- [Return to Caching](#)

Cache Manager Configuration	
Name	SessionID_CacheManager
Terracotta Server Array URLs	<input type="text" value="192.168.1.167:9510,192.168.1.168:9510"/> Use commas (,) to separate entries.
Rejoin	<input checked="" type="checkbox"/>
<input type="button" value="Save Changes"/>	

- Terracotta Server URLs 에는 tc-config.xml에 설정한 대로 입력
  - 각 서버 IP : TSA Port 나열
- 생성 된 Cache Manager에 Cache 생성

## Caching

- [Add Cache Manager](#)

System Cache Managers				
Name	Reload	Start/Shutdown	Delete	
SoftwareAG.IS.ART.skbs_test				
SoftwareAG.IS.Core.skbs_test				
SoftwareAG.IS.IData				
SoftwareAG.IS.Services				
SoftwareAG.IS.WMN.skbs_test				

Public Cache Managers				
Name	Reload	Start/Shutdown	Delete	
SessionID_CacheManager		<a href="#">Start</a>		

## View cache manager

- [Return to Caching](#)
- [Add Cache](#)
- [Edit Cache Manager Configuration](#)

Cache Manager Configuration	
Name	SessionID_CacheManager
Terracotta Server Array URLs	192.168.1.167:9510,192.168.1.168:9510
Rejoin	Yes
BigMemory Allocated	None

### Add/Modify cache

[Return to SessionID\\_CacheManager](#)

**Cache Configuration**

Cache Name	sessionId
Maximum Elements in Memory	1
External	<input checked="" type="checkbox"/>
Time to Live	0 seconds
Time to Idle	0 seconds
Overflow to Disk	<input type="checkbox"/>
Persist to Disk	<input type="checkbox"/>
Maximum Entries Local Disk	0

**Cache Configuration Advanced Settings (Optional)**

Disk Expiry Thread Interval	20 seconds
Disk Spool Buffer Size	50 megabytes
Clear on Flush	<input checked="" type="checkbox"/>
Copy on Read	<input type="checkbox"/>
Copy on Write	<input type="checkbox"/>
Memory Store Eviction Policy	LRU (Least Recently Used)
Logging	<input type="checkbox"/>

**BigMemory**

Overflow to Off-heap	<input type="checkbox"/>
Maximum Off-heap	Format: <Integer>m/M/g/G/t/T>

[Save Changes](#)

**Distributed Cache Configuration**

Distributed	<input checked="" type="checkbox"/>
Consistency	strong
Maximum Entries in Cache	1
Enable High Availability	<input type="checkbox"/>
Timeout	0000 milliseconds
Timeout Behavior	exception
Immediate Timeout When Disconnected	<input checked="" type="checkbox"/>
Synchronous Writes	<input checked="" type="checkbox"/>

**Search Configuration (Optional)**

Searchable	<input checked="" type="checkbox"/>
Allow Automatic Indexing	<input checked="" type="checkbox"/>
Key	<input checked="" type="checkbox"/>
Value	<input checked="" type="checkbox"/>

**Search Attributes (Optional)**

Attribute	Extract Method	Extract By	Add/Remove
		Class/Expression Properties	+ X
	<input checked="" type="radio"/> Expression	<input type="radio"/> Class	

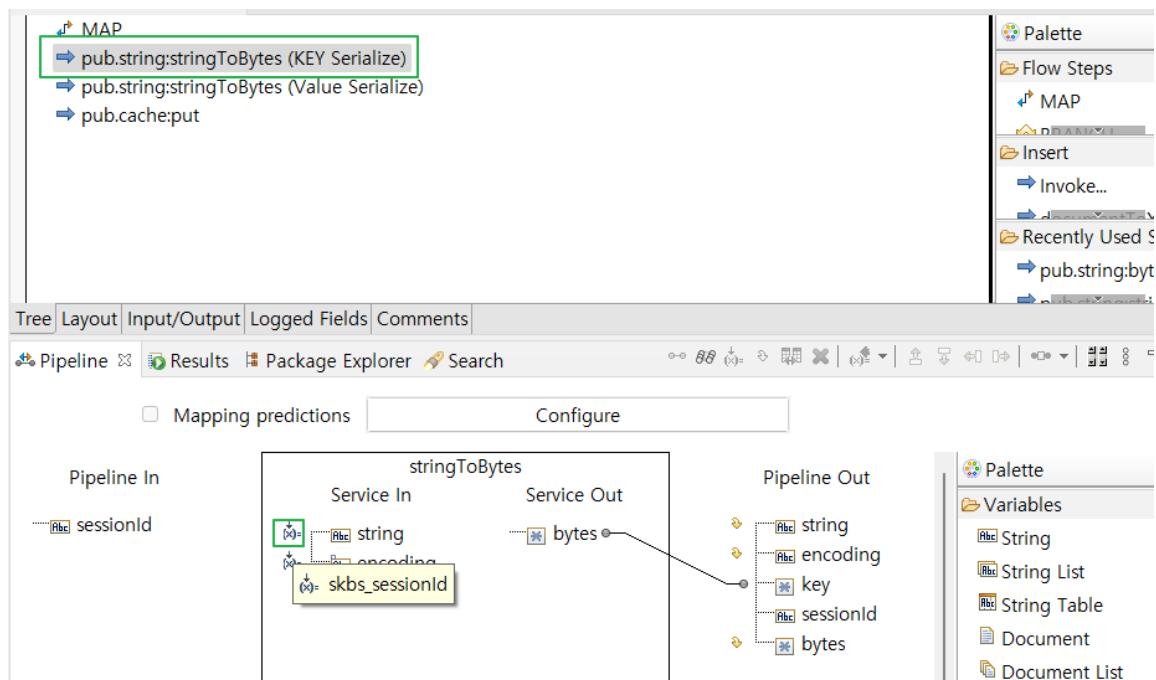
### 주요 설정 (참고 - Documentation)

- Maximum Elements in Memory > 저장할 총 요소의 수 > 0일 경우 제한이 없어 캐시가 무제한으로 증가하여 힙 메모리를 차지, 성능에 저하를 줄 수 있으므로 설정 주의
- Distributed Cache Configuration (분산 캐시 구성) > Distributed 선택
- Consistency (일관성) > Strong (즉시 동기화)

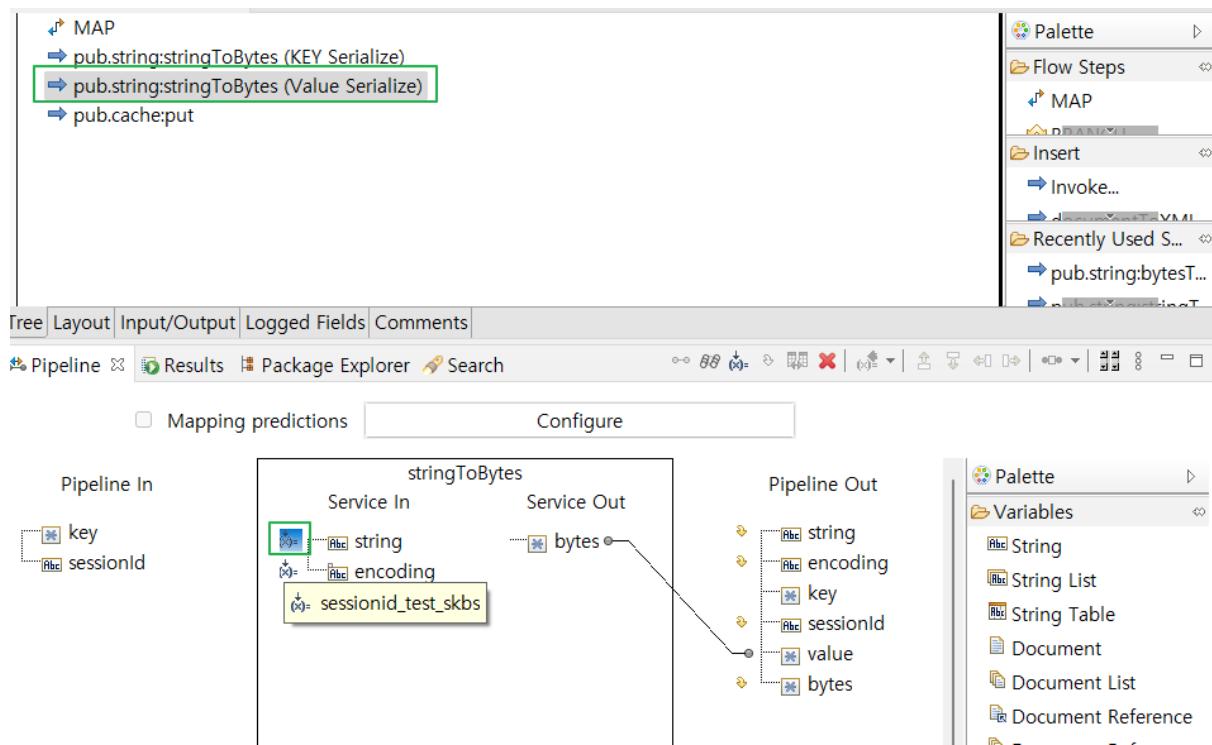
설정 완료 예시

- STEP 2. 서비스(pub.cache:) 생성 및 테스트

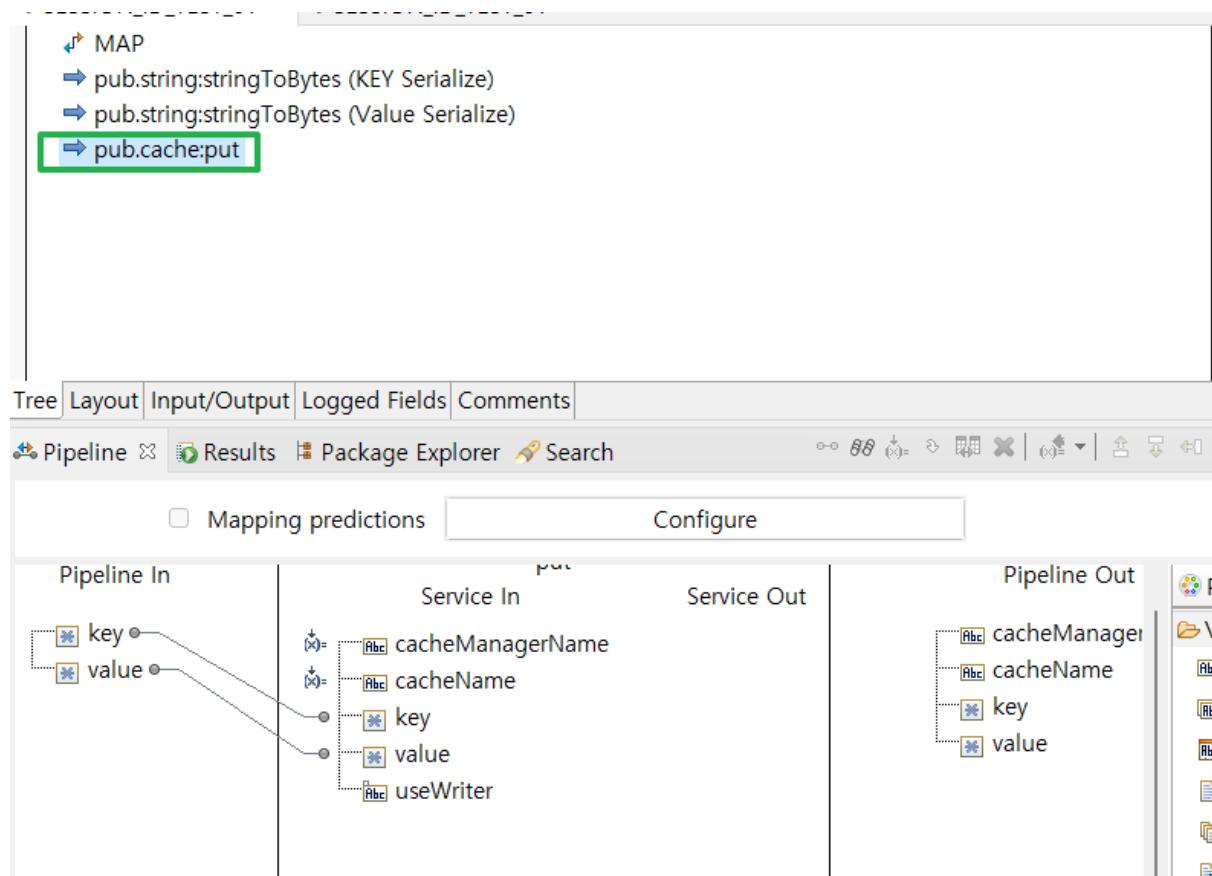
- 기 생성 된 캐시에 데이터(Key-Value) 저장 (pub.cache:put)



- key - "skbs\_sessionId" 값을 bytes로 변환



- value - "sessionId\_test\_skbs" 값을 bytes로 변환



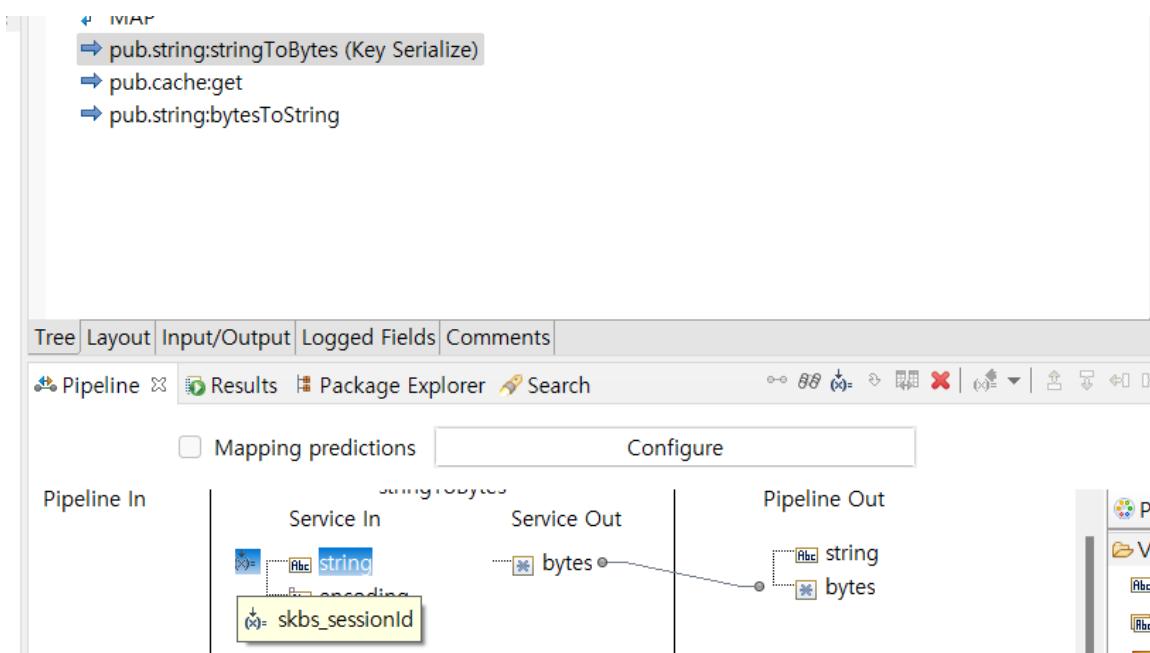
- cacheManagerName : 생성한 CacheManager의 명

- cacheName : Cache Manager 내부에 생성한 Cache 명
- 호출 결과

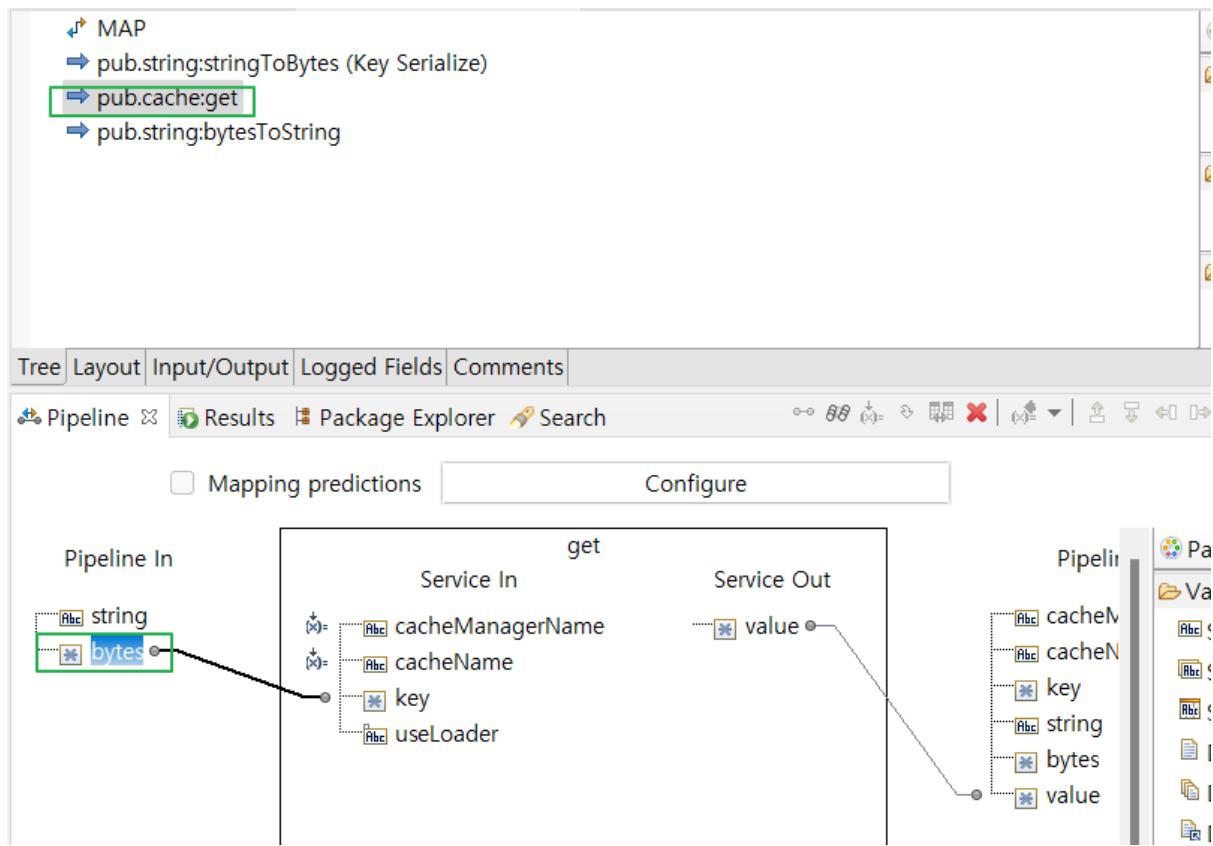
Package Explorer Search [192.168.1.168:5555] SKBS\_TEST.svc:SESSION\_ID\_TEST\_01 (2022. 10. 27. 오전 11:03:12)

Name	Value
cacheManagerName	SessionID_CacheManager
cacheName	sessionId
key	[B@7d943719
value	[B@7353c6db

- 캐싱 된 데이터 가져오기 (pub.cache:get)



- key - "skbs\_sessionId" 동일한 키값을 Serialize



- 변환 된 bytes 값을 pub.cache:get - key 값에 매핑
- Service Out 된 value(bytes) 를 bytes to string 으로 변환
- 호출 결과

Package Explorer Search

[192.168.1.167:5555] SKBS\_TEST.srvc:SESSION\_ID\_TEST\_01 (2022. 10. 27. 오전 11:05:42)

Name	Value
string	sessionid_test_skbs