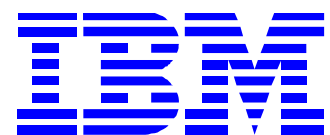


## **WebSphere eXtreme Scale v8.6**

설치 및 구성 가이드 ( HTTP 세션 관리 )

(2014. 10. )

IBM SWG WebSphere

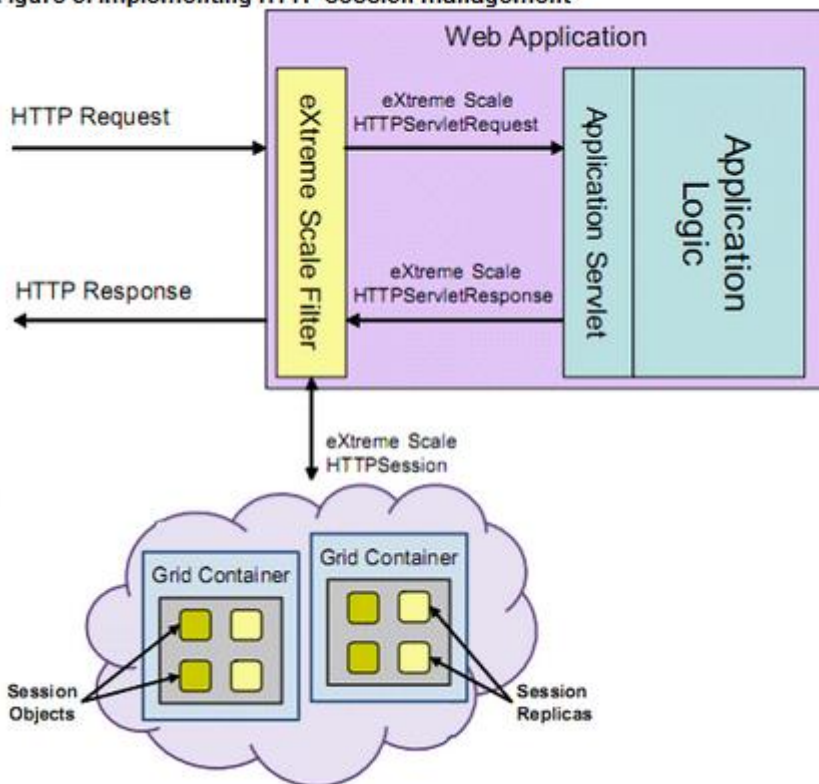


## 0) IBM WebSphere eXtreme Scale(WXS) 을 이용한 HTTP 세션 관리란?

IBM WebSphere eXtreme Scale 을 이용한 HTTP 세션 관리라는 것은 IBM WAS 가 HTTP 세션 데이터를 활용하는데 WAS 장애가 발생하고 fail-over 가 일어난다고 해도 이미 생성되어진 HTTP 세션 데이터를 잃어버리는 것이 아니라 그대로 유지할 수 있도록 IBM WXS 에서 HTTP 세션 데이터를 캐시 해두고 사용하는 구조를 의미합니다.(일반적으로 세션 클러스터라는 의미로도 불리워집니다.)

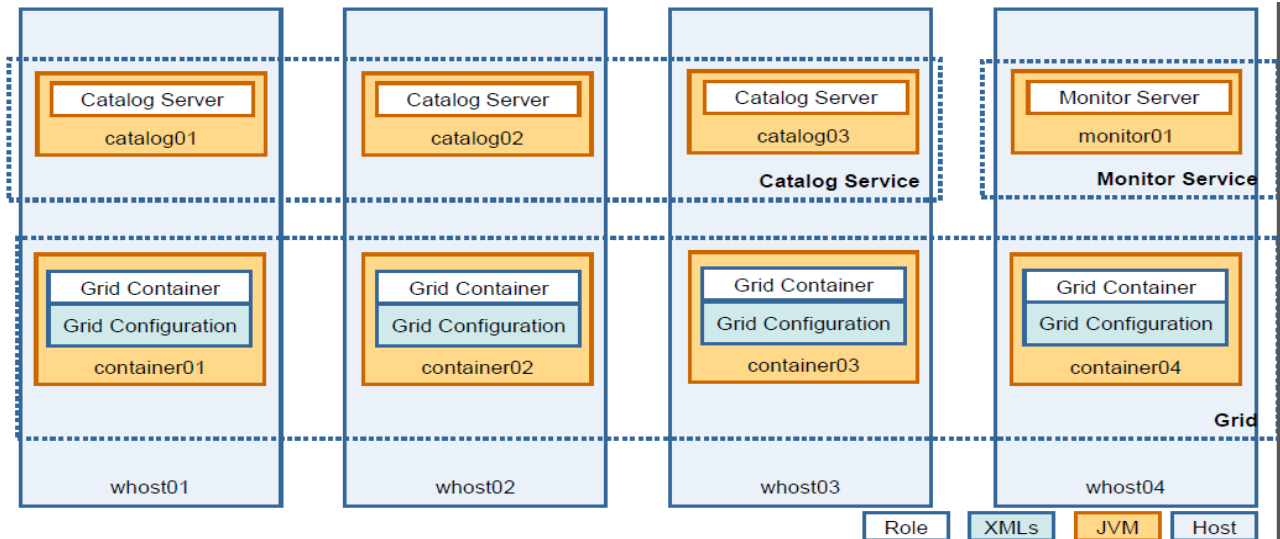
하단의 구조도를 보면 바로 이해할 수 있는 것처럼 IBM WAS 의 HTTP 세션 데이터를 유지하기 위해서 WXS 가 Filter 를 활용하여 해당 HTTP 세션 데이터를 Grid 컨테이너에 보관하고 관리하는 것을 의미합니다. (별도로 프로그래밍은 필요하지 않음)

**Figure 3. Implementing HTTP session management**

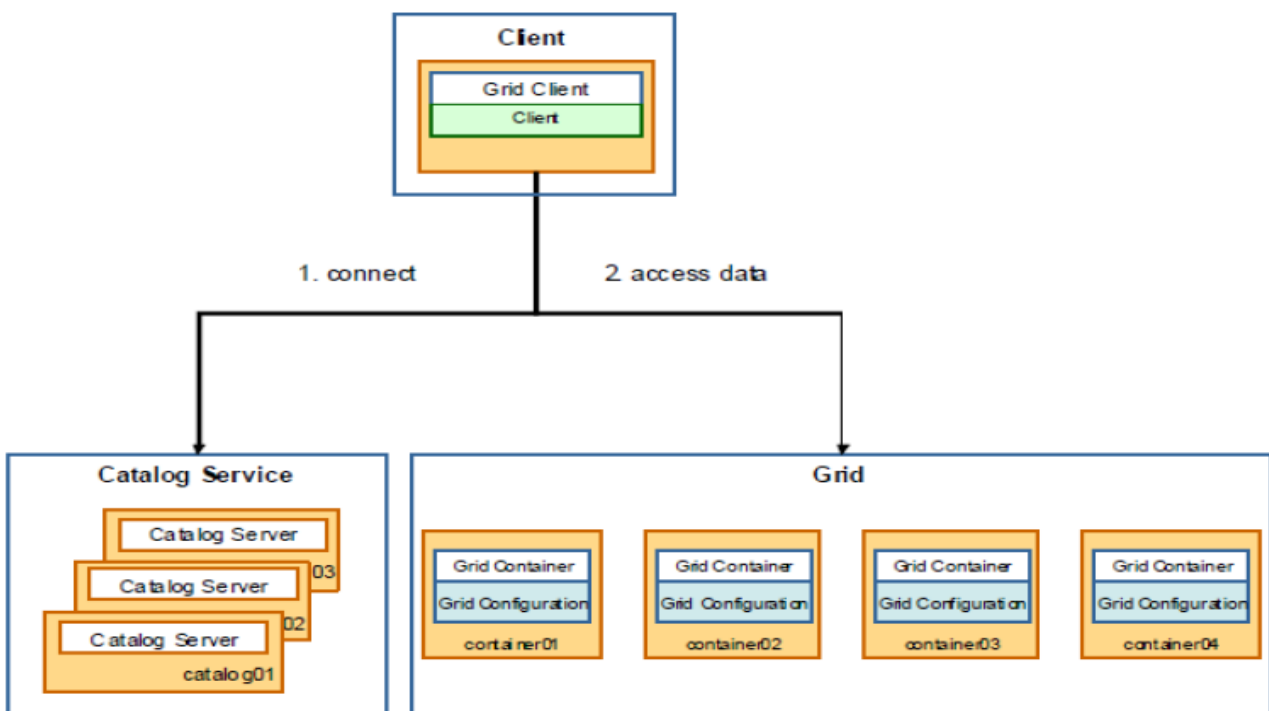


## 1) IBM WebSphere eXtreme Scale(WXS) 구조

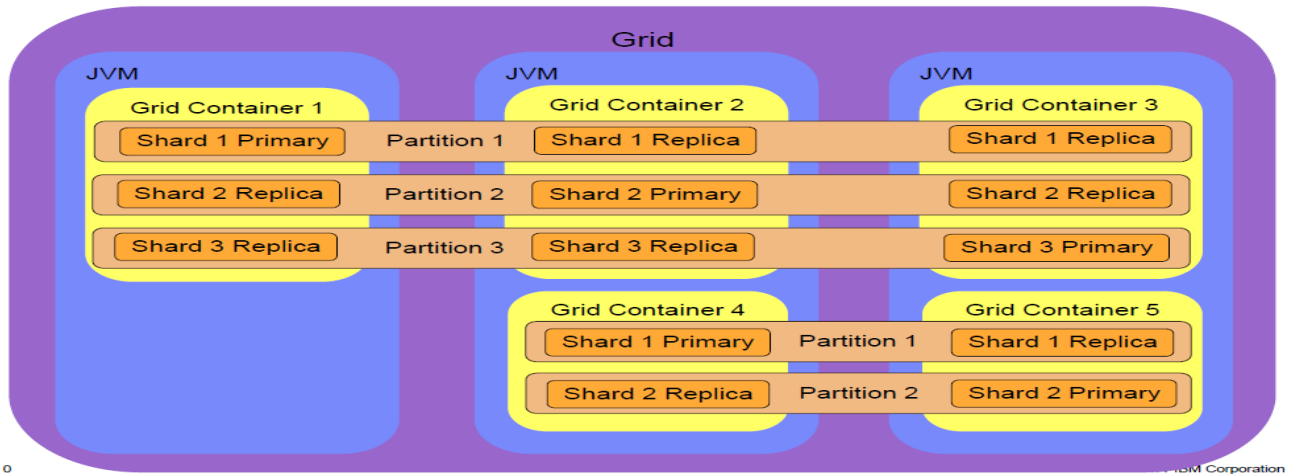
설치 및 구성을 살펴보기 전에 IBM WXS 구조를 먼저 이해하고 들어가도록 하겠습니다. WXS 는 하단과 같이 카탈로그 서버와 컨테이너 서버라는 두 가지 서버로 존재합니다.



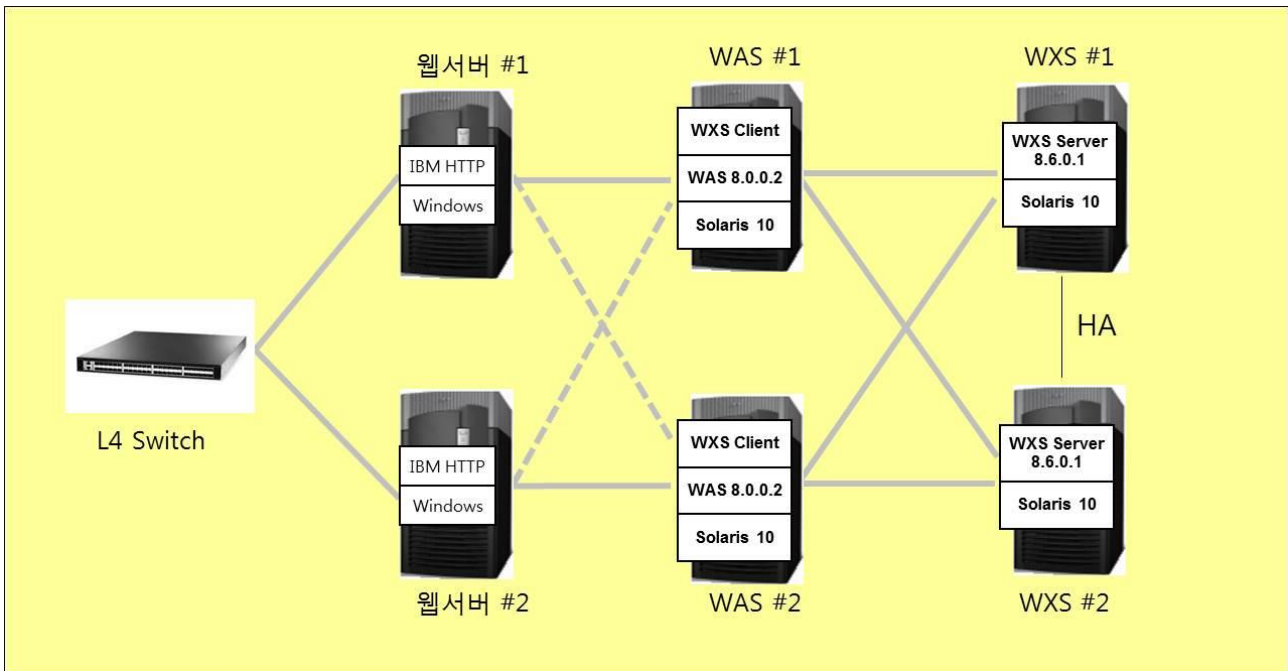
실제로 세션이 저장되는 저장 공간으로 사용되는 것은 컨테이너 서버이며 카탈로그 서버는 쉽게 설명하자면 머리 겸 인덱스 서버 역할을 수행합니다. 이를 다시 풀어서 말씀 드리면, 카탈로그 서버는 컨테이너 서버 제어 및 어떤 세션이 어느 컨테이너 서버에 있는지의 정보를 저장하는 역할을 수행합니다. 그래서 일반적으로 client 는 최초에 카탈로그 서버에 접속하여 인덱스 정보를 가지고 오고 이를 통해서 Grid 로 되어있는 컨테이너 서버에서 정확히 필요한 세션을 가지고 올 수 있는 것입니다.



조금 더 자세하게 컨테이너 서버의 내부를 들여다보면 하단과 같이 컨테이너는 파티션으로 구분되어 있고 파티션 안에는 원본 샤드와 그에 대응하는 복제본 샤드(원하는 n개를 설정할 수 있음)로 구성되어 있는 것을 확인할 수 있습니다. 그림을 보고 바로 이해하셨겠지만 이 샤드라는 곳이 실제 세션이나 캐쉬된 Data 가 들어있는 통으로 이해하시면 됩니다.



## 2) 샘플 토폴로지



상단의 그림에 있는 방식이 IBM WAS 와 WXS 를 사용했을 경우에 많이 보게 되는 일반적인 토폴로지 입니다. L4 가 HA 를 위해서 2대의 웹 서버를 바라보며 웹 서버는 plugin 을 통하여 active-active 방식으로 WAS 를 연결하고 WXS 도 이중화 구성으로 active-active 방식의 HA 를 구성합니다. 아직은 일반적으로 WAS 와 WXS 간의 하드웨어를 분리하는 경우보다 같은 하드웨어에서 구성 및 운영되는 경우가 더 많습니다. WXS 를 이용해서 단순 업무에 HTTP 세션 관리기능을 이용하신다면 언급한 것처럼 굳이 하드웨어를 분리하지 않아도 충분합니다. 그러나, WXS 가 여러 업무에 대한 HTTP 세션 용도이거나 다른 캐시영역으로도 많이 사용되는 경우라면 성능 및 운영 관리성, 비용 효율성을 따져보고 WAS 와 WXS 의 하드웨어를 분리하는 것을 추천 드립니다.

**참고 : WXS 8.6 InfoCenter - Tuning operating systems and network settings**

[http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Fwelcome%2Fwelcome\\_xs.html](http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Fwelcome%2Fwelcome_xs.html)

하드웨어 관련해서 적용하면 좋을 네트워크 설정

**AIX®:**

```
/usr/sbin/no -o tcp_sendspace=65536
/usr/sbin/no -o tcp_recvspace=65536
/usr/sbin/no -o udp_sendspace=65536
/usr/sbin/no -o udp_recvspace=65536
/usr/sbin/no -o somaxconn=10000
/usr/sbin/no -o tcp_nodelayack=1
/usr/sbin/no -o tcp_keepinit=40
/usr/sbin/no -o tcp_keepintvl=10
```

**Linux:**

```
sysctl -w net.ipv4.tcp_timestamps=0
sysctl -w net.ipv4.tcp_tw_reuse=1
sysctl -w net.ipv4.tcp_tw_recycle=1
sysctl -w net.ipv4.tcp_fin_timeout=30
sysctl -w net.ipv4.tcp_keepalive_time=1800
sysctl -w net.ipv4.tcp_rmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_wmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_max_syn_backlog=4096
```

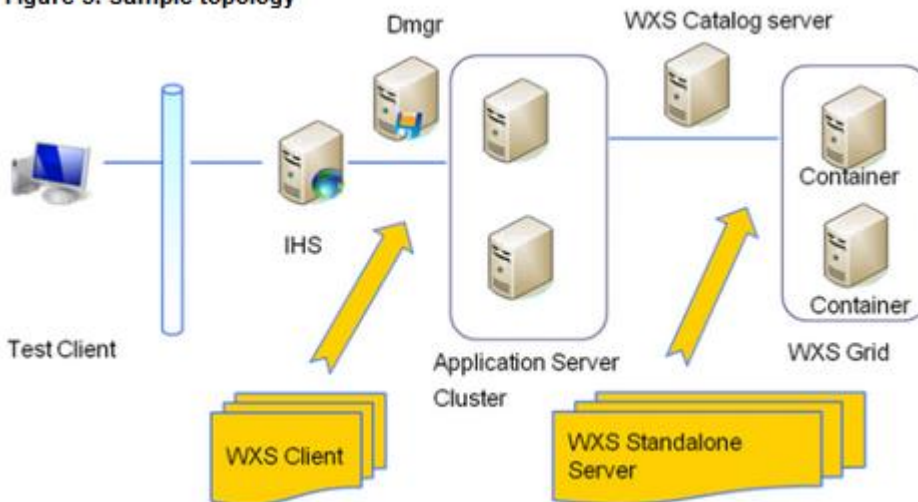
**HP-UX:**

```
ndd -set /dev/tcp tcp_ip_abort_cinterval 20000
```

### 3) IBM WebSphere eXtreme Scale 설치

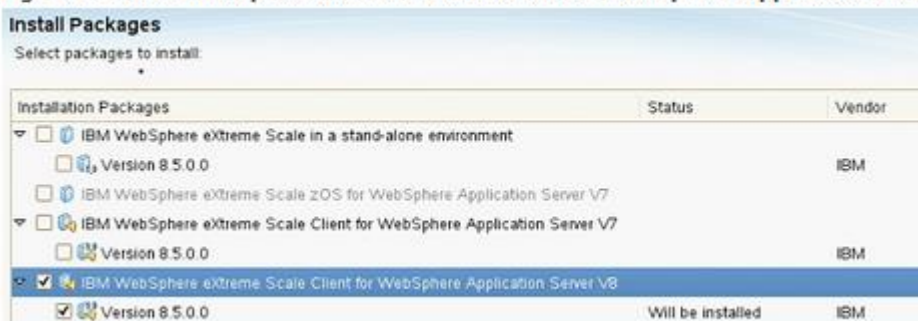
토폴로지가 확립되면 일반적으로 IBM WXS 설치를 어떤 방식으로 하는 것이 좋느냐는 물음이 많은데 대단위 서버를 대상으로 하지 않으면서 Http 세션 관리가 목적이라면 IBM WAS 위에 WXS 를 같이 올려서 프로파일을 강화하는 것 보다는 IBM WXS 단독(stand-alone) 형태로 설치하는 것이 향후 안정성 및 관리성에 더 나은 것으로 보여집니다.

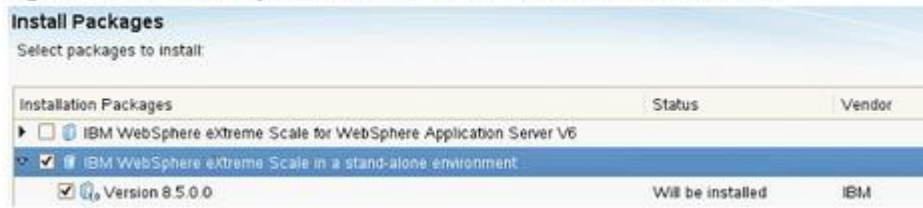
Figure 5. Sample topology



샘플 토폴로지의 경우라면 WAS 가 설치된 장비에 WAS 목적으로 설치한 프로파일을 다시금 WXS 로 강화를 하지 말고 WXS client 만 설치를 하고 이 후에 WXS Server 는 별도의 장비나 같은 장비의 다른 폴더에 설치하면 됩니다. 이렇게 단독 형태로 설치하게 되면 WAS 랑 생명주기가 완전히 분리되며 운영 효율성을 가지고 WXS 는 스크립트를 이용해서 손 쉽게 시작/중지와 같은 관리작업을 별도로 수행할 수 있습니다.

Figure 6. Select WebSphere eXtreme Scale client for WebSphere Application Server V8



**Figure 7. Select WebSphere eXtreme Scale stand-alone environment**

당연히 대단위 서버를 대상으로 한다면 이런 스크립트로 관리하는 것보다 관리콘솔을 이용한 관리의 편의성이 보다 더 큰 조건이 될 것이기 때문에 IBM WAS ND 를 사용하여 통합 관리하는 형태를 권고 드립니다.



## 4) IBM WebSphere eXtream Scale Server 구성

### ① IBM WXS 의 카탈로그 서버와 컨테이너 서버의 구성을 결정하는 설정 정보 파일 준비

일반적으로 설치되어 있는 [WXS 설치디렉토리]/ObjectGrid/session/samples 디렉토리에 설정 샘플파일이 들어있으며 해당 샘플파일을(objectGridDeploymentStandAlone.xml objectGridStandAlone.xml) 별도의 폴더(예:/wxspoc/xml) 로 복사해 두는 것이 좋습니다. 당연히 복사한 설정파일들은 구분을 위해서 원하는 형태로 이름을 변경해 둡니다.(예를 들어 PocObjectGrid.xml 과 PocDeployment.xml)

해당 설정 파일들이 IBM WXS 의 카탈로그 서버와 컨테이너 서버의 구성을 결정하는 설정 정보 파일이며 HTTP 세션 서버용으로 Best practice 가 적용된 설정 정보입니다. 다시 말해서, 언급하고자 하는 HTTP 세션 관리를 위한 목적이라면 변경없이 바로 사용할 수 있습니다. 해당 설정의 변경을 원하면 하단의 첨부된 세부 내용을 참고하시기 바랍니다.

참고 : WXS 8.6 InfoCenter - ObjectGrid descriptor XML file

<http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/rxsogref.html>

#### objectGrid.xml 샘플

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="session" txTimeout="30">
            <bean
                                id="ObjectGridEventListener"
className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
            <backingMap
                                name="objectgridSessionMetadata"
pluginCollectionRef="objectgridSessionMetadata"
                                readOnly="false"
                                lockStrategy="PESSIMISTIC"
ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
            <backingMap
name="objectgridSessionAttribute.*"
template="true"
readOnly="false"
lockStrategy="PESSIMISTIC"
ttlEvictorType="NONE"
copyMode="NO_COPY"/>
            <backingMap
name="objectgridSessionTTL.*"
template="true"
readOnly="false"
lockStrategy="PESSIMISTIC"
ttlEvictorType="LAST_ACCESS_TIME"
timeToLive="3600"
copyMode="NO_COPY"/>
        </objectGrid>
    </objectGrids>
    <backingMapPluginCollections>
        <backingMapPluginCollection id="objectgridSessionMetadata">
            <bean id="MapEventListener"
className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

```
</backingMapPluginCollections>
```

```
</objectGridConfig>
```

**txTimeout** : 트랜잭션 타임아웃 값, 트랜잭션을 완료하는데 해당 시간 보다 오래 걸린다면 TransactionTimeoutException 을 반환하고 해당 트랜잭션은 롤백 됩니다.

**readOnly** : backingMap 인스턴스 읽기/쓰기 모드 설정 값, true 로 변경하면 해당 backingMap 은 오직 읽기만 가능합니다.

**lockStrategy** : 트랜잭션에 의해 map 엔트리에 접근할 때 마다 internal lock manager 가 사용되며 3가지 전략을 제공합니다. - OPTIMISTIC, PESSIMISTIC, NONE

- Pessimistic : Data 가 자주 변하는 경우에 사용되며 cache 엔트리를 read 할 때 lock 을 얻어오고 조건부로 트랜잭션이 완료될 때까지 lock 을 유지합니다. Lock 을 유지하는 시간은 Transaction isolation level 튜닝을 통해 조절 가능합니다.
- Optimistic : pessimistic 에 비해서 성능이나 확장성이 더 향상된 locking 전략이며 애플리케이션이 때때로 발생하는 update 실패에 대해서 큰 이슈가 없다면 Optimistic 전략 보다 좋은 성능을 보장합니다. 주로 read 가 많으며 write 나 update 가 간헐적인 경우에 적합한 lock 전략입니다.
- None : internal lock manager 가 필요없을 경우에 사용되며 동시성 제어는 eXtreme Scale 외부에서 이루어져야 합니다..

**ttlEvictorType** : BackingMap 엔트리에 대한 만료 시간을 어떻게 계산할 것인지에 대한 옵션 값이며 CREATION\_TIME, LAST\_ACCESS\_TIME, LAST\_UPDATE\_TIME, NONE 중에 선택이 가능합니다.

**timeToLive** : 여기서 지정된 값을 기반으로 TTL evictor 가 map 엔트리를 메모리에서 퇴출시킵니다.

**copyMode** : get 명령시에 BackingMap 인스턴스가 실제 값을 어떻게 반환할지에 대한 옵션 값입니다.

**COPY\_ON\_READ\_AND\_COMMIT** : 애플리케이션이 BackingMap 인스턴스의 value object 에 대한 참조를 가지고 있지 않는 것을 보장합니다. 대신에, 애플리케이션은 항상 BackingMap instance 값의 복제본을 가지고 작업합니다.

- COPY\_ON\_READ : 트랜잭션이 완료되었을 때 발생하는 복제본을 제거함으로써 COPY\_ON\_READ\_AND\_COMMIT 보다 성능이 더 좋습니다. BackingMap data 의 무결성을 보존하기 위하여 애플리케이션은 트랜잭션이 완료된 후 엔트리에 대한 모든 참조를 지우는 작업을 수행합니다. 해당 옵션을 사용하면 ObjectMap.get 메소드 수행시 트랜잭션이 완료되기 전까지 애플리케이션의 변경 작업이 BackingMap 요소에 영향을 미치지 못하도록 value 에 대한 참조가 아니라 복사본을 반환합니다.
- COPY\_ON\_WRITE : ObjectMap.get 메소드 수행시 value object 에 대한 직접 참조 대신에 value 에 대한 proxy 를 반환합니다. Proxy 는 애플리케이션에 의해서 value interface 의 set 메소드가 호출되지 않는다면 value 의 복제본을 만들지 않는 것을 보장합니다.
- NO\_COPY : 성능 향상을 위하여 ObjectMap.get 메소드를 통해 가지고 온 value object 가 절대 변경되지 않게 합니다.
- COPY\_TO\_BYTES : 복잡한 object 타입의 메모리 풋 프린트를 줄이고 복제본을 만들기 위하여 직렬화가 필요한 객체의 복제 성능을 향상하기 위해 사용됩니다.

참고 : WXS 8.6 InfoCenter - Deployment policy descriptor XML file

[http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/](http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/rxsdplcyref.html)

[com.ibm.websphere.extremescale.doc/rxsdplcyref.html](http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/rxsdplcyref.html)

### objectGridDeployment.xml 샘플

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
```

```
    xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
```

```
    <objectgridDeployment objectgridName="session">
```

```
        <mapSet name="sessionMapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="0"
```

```
maxAsyncReplicas="1" developmentMode="true" placementStrategy="PER_CONTAINER">
```

```
        <map ref="objectgridSessionMetadata"/>
```

```
        <map ref="objectgridSessionAttribute.*/>
```

```
        <map ref="objectgridSessionTTL.*/>
```

```
    </mapSet>
```

```
    </objectgridDeployment>
```

```
</deploymentPolicy>
```

**numberOfPartitions** : mapSet 요소의 파티션 개수 지정

**minSyncReplicas** : 각 파티션에 대한 동기식 복제본의 최소 개수 지정

**maxSyncReplicas** : 각 파티션에 대한 동기식 복제본의 최대 개수 지정

**maxAsyncReplicas** : 각 파티션에 대한 비동기식 복제본의 최대 개수 지정

**developmentMode** : 개발 모드 사용 여부에 대한 옵션(true/false), true 일 경우에는 하나의 컴퓨터에 원본 파티션과 복제본 파티션이 동시에 존재할 수 있음

**replicaReadEnabled** : read 요청에 대해서 primary 파티션 뿐만 아니라 복제본에서도 처리할 수 있도록 지정하는 옵션

**placementStrategy** : FIXED\_PARTITION - 가용한 컨테이너 서버들에 primary 샤드가 지정된 수치 만큼 배치됨, PER\_CONTAINER - 컨테이너 서버별로 지정된 수치 만큼 primary 샤드가 배치됨

## ② IBM WXS 의 카탈로그 서버와 컨테이너 서버의 구동시 참조하는 설정 정보 파일 준비

설치되어 있는 [WXS 설치디렉토리]/ObjectGrid/properties 디렉토리에 샘플파일이 들어있으며 해당 샘플 파일을(sampleServer.properties) 별도의 폴더(예:/wxspoc) 로 복사해 두는 것이 좋습니다. 당연히 복사한 설정파일들은 구분을 위해서 원하는 형태로 이름을 바꾸는데 이때 하나를 더 복제해두어서 카탈로그 서버용과 컨테이너 서버용을 구분해서 파일을 만들어 두는 것이 좋습니다.(예 : PocCatalog.properties 와 PocContainer.properties)

해당 파일은 이름 그래도 property 가 모여있는 설정 파일이며 IBM WXS 를 시작할 때 위치를 지정하여 해당 파일에 있는 property 설정을 적용하여 구동하는데 사용합니다.(예를 들어 포트 정보와 스레드 개수 정보와 같은 내용이 들어가 있습니다.)

해당 설정중에서 서로 살아있는지 heartbeat 를 교류하기 위한 haManagerPort, client 와의 서비스를 위해서 eXtremeIO (XIO) transport protocol 을 받는 listener 역할을 하는 listenerPort, JMX 서비스 정보를 얻기 위하여 호출할 수 있는 JMXServicePort 정보들을 기본적으로 수정해야 합니다. 예를 들어 하단과 같이 서로 port 가 충돌나지 않도록 설정해 주면 됩니다.

PocCatalog.properties 수정

haManagerPort=4900

listenerPort=4809

JMXServicePort=4099

PocContainer.properties 수정

haManagerPort=4910

listenerPort=4810

JMXServicePort=4100

참고 : WXS 8.6 InfoCenter - Server properties file

<http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2F>

[com.ibm.websphere.extremescale.doc%2Frxsogref.html](http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Frxsogref.html)

추가적으로 보면 도움이 될만한 설정

**minThreads, maxThreads** : evictor 나 오퍼레이션을 위한 런타임에 사용될 내부 thread pool 사이즈

**minXIOWorkerThreads, maxXIOWorkerThreads** : eXtremeIO 전송 요청 처리를 위한 thread pool 사이즈

**minXIONetworkThreads, maxXIONetworkThreads** : eXtremeIO 전송 네트워크를 위한 thread pool 사이즈

**xioTimeout** : eXtremeIO (XIO) transport 를 사용한 서버 요청의 타임아웃 값 (기본 - 30 초)

### ③ IBM WXS 의 카탈로그 서버와 컨테이너 서버의 구동을 위한 script 준비

WXS 를 단독 형태로 설치했을 경우 시작/중지등은 script 를 이용해서 진행해야 합니다. 이를 위해 하단과 같이 사전에 script 를 만들어두면 향후 보다 더 쉽게 관리작업이 가능합니다.

카탈로그 서버 시작을 위한 /wxspoc/bin/startCat.sh 작성

#### - WXS #1 서버 : PocWXS-1

```
[WXS설치경로]/ObjectGrid/bin/startXsServer.sh cat1 -listenerPort 4809 -catalogServiceEndPoints
cat1:PocWXS-1:6600:6601, cat2:PocWXS-2:6600:6601 -serverProps
/wxspoc/config/PocCatalog.properties -jvmArgs -Xms1024m -Xmx1024m -
javaagent:/opt/IBM/eXtremeScale/ObjectGrid/lib/wxssizeagent.jar
```

#### - WXS #2 서버 : PocWXS-2

```
[WXS설치경로]/ObjectGrid/bin/startXsServer.sh cat2 -listenerPort 4809 -catalogServiceEndPoints
cat2:PocWXS-2:6600:6601, cat1:PocWXS-1:6600:6601 -serverProps
/wxspoc/config/PocCatalog.properties -jvmArgs -Xms1024m -Xmx1024m -
javaagent:/opt/IBM/eXtremeScale/ObjectGrid/lib/wxssizeagent.jar
```

listenerPort : 카탈로그 서버가 리스닝하는 포트이며 해당 포트로 커뮤니케이션 합니다.

catalogServiceEndPoints : 카탈로그 서비스 리스트의 엔드포인트를 정의합니다. (하나 이상 가능)

- 카탈로그 서버 이름:호스트명:client port:peer port

- client port 와 peer port 는 2 대 이상의 카탈로그 서비스를 사용할시에 서로간의 내부 커뮤니케이션을 위해서 사용되는 포트입니다.

serverProps : 서버 프로퍼티 파일 위치(옵션)

jvmArgs : 추가적으로 필요한 JVM 사용자 정의값을 넣으실 수 있습니다.(옵션)

-javaagent:/opt/IBM/eXtremeScale/ObjectGrid/lib/wxssizeagent.jar : sizing agent 를 추가하는 옵션, JVM에 대한 사이징 예측의 정확도를 높이도록 추가적인 정보를 WXS 가 얻을 수 있도록 하는 옵션

컨테이너 서버 시작을 위한 `/wxspoc/bin/startContainer.sh` 작성

**- WXS #1 서버 : PocWXS-1**

```
[WXS설치경로]/ObjectGrid/bin/startXsServer.sh c1 -objectGridFile /wxspoc/xml/PocObjectGrid.xml -  
deploymentPolicyFile /wxspoc/xml/PocDeployment.xml -catalogServiceEndPoints PocWXS-1:4809,  
PocWXS-2:4809 -serverProps /wxspoc/session/PocContainer.properties -jvmArgs -Xms2048m -  
Xmx2048m
```

**- WXS #2 서버 : PocWXS-2**

```
[WXS설치경로]/ObjectGrid/bin/startXsServer.sh c2 -objectGridFile /wxspoc/xml/PocObjectGrid.xml -  
deploymentPolicyFile /wxspoc/xml/PocDeployment.xml -catalogServiceEndPoints PocWXS-2:4809,  
PocWXS-1:4809 -serverProps /wxspoc/session/PocContainer.properties -jvmArgs -Xms2048m -  
Xmx2048m
```

서버 종지를 위한 `/wxspoc/bin/stopServer.sh` 작성

**- WXS #1 서버 : PocWXS-1**

```
[WXS설치경로]/ObjectGrid/bin/stopXsServer.sh "$@" -catalogServiceEndPoints PocWXS-1:4809,  
PocWXS-2:4809
```

**- WXS #2 서버 : PocWXS-2**

```
[WXS설치경로]/ObjectGrid/bin/stopXsServer.sh "$@" -catalogServiceEndPoints PocWXS-2:4809,  
PocWXS-1:4809
```

이렇게 하셨다면 WXS Server 에 대한 준비가 완료된 것 입니다.

## 5) WebSphere eXtream Scale Client 구성 - WAS

Client 에서 WXS 를 사용하기 위해서는 WXS 이 제공하는 HTTP Session Manager 를 사용하기 위해 적용하고자 하는 애플리케이션의 web.xml 파일에 하단의 내용을 삽입합니다.

(이미 애플리케이션이 설치되어 있는 상태에서 직접 설정을 변경하려고 하면 web\_merged.xml 파일을 수정해야 합니다. 또는 WAR, EAR 로 만들어진 애플리케이션이 있다면 하단의 링크에 설명이 나온 것처럼 제공되는 스크립트와 splicer.properties 파일을 이용해서 WAR, EAR 에 동적으로 하단의 내용을 추가 가능합니다.

Splicing a session data grid application with the addObjectGridFilter script

<http://pic.dhe.ibm.com/infocenter/wxinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/txssplicescript.html?resultof=%22%53%70%6c%69%63%65%72%2e%70%72%6f%70%65%72%74%69%65%73%22%20>)

```
<context-param>
    <param-name>securityEnabled</param-name>
    <param-value>>false</param-value>
</context-param>
<context-param>
    <param-name>fragmentedSession</param-name>
    <param-value>true</param-value>
</context-param>
<context-param>
    <param-name>sessionTableSize</param-name>
    <param-value>1000</param-value>
</context-param>
<context-param>
    <param-name>objectGridType</param-name>
    <param-value>REMOTE</param-value>
</context-param>
<context-param>
    <param-name>replicationInterval</param-name>
    <param-value>10</param-value>
</context-param>
<context-param>
    <param-name>catalogHostPort</param-name>
    <param-value>PocWXS-1:3809,PocWXS-2:3809</param-value>
</context-param>
<context-param>
```



```
<param-name>objectGridName</param-name>
<param-value>session</param-value>
</context-param>
<filter>
  <description>Filter that provides for an ObjectGrid based Session Manager.</description>
  <display-name>HttpSessionFilter</display-name>
  <filter-name>HttpSessionFilter</filter-name>
  <filter-class>com.ibm.ws.xs.sessionmanager.HttpSessionFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>HttpSessionFilter</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
</filter-mapping>
<listener>
  <description>IBMSessionListener</description>
  <display-name>IBMSessionListener</display-name>
  <listener-class>com.ibm.ws.xs.sessionmanager.IBMHttpSessionListener</listener-class>
</listener>
```

- **catalogHostPort** = PocWXS-1:3809,PocWXS-2:3809(카탈로그 서비스가 수행되는 호스트 명과 port)  
카탈로그 서비스에 대한 위치를 의미

- **fragmentedSession** = true

true 인 경우는 session.setAttribute 가 호출될 때만 해당 attribute 가 복제되며 해당 값이 false 면 getSession 이 호출될 때마다 전체 session data 가 복제됩니다.

- **replicationInterval** = 10

Session 복제가 비동기식으로 10초 마다 수행됨. 값이 0인 경우 Servlet의 종료시점에 동기식으로 복제되기 때문에 성능저하 발생될 수 있습니다만 session 이 반드시 유지 되어야 하는 경우라면 0 설정도 가능합니다.

애플리케이션을 위한 Web.xml 설정이 완료되면 각 서버 별로 IBM WAS 의 Admin Console 에서 Application servers > server1 > Web container > Custom properties 로 이동하여 하단의 정보가 추가되어야 합니다.

**HttpSessionCloneId** = "구분 되는 이름" - Session Affinity 를 위한 구분되는 이름을 넣는 옵션 - 예를

들어 baseServer01 이라고 입력하면 됨

**HttpSessionIdReuse** = "true" - 두 대의 WAS 서버를 교차할 때도 SessionId 를 재사용하는 옵션

이렇게 하면 아주 간단하게 WXS client 준비가 완료된 것입니다.

## 6) 웹 서버를 위한 HTTP Server Plugin 구성

관리콘솔 또는 "GenPluginCfg -server.name server1 -output.file.name 파일명" 명령을 통해서 각 WAS 서버별로 Plugin-cfg.xml 을 생성합니다. 수작업을 통해서 각 서버별로 만들어진 plugin-cfg.xml 파일을 하나로 병합하는 것이 가능하며 하단의 script 를 통하게 되면 자동으로도 가능합니다.  
pluginCfgMerge.sh

병합된 plugin-cfg.xml 설정에 GetDWLMTable 옵션을 true 로 설정합니다.  
GetDWLMTable="true"

### **Recommended values for web server plug-in config**

<http://www-01.ibm.com/support/docview.wss?uid=swg21318463>

## 7) Server 시작 및 테스트

### 1. WXS 서버 시작 .

각 서버별로 /wxspoc/bin/startCatalog.sh 을 실행, 모든 Catalog Server 가 실행 완료됨을 확인.  
서버별로 /wxspoc/bin/startContainer.sh 을 실행. 모든 Container Server 가 실행 완료됨을 확인

서버정보 확인 : xscmd.sh -c showinfo -cep PocWXS-1:4809, PocWXS-2:4809

### 2. HTTP Server 시작

### 3. WAS/Application 시작

### 4. Application 실행하고 WXS 에 Session 이 저장됨을 확인.

Grid 사이즈 확인 - xscmd.sh -c showmapsizes -cep PocWXS-1:4809, PocWXS-2:4809

Grid 데이터 확인 - xscmd.sh -c findbykey -g session -m objectgridSessionMetafddata -fs ".\*" -rv -cep  
PocWXS-1:4809, PocWXS-2:4809

## 9) 참고 자료

Deep dive into WebSphere eXtreme Scale HTTP session management, Part 1: Understanding HTTP session management and how it works in WebSphere eXtreme Scale

[http://www.ibm.com/developerworks/websphere/techjournal/1301\\_ying/1301\\_ying.html](http://www.ibm.com/developerworks/websphere/techjournal/1301_ying/1301_ying.html)

IBM WebSphere eXtreme Scale Version 8.6 Information Center

[http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Fwelcome%2Fwelcome\\_xs.html](http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Fwelcome%2Fwelcome_xs.html)