

IRIS Getting Started

Mobigen, November 20, 2015

목차

1	지원 정보	4
1.1	문서 버전	4
1.2	IRIS 버전	4
1.3	사용 문의 및 기술 정보	4
2	처음 시작하기	5
2.1	Single Node IRIS 설치하기	5
2.1.1	Single Node IRIS의 시작	5
2.1.2	IRIS-Startup : Master Node	7
2.1.3	IRIS-Startup : Slave Node	8
2.1.4	IRIS 동작 확인	12
2.2	Multi Node IRIS 설치하기	13
2.2.1	Multi Node IRIS 설치를 위한 준비	14
2.2.2	IRIS 설치	17
2.3	Client Package 설치하기	30
2.3.1	Overview	30
2.3.2	Client Package Overview	31
2.3.3	Client Package Install	31
2.3.4	IRIS CLI 실행하기	32
2.4	How to use IRIS CLI	33

2.4.1	iplus	33
2.4.2	iplus 실행하기	33
2.4.3	쿼리로 데이터 베이스 사용하기	34

1 지원 정보

1.1 문서 버전

0.1

1.2 IRIS 버전

1.5.1

1.3 사용 문의 및 기술 정보

(주) 모비젠

본사

- 135-280 서울 강남구 대치동 967-3번지 KM빌딩 2층, (주) 모비젠
- T : 02 - 538 - 9360
- F : 02 - 538 - 9369

기술 연구소

- 135-280 서울 강남구 대치동 967-3번지 KM빌딩 5층, (주) 모비젠
- T : 02 - 538 - 9364
- F : 02 - 538 - 9368

모비젠 IRIS 기술 지원

- T : 02 - 538 - 9364
- M : iris@mobigen.com

2 처음 시작하기

본 챗터에서는 IRIS를 처음 접하는 사용자를 위해 간단하게 IRIS를 설치하는 방법과 IRIS에서 제공하는 Command Line Interface인 iplus의 사용방법에 대하여 설명합니다.

2.1 Single Node IRIS 설치하기

본 챗터에서는 IRIS를 가장 빠르게 경험하기 위한 싱글 노드 IRIS 설치 방법에 대해서 설명합니다.

- [VAGRANT](#)

VAGRANT는 가상 머신을 관리하는 툴로 자세한 사용법은 위의 링크에서 확인 할 수 있습니다. 이미 생성되어 있는 가상 머신을 이용하여 IRIS를 경험 할 수 있습니다.

주의 사항 :
본 챗터에서 설명하는 Single Node IRIS는 간단한 테스트를 위해 사용 될 수 있지만 성능이 충분하지 않습니다. 실제 데이터를 이용한 테스트를 위해서는 Multi Node IRIS 설치를 참고하여 가상 머신이 아닌 실제 머신에서 동작 시키기를 추천합니다.

2.1.1 Single Node IRIS의 시작

Single Node IRIS의 시작을 위해서는 먼저 파일이 하나 필요합니다.

- Vagrantfile

위의 파일은 가상 머신의 설정을 미리 저장해둔 파일로 Vagrant를 이용하여 가상 머신을 동작시킬 때 필요한 파일입니다. 이 파일은 아래 메일 주소로 연락하여 요청할 경우 제공하고 있습니다.

```
iris@mobigen.com
```

먼저 Vagrant 설치가 완료 되었고 위의 Vagrantfile을 입수하였다면 다음 단계로 진행이 가능합니다. 가장 먼저 수행할 작업은 Vagrant가 실행될 디렉토리를 만들고 가상 머신을 생성합니다.

```
[vagrant@vagrant ~]$ mkdir iris_single_node_test
[vagrant@vagrant ~]$ cp [Vagrantfile Location] iris_single_node_test
[vagrant@vagrant ~]$ cd iris_single_node_test
[vagrant@vagrant ~]$ vagrant up
Bringing machine 'SingleNodeIRIS' up with 'virtualbox' provider...
[SingleNodeIRIS] Box 'MobigenSingleNodeIRIS' was not found. Fetching box from specified
URL for the provider 'virtualbox'. Note that if the URL does not have
a box for this provider, you should interrupt Vagrant now and add
the box yourself. Otherwise Vagrant will attempt to download the
full box prior to discovering this error.
.
.
.
[SingleNodeIRIS] Setting hostname...
[SingleNodeIRIS] Mounting shared folders...
[SingleNodeIRIS] -- /vagrant
[vagrant@vagrant ~]$
```

위의 과정을 수행한 후 가상 머신에 접속하여 IRIS를 실행합니다. 아래 화면은 vagrant 명령어를 이용하여 가상 머신에 접속했을 때 나타나는 화면입니다. 처음 접속시 계정명은 vagrant, 호스트네임은 snIRIS 라면 정상입니다.

```
[vagrant@vagrant ~]$ vagrant ssh
Last login: Fri Apr 24 06:53:25 2015 from 10.0.2.2
Welcome to your Vagrant-built virtual machine.
[vagrant@snIRIS ~]$
```

2.1.2 IRIS-Startup : Master Node

IRIS 를 실행할 때 사용하는 IRIS-Startup 명령어가 있습니다. 이 명령어를 이용하여 마스터와 슬레이브 노드에 필요한 데몬을 실행하고 자동으로 환경을 구성합니다. Single Node IRIS의 경우 마스터 노드와 슬레이브 노드가 하나의 노드에 포함되어 있으므로 iris 계정과 iriss 계정을 분리하여 사용합니다. 먼저 마스터 노드의 데몬들을 동작 시키기 위해서 아래와 같이 진행합니다.

```
[vagrant@snIRIS ~]$ su - iris
암호:iris
[iris@snIRIS ~]$ ~/IRIS/bin/Admin/IRIS-Startup
! NODE TYPE is MASTER
! COMMAND TYPE is START
! MPS DAEMON PORT is 5998
! RUN RECOVER is True
Do you want continue? (y/N) : y
!!! IRIS MASTER node start soon !!!

- mps daemon check
  - mps daemon is not running.
  - mps daemon start : True

- run all IRIS daemon.

- please check IRIS daemon.
-----
ABN  MID  NAME      DESC          GROUP      MODE  PID  CMD  STA  ...
-----
<begin>
OK | 17000 | NSD      | node selector | Master    | NML  | 1787 | ACT | ACT | ...
OK | 17050 | LISTENER | listener      | Master    | NML  | 1781 | ACT | ACT | ...
OK | 17200 | GTAD     | global table  | Master    | NML  | 1777 | ACT | ACT | ...
OK | 17210 | DLD     | data locator  | Master    | NML  | 1780 | ACT | ACT | ...
OK | 17432 | PGD     | priviledge grant | Master    | NML  | 1776 | ACT | ACT | ...
OK | 17478 | EHD     | environment   | Master    | NML  | 1793 | ACT | ACT | ...
```

```

OK | 17500 | BIM          | backend integrity | Master   | NML | 1788 | ACT | ACT | ...
  OK | 17990 | SMD          | system monitor   | Master   | NML | 1872 | ACT | ACT | ...
  OK | 97799 | DTD          | data transfer    | Common   | NML | 1792 | ACT | ACT | ...
OK | 97902 | SM           | status monitor   | Common   | NML | 1779 | ACT | ACT | ...
OK | 97999 | LogDaemon    | log daemon       | Common   | NML | 1791 | ACT | ACT | ...
<end>
[iris@snIRIS ~]$

```

iris 계정으로 로그인 한 후에 IRIS-Startup 명령어를 실행하면 먼저 사용자에게 진행 할 것인지 확인 하는 과정을 거칩니다. y를 입력하여 계속 진행하면 아래와 같은 화면이 나오면서 데몬이 실행됩니다.

2.1.3 IRIS-Startup : Slave Node

다음은 슬레이브 노드에서 사용할 데몬을 띄우는 순서입니다. 이번에는 마스터 노드 때와 달리 iriss 계정으로 접속합니다.

```

[vagrant@snIRIS ~]$ su - iriss
암호: iriss
[iriss@snIRIS ~]$ ~/IRIS/bin/Admin/IRIS-Startup
! NODE TYPE is SLAVE
! COMMAND TYPE is START
! MPS DAEMON PORT is 15998
! RUN RECOVER is True
Do you want continue? (y/N) : y
!!! IRIS SLAVE node start soon !!!

- mps daemon check
  - mps daemon is not running.
  - mps daemon start : True

- ramfs check
  - ramfs check : /home/iriss/IRIS/data/master
    - directory check : OK

```



```

- ramfs is not mounted.
[sudo] password for iriss:iriss
- ramfs mount : OK
- ramfs wrong owner.
[sudo] password for iriss:iriss
- owner change : OK
- ramfs check : /home/iriss/IRIS/data/slave
- directory check : OK
- ramfs is not mounted.
[sudo] password for iriss:iriss
- ramfs mount : OK
- ramfs wrong owner.
[sudo] password for iriss:iriss
- owner change : OK

Sync SYS_NODE_INFO : OK
Sync SYS_TABLE_INFO : OK
Sync SYS_INDEX_INFO : OK
Sync SYS_RECYCLEBIN : OK
Sync SYS_PROC_INFO : OK
Sync SYS_SCHED_INFO : OK
Sync SYS_HELP : OK
Sync SYS_USERS : OK
Sync SYS_DENY_USERS : OK
Sync SYS_TABLE_PRIV : OK
Sync SYS_PROC_PRIV : OK
Sync SYS_PM_CONF : OK
- recover start
=== RECOVER OPTION =====
option HARD          : False
option ALLNODE       : False
option THREAD COUNT  : 2
option MASTER ACCOUNT : None
option USE-SCP       : False
option FAST          : False
=====
Sync SYS_NODE_INFO : OK
Sync SYS_TABLE_INFO : OK

```

```

Sync SYS_INDEX_INFO : OK
Sync SYS_RECYCLEBIN : OK
Sync SYS_PROC_INFO : OK
Sync SYS_SCHED_INFO : OK
Sync SYS_HELP : OK
Sync SYS_USERS : OK
Sync SYS_DENY_USERS : OK
Sync SYS_TABLE_PRIV : OK
Sync SYS_PROC_PRIV : OK
Sync SYS_PM_CONF : OK
Target Table List : ['SYS_CPU_INFO', 'SYS_DISK_INFO', 'SYS_ERROR_LOG_INFO',
                    'SYS_NETWORK_INFO', 'SYS_OS_INFO', 'SYS_RAM_INFO', 'SYS_SESSION_DATA',
                    'SYS_TABLE_SIZE_INFO']

Estimated Total Time : 00:00:00

Table Recovering Start.

Nothing to Recover : SYS_CPU_INFO
Nothing to Recover : SYS_DISK_INFO
SYS_ERROR_LOG_INFO Start
[=====] 100%          2\2 R 00:00:01
SYS_ERROR_LOG_INFO End

Estimated Remain Time : 00:00:00
Nothing to Recover : SYS_NETWORK_INFO
Nothing to Recover : SYS_OS_INFO
Nothing to Recover : SYS_RAM_INFO
Nothing to Recover : SYS_SESSION_DATA
Nothing to Recover : SYS_TABLE_SIZE_INFO
=== SUMMARY =====
File Count : Total 2, Skip 0, Success 0, Fail 0, Delete 0
Run Time   : 00:00:02
Start Time : Tue Apr 28 00:40:36 2015
End Time   : Tue Apr 28 00:40:38 2015
=====
=== RECOVER OPTION =====
option HARD           : False
option ALLNODE        : False
option THREAD COUNT   : 2
option MASTER ACCOUNT : None

```

```

option USE-SCP      : False
option FAST        : False
=====
Recover is DONE.
- run all IRIS daemon.

- please check IRIS daemon.
-----
ABN  MID  NAME      DESC              GROUP  MODE  PID  CMD  STA  ...
-----
<begin>
OK | 27100 | UDM      | userquery dist. | Slave | NML | 27370 | ACT | ACT | ...
OK | 27110 | DSD      | data selector   | Slave | NML | 27357 | ACT | ACT | ...
OK | 27120 | DCD      | data changer    | Slave | NML | 27366 | ACT | ACT | ...
OK | 27610 | LSM      | local session   | Slave | NML | 27365 | ACT | ACT | ...
OK | 27903 | PL       | partition locator | Slave | NML | 27372 | ACT | ACT | ...
OK | 27904 | PR       | partition remover | Slave | NML | 27363 | ACT | ACT | ...
  OK | 27799 | DTD      | data transfer   | Common | NML | 27431 | ACT | ACT | ...
OK | 97902 | SM       | status monitor  | Common | NML | 27368 | ACT | ACT | ...
OK | 97999 | LogDaemon | log daemon     | Common | NML | 27361 | ACT | ACT | ...
<end>

```

마스터 노드의 경우에는 Startup 시에 sudo 권한을 사용하지 않기 때문에 패스워드 입력 화면이 없으나 슬레이브 노드에서는 sudo 권한을 사용하기 때문에 중간 중간에 정확한 패스워드를 입력해야 합니다. sudo 권한을 사용하는 이유는 다음과 같습니다.

- IRIS Master Data ramfs Mount
- IRIS Slave Data ramfs Mount
- ramfs 사용자 권한 설정

위의 과정을 거치고 나면 IRIS 는 실행 상태가 됩니다. 다음 챕터에서는 IRIS 가 정상 동작하는지 확인하는 방법에 대해서 설명합니다.

2.1.4 IRIS 동작 확인

IRIS 정상 동작 확인을 위해서는 아래와 같은 명령어를 사용합니다.

- ~/IRIS/bin/ntop
- ~/IRIS/bin/mps

가장 먼저 확인이 가능한 방법은 mps 명령어를 확인하여 데몬이 정상적으로 떠 있는지 확인하는 과정입니다.

2.1.4.1 mps 데몬 mps 데몬은 IRIS 동작에 필요한 데몬을 관리하는 데몬입니다. mps 데몬은 설정 파일을 참고하여 자기가 실행시킬 데몬을 실행 하거나 종료 시킬 수 있으며 아래와 같은 명령어를 받아 데몬 동작을 결정합니다.

mps 명령어	설명
mps	현재 동작 중인 데몬 리스트 출력
mps act [mid] [mid] ...	mid에 해당하는 데몬들 실행
mps term [mid] [mid] ...	mid에 해당하는 데몬을 종료
mps act [group_name]	그룹 이름에 해당하는 데몬들을 실행
mps term [group_name]	그룹 이름에 해당하는 데몬들을 종료

mps 명령어에 옵션이 없는 경우 아래와 같은 화면을 출력합니다.

```
[iris@snIRIS ~]$ mps
-----
ABN  MID  NAME      DESC          GROUP      MODE  PID  CMD  STA  ...
-----
<begin>
OK | 17000 | NSD      | node selector | Master    | NML  | 1787 | ACT | ACT | ...
OK | 17050 | LISTENER | listener      | Master    | NML  | 1781 | ACT | ACT | ...
OK | 17200 | GTAD     | global table  | Master    | NML  | 1777 | ACT | ACT | ...
OK | 17210 | DLD     | data locator  | Master    | NML  | 1780 | ACT | ACT | ...
OK | 17432 | PGD     | priviledge grant | Master    | NML  | 1776 | ACT | ACT | ...
```

```

OK | 17478 | EHD          | environment      | Master | NML | 1793 | ACT | ACT | ...
OK | 17500 | BIM          | backend integrity | Master | NML | 1788 | ACT | ACT | ...
  OK | 17990 | SMD          | system monitor   | Master | NML | 1872 | ACT | ACT | ...
  OK | 97799 | DTD          | data transfer    | Common | NML | 1792 | ACT | ACT | ...
OK | 97902 | SM           | status monitor   | Common | NML | 1779 | ACT | ACT | ...
OK | 97999 | LogDaemon   | log daemon       | Common | NML | 1791 | ACT | ACT | ...
<end>
[iris@snIRIS ~]$

```

위에서 정상동작 확인을 위해 확인해야 하는 부분은 ABN 항목입니다. 해당 부분이 ABN, 혹은 *** 이라면 정상 동작하는 상황이 아니니 점검이 필요합니다.

2.1.4.2 ntop ntop은 IRIS가 동작하는 모든 노드의 상태와 시스템 정보를 출력하는 유틸리티입니다. ntop은 10초 간격으로 새로 고침되며, 신규 시스템 정보를 출력합니다. 아래는 ntop의 화면입니다.

```

[iris@snIRIS ~]$ ntop
  NODE_ID,  SYS_STATUS,  ADM_STATUS,  UPDATE_TIME,  IP      , ...
-----
NODE:      0,          VALID,       ENABLE,       20150428052019,  127.0.0.1, ...
NODE:      1,          VALID,       ENABLE,       20150428052017,  127.0.0.2, ...

```

위의 화면에서 살펴야 할 부분은 바로 SYS_STATUS 부분입니다. 현재 상태를 의미하는 것으로 WAIT_RETRY, BUSY, READY 상태의 경우 시간이 지나면 정상 동작하지만 현재는 사용 불가라는 의미이며, INVALID의 경우 정상 동작 하지 않는다는 의미입니다. VALID는 정상 동작을 의미합니다.

2.2 Multi Node IRIS 설치하기

Single Node IRIS로는 성능에 대한 제약 사항이 많기 때문에 기본적으로 IRIS는 Multi Node 를 추천하고 있습니다. 본 챕터에서는 Multi Node IRIS

설치 방법에 대해서 자세하게 설명합니다.

2.2.1 Multi Node IRIS 설치를 위한 준비

다수의 노드에 IRIS를 설치하기 위해서는 사전 준비가 몇가지 필요합니다.

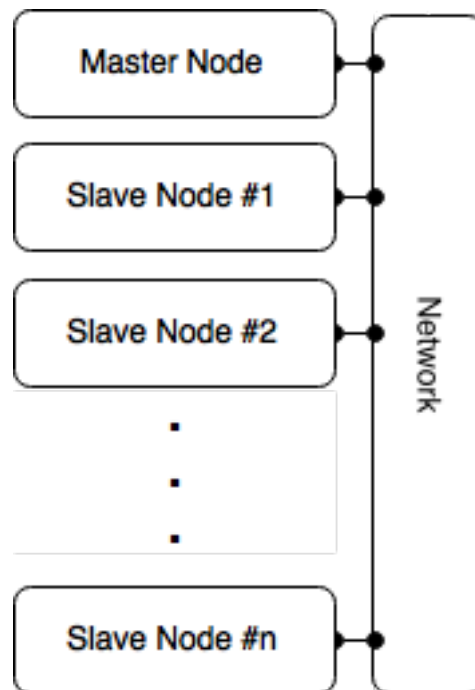


그림 1: Multi Node Default Architecture

2.2.1.1 다수 개의 서버 가장 먼저 이야기 해야 할 것은 바로 IRIS의 구조입니다. IRIS의 구조는 하나의 마스터 노드, 하나 이상의 슬레이브 노드로 구성이 됩니다. 즉, 두 대 이상의 독립된 서버가 필요합니다.

2.2.1.2 서버의 운영 체제 IRIS는 기본적으로 Redhat 기반의 운영체제를 지원합니다. 가장 많이 사용되는 운영체제는 CentOS 6.4(final) 버전이 가장 많이 사용되며 사용가능한 운영 체제의 리스트는 아래와 같습니다.

OS Name and Version	Work
CentOS 5.x	Fully Work
CentOS 6.x	Fully Work
RHEL 5	Fully Work
Ubuntu 14.04	Not Fully Work

* 주의 사항
기본적으로 IRIS Redhat 기반의 운영체제만 지원을 하며
Ubuntu 14.04의 경우 기본적인 동작 확인만 되어있습니다.

2.2.1.3 서버 설정 서버 설정은 아래와 같은 환경이 필요합니다.

- 계정
- 방화벽
- sudoer
- limit

2.2.1.3.1 계정 IRIS는 root 계정으로 실행 될 수 없습니다. 이를 위해서 iris 계정 사용을 추천하고 있습니다. 아래의 내용은 마스터 노드로 사용될 노드에 iris 계정을 추가하는 작업입니다.

```
[root@master ~]# adduser iris
[root@master ~]# passwd iris
```

모든 노드에서 위와 동일한 작업을 수행합니다.

* 주의 사항
모든 IRIS 데몬 (마스터, 슬레이브 모두) 은 동일한 계정 명에서 실행되어야 합니다.

2.2.1.3.2 방화벽 IRIS는 분산된 노드간에 다양한 통신을 소켓 통신으로 수행합니다. 이러한 이유로 다수의 네트워크 포트가 필요하게 되고 방화벽에서 해당 포트를 열어주는 작업이 필요합니다. 포트는 다음과 같습니다.

Master Daemon	Master Port	Slave Daemon	Slave Port
NSD	5000:tcp	UDM	5100:tcp
LISTENER	5050:tcp	DSD	5110:tcp
GTAD	5200:tcp	DCD	5120:tcp
DLD	5210:tcp	LSM	5610:tcp
DTD	5995:tcp	DTD	6995:tcp
PGD	5432:tcp	-	-
EHD	5678:tcp	-	-
BIM	5500:tcp	-	-
SMD	5996:tcp	-	-
CMDD	5800:tcp	CMDD	5800:tcp
SM	5902:tcp	SM	5902:tcp
LogDaemon	5999:tcp	LogDaemon	6999:tcp
mps	5998:tcp	mps	6998:tcp
HA	5997:tcp	-	-

마스터 노드의 경우 총 12개의 포트, 슬레이브 노드의 경우 총 8개의 포트 오픈이 필요합니다.

* 마스터에서 최하위 항목에 위치한 HA의 경우
HA 는 마스터 이중화와 관련된 포트입니다. 이에 대해서는 High Availability 항목을 참조해주세요.

2.2.1.3.3 SUDOER IRIS가 동작하기 위해서는 내부적으로 root 권한이 필요한 경우가 있습니다. 중요하게 필요한 경우는 다음과 같습니다.

- IRIS 슬레이브 노드 시작시
- IRIS 마스터 노드 이중화 동작시

위와 같은 경우를 위해서 sudoer에 IRIS 계정을 추가해줍니다. 본 설명에서는 이전 계정 설명에서 추천한 iris 라는 계정을 기준으로 설명합니다.


```
[root@master ~]# vim /etc/sudoers
-----
.
.
.
root    ALL=(ALL)  ALL
iris    ALL=(ALL)  ALL
```

위와 같이 수정한 후에 저장 후 종료합니다.

2.2.1.3.4 LIMIT IRIS에서는 다수의 프로세스가 동작하고 많은 파일에 동시에 접근하기 때문에 리눅스에서 기본 설정되어 있는 LIMIT값으로는 성능을 제대로 발휘하기 힘듭니다. 이를 해결하기 위해서 `limits.conf`를 수정합니다. 아래 내용을 추가해야 합니다.

```
[root@master ~]# vim /etc/security/limits.conf
-----
.
.
.
iris    soft    nproc   65536
iris    hard    nproc   65536
iris    soft    nofile  65536
iris    hard    nofile  65536
```

2.2.2 IRIS 설치

IRIS 설치는 마스터 노드 설치, 슬레이브 노드 설치로 구성됩니다. IRIS 설치를 하기 위해서는 사전에 마스터 노드와 슬레이브 노드의 IP 리스트가 필요합니다. 본 문서에서 사용한 노드의 IP는 다음과 같습니다.

NodeType	IP Address
Master Node	192.168.111.100
Slave Node #1	192.168.111.201
Slave Node #2	192.168.111.202

본 문서에서 사용할 노드의 디스크 정보는 다음과 같습니다.

Node Name	Disk Mount Location	Detail
Master Node	/	
Slave Node #1	/	
Slave Node #1	/DATA1	Data Disk 1
Slave Node #1	/DATA2	Data Disk 2
Slave Node #2	/	
Slave Node #2	/DATA1	Data Disk 1
Slave Node #2	/DATA2	Data Disk 2

마스터 노드는 하나의 디스크를 이용하여 IRIS를 구동합니다. 그에 반해 슬레이브 노드의 경우 주요 설정 파일 및 라이브러리는 / 디스크에 설치 되고 데이터는 /DATA1, /DATA2에 저장합니다.

2.2.2.1 설치 파일 얻기 IRIS 설치 파일은 IRIS 지원 연락처를 통해서 구할 수 있습니다. IRIS 설치 파일은 하나의 압축 파일로 구성되어 있으며 파일명은 다음과 같이 명시됩니다.

```
IRIS_[VERSION]_[OS_NAME]_[DATE].tgz
```

2.2.2.2 마스터 노드 설치

2.2.2.2.1 설치 파일 압축 해제 IRIS 설치 파일을 구한 후 해당 계정의 디렉토리에 복사한 후에 다음 명령어로 압축을 해제합니다.

```
[iris@master ~]$ ls
IRIS_1.5.0.b.0.tgz
[iris@master ~]$ tar xzvf IRIS_[VERSION]_[OS_NAME]_[DATE].tgz
.
.
.
[iris@master ~]$ ls
IRIS IRIS_1.5.0.b.0.tgz
[iris@master ~]$
```

압축을 해제하면 IRIS 폴더가 생성됩니다.

2.2.2.2.2 기초 IRIS 설치 설정 IRIS 설치 서문에서도 언급했지만 IP Address 지정해주는 설정작업이 필요합니다. IRIS/setup/pre-config/ 폴더에 파일을 만들어 주면 기초 설정은 종료됩니다. 필요한 파일은 두가지로 master, slaves 파일입니다. 이 두 파일은 모든 노드에 생성해야 합니다. 먼저 master 파일을 생성합니다.

```
[iris@master ~]$ cd ~/IRIS/setup/pre-config/
[iris@master pre-config]$ vi master
-----
0|192.168.111.100
```

master 파일은 위에 한줄을 작성해주면 됩니다. 포맷의 의미는 다음과 같습니다. 먼저 첫번째 0은 노드들이 가지는 ID입니다. 모든 노드는 정수 포맷의 ID를 가집니다. 마스터 노드의 경우 무조건 0으로 세팅하여야 합니다. 0 직후에 나오는 '|'는 분리자로 대문자 "I" 혹은 소문자 "i", 숫자 "1" 이 아니라 파이프기호 입니다. 다음 값으로 마스터 노드의 IP 주소가 들어갑니다. 다음은 slaves 파일을 생성할 차례입니다.

```
[iris@master ~]$ cd ~/IRIS/setup/pre-config/
[iris@master pre-config]$ vi slaves
-----
1|192.168.111.201
2|192.168.111.202
```

slaves 파일 역시 master 파일의 구성과 같습니다. master 파일의 경우 노드 0 하나만 입력하면 되지만 slaves는 모든 슬레이브 노드 정보를 리스트 형태로 작성해야 합니다. 노드 ID의 경우 0보다 큰 값은 모두 사용 가능합니다.

2.2.2.2.3 설치 스크립트 실행 기초 설치 설정이 완료되면 설치 스크립트를 실행합니다.

```
[iris@master ~]$ cd ~/IRIS/setup/
[iris@master setup]$ ./Install.sh
!! installation check start
  !! SUCCESS : 00.pre-config.sh : + ok
  !! SUCCESS : 10.user-config.sh : + ok
  !! SUCCESS : 20.ha-check.sh : ! HA FALSE
  !! SUCCESS : 30.master_ip.sh : ! MASTERIP 192.168.111.100
  !! SUCCESS : 40.slave_ip.sh : ! SLAVEIP 192.168.111.201
  !! SUCCESS : 50.my_ip.sh : ! MYIP 192.168.111.100
  !! SUCCESS : 51.my_mode.sh : ! MYMODE MASTER
  !! SUCCESS : 52.my_node_id.sh : ! MYNODEID 0
!! installation process start
  !! 00.env-install.sh process start
    !! remove old env
    !! create new env
    !! add to bash profile
  !! 10.user-limits.sh process start
    !! this script need sudoer
    !! you must enter your password for edit limits configuration
    !! security nproc limit setting
      !! limits nproc is already setted
    !! security nofile limit setting
```

```

    !! limits nofile is already setted
!! 20.create-script.sh process start
    !! create runUDMCheck
    !! create ramdisk_noti.sh
!! 30.create-directory.sh process start
!! 40.m6_config.sh process start
!! 50.help.sh process start
!! 60.mps_config.sh process start
!! 70.crontab.sh process start
!! 80.add_stickybit.sh process start
    !! this script need sudoer pass. please enter password

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for iris:
[iris@master setup]$

```

중간에 root 권한을 사용하는 부분이 있으므로 iris 계정의 비밀번호를 입력합니다. 설치 스크립트 완료 후 환경 변수 적용을 위해 로그 아웃한 후에 다시 로그인 합니다.

2.2.2.2.4 데이터 베이스 초기화 하기 설치 스크립트 완료 후 로그 아웃한 후 다시 로그인 했다면, 이로써 기본적인 라이브러리 및 IRIS 라이브러리 설치는 완료입니다. 하지만 현재 IRIS 마스터 노드는 외형만 준비 된 것이지 코어 부분은 데이터 베이스 초기화를 거쳐야 생성됩니다. 아래 명령을 수행합니다.

```

[iris@master ~]$ ~/IRIS/bin/Admin/DatabaseInit
Database Init Start.
Database Init Complete.
[iris@master ~]$

```

마스터 노드 설치과 완료 되었습니다. 슬레이브 노드를 설치하기 전에 데몬을 실행하고 정상 동작을 확인해야 합니다.

2.2.2.2.5 IRIS Startup IRIS 에서는 IRIS 데몬들을 실행하기 위해서 Startup 명령어를 제공한다. 본 챕터에서는 이를 이용하여 마스터 노드를 동작시키고 정상 동작하는지 확인합니다. 먼저 아래 명령어를 수행해서 IRIS 마스터 노드의 데몬을 실행합니다.

```
[iris@master ~]$ ~/IRIS/bin/Admin/IRIS-Startup
! NODE TYPE is MASTER
! COMMAND TYPE is START
! MPS DAEMON PORT is 5998
! RUN RECOVER is True
Do you want continue? (y/N) : y
!!! IRIS MASTER node start soon !!!

- mps daemon check
  - mps daemon is not running.
  - mps daemon start : True

- run all IRIS daemon.

- please check IRIS daemon.
-----
ABN  MID  NAME      DESC          GROUP      MODE  PID  CMD  STA  ...
-----
<begin>
OK | 17000 | NSD      | node selector | Master    | NML  | 2899 | ACT | ACT | ...
OK | 17050 | LISTENER | listener      | Master    | NML  | 2896 | ACT | ACT | ...
OK | 17200 | GTAD     | global table  | Master    | NML  | 2891 | ACT | ACT | ...
OK | 17210 | DLD      | data locator  | Master    | NML  | 2895 | ACT | ACT | ...
OK | 17432 | PGD      | priviledge grant | Master    | NML  | 2888 | ACT | ACT | ...
OK | 17478 | EHD      | environment   | Master    | NML  | 2908 | ACT | ACT | ...
OK | 17500 | BIM      | backend integrity | Master    | NML  | 2900 | ACT | ACT | ...
```

```

    OK | 17990 | SMD          | system monitor   | Master   | NML | 1872 | ACT | ACT | ...
    OK | 97799 | DTD          | data transfer    | Common   | NML | 1792 | ACT | ACT | ...
OK | 97800 | CMDD        | command daemon   | Common   | NML | 2890 | ACT | ACT | ...
OK | 97902 | SM          | status monitor   | Common   | NML | 2892 | ACT | ACT | ...
OK | 97999 | LogDaemon   | log daemon       | Common   | NML | 2903 | ACT | ACT | ...
<end>
[iris@master ~]$

```

마지막 결과 화면을 보면 현재 데몬들이 어떻게 동작하는지 확인할 수 있습니다. 다음 명령어를 이용하여 결과 화면을 다시 한번 살펴봐야 합니다.

```

[iris@master ~]$ mps
-----...
  ABN  MID  NAME          DESC                GROUP    MODE  PID  CMD  ...
-----...
<begin>
  OK | 17000 | NSD          | node selector     | Master  | NML | 2899 | ACT | ...
  OK | 17050 | LISTENER    | listener          | Master  | NML | 2896 | ACT | ...
  OK | 17200 | GTAD        | global table      | Master  | NML | 2891 | ACT | ...
  OK | 17210 | DLD         | data locator      | Master  | NML | 2895 | ACT | ...
  OK | 17432 | PGD         | privilege grant   | Master  | NML | 2888 | ACT | ...
  OK | 17478 | EHD         | environment       | Master  | NML | 2908 | ACT | ...
  OK | 17500 | BIM         | backend integrity | Master  | NML | 2900 | ACT | ...
    OK | 17990 | SMD          | system monitor    | Master  | NML | 1872 | ACT | ...
    OK | 97799 | DTD          | data transfer     | Common  | NML | 1792 | ACT | ...
  OK | 97800 | CMDD        | command daemon    | Common  | NML | 2890 | ACT | ...
  OK | 97902 | SM          | status monitor    | Common  | NML | 2892 | ACT | ...
  OK | 97999 | LogDaemon   | log daemon        | Common  | NML | 2903 | ACT | ...
<end>
[iris@master ~]$

```

mps 라는 명령어를 사용하면 현재 동작중인 데몬의 상태 리스트를 확인할 수 있습니다. 가장 첫번째 항목이 가장 중요한 것으로 세가지의 상태를 가지고 있습니다.

- OK

- ABN
- ***

먼저 OK는 “정상 실행중” 이라는 의미입니다. ABN은 “비정상 동작중” 의미입니다. 마지막으로 ***은 “정상 종료”를 의미합니다. 모두 OK라면 현재 모든 데몬이 정상적으로 동작중이라는 의미입니다. 모든 데몬이 정상적으로 동작하는 것을 확인하였다면 슬레이브 노드 설치를 진행합니다.

2.2.2.3 슬레이브 노드 설치

2.2.2.3.1 기본 설치 작업 기본적인 설치 작업은 마스터와 동일합니다. 본 챗터에서는 해당 내용을 간단하게 설명합니다.

```
[iris@slave1 ~]$ ls
IRIS_1.5.0.b.0.tgz
[iris@slave1 ~]$ tar xzvf IRIS_[VERSION]_[OS_NAME]_[DATE].tgz
.
.
.
[iris@slave1 ~]$ ls
IRIS IRIS_1.5.0.b.0.tgz
[iris@slave1 ~]$ cd ~/IRIS/setup/pre-config
[iris@slave1 pre-config]$ vi master
.
.
.
[iris@slave1 pre-config]$ vi slaves
.
.
.
[iris@slave1 pre-config]$ cd ~/IRIS/setup
[iris@slave1 setup]$ ./Install.sh
!! installation check start
!! SUCCESS : 00.pre-config.sh : + ok
```



```
!! SUCCESS : 10.user-config.sh : + ok
!! SUCCESS : 20.ha-check.sh : ! HA FALSE
!! SUCCESS : 30.master_ip.sh : ! MASTERIP 192.168.111.100
!! SUCCESS : 40.slave_ip.sh : ! SLAVEIP 192.168.111.201,192.168.111.202
!! SUCCESS : 50.my_ip.sh : ! MYIP 192.168.111.201
!! SUCCESS : 51.my_mode.sh : ! MYMODE SLAVES
!! SUCCESS : 52.my_node_id.sh : ! MYNODEID 1
!! installation process start
!! 00.env-install.sh process start
    !! remove old env
    !! create new env
    !! add to bash profile
!! 10.user-limits.sh process start
    !! this script need sudoer
    !! you must enter your password for edit limits configuration
    !! security nproc limit setting
        !! limits nproc is already setted
    !! security nofile limit setting
        !! limits nofile is already setted
!! 20.create-script.sh process start
    !! create runUDMCheck
    !! create ramdisk_noti.sh
!! 30.create-directory.sh process start
!! 40.m6_config.sh process start
!! 50.help.sh process start
!! 60.mps_config.sh process start
!! 70.crontab.sh process start
!! 80.add_stickybit.sh process start
    !! this script need sudoer pass. please enter password

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for iris:
[iris@slave1 setup]$ exit
```

설치 스크립트 실행 완료 후 다시 로그인합니다. 로그인 한 후에 다음 과정을 따릅니다. 마스터 노드와 다른 점은 데이터 베이스 초기화 과정은 생략된다는 점입니다.

2.2.2.3.2 슬레이브 노드 기본 설정 데이터 베이스 초기화 과정은 생략되지만 대신에 데이터 디스크로 사용할 디스크 링크 생성과정이 추가됩니다. 본 문서에서 구성되어 있는 설치 시스템의 경우 이전 시스템 설명에서와 같이 2개의 데이터 디스크를 따로 운영하고 있습니다. 해당 디스크에 IRIS에서 사용할 폴더를 만든 후에 권한 설정 후 링크를 추가합니다. 해당 내용은 아래와 같이 진행됩니다.

```
[iris@slave1 ~]$ sudo mkdir -p /DATA1/iris/data
[iris@slave1 ~]$ sudo chown -R iris:iris /DATA1/iris
[iris@slave1 ~]$ sudo mkdir -p /DATA2/iris/data
[iris@slave1 ~]$ sudo chown -R iris:iris /DATA2/iris
[iris@slave1 ~]$ cd ~/IRIS/data/slave_disk
[iris@slave1 slave_disk] ln -s /DATA1/iris/data part00
[iris@slave1 slave_disk] ln -s /DATA2/iris/data part01
```

2.2.2.3.3 IRIS Startup 위의 모든 작업이 완료 되면 Startup 명령을 수행합니다.

```
[iris@slave1 ~] ~/IRIS/bin/Admin/IRIS-Startup
! NODE TYPE is SLAVE
! COMMAND TYPE is START
! MPS DAEMON PORT is 5998
! RUN RECOVER is True
Do you want continue? (y/N) : y
!!! IRIS SLAVE node start soon !!!

- mps daemon check
- mps daemon is not running.
```

```
- mps daemon start : True

- ramfs check
- ramfs check : /home/iris/IRIS/data/master
  - directory check : OK
  - ramfs is not mounted.
[sudo] password for iris:
  - ramfs mount : OK
  - ramfs wrong owner.
[sudo] password for iris:
  - owner change : OK
- ramfs check : /home/iris/IRIS/data/slave
  - directory check : OK
  - ramfs is not mounted.
[sudo] password for iris:
  - ramfs mount : OK
  - ramfs wrong owner.
[sudo] password for iris:
  - owner change : OK

Sync SYS_NODE_INFO : OK
Sync SYS_TABLE_INFO : OK
Sync SYS_INDEX_INFO : OK
Sync SYS_RECYCLEBIN : OK
Sync SYS_PROC_INFO : OK
Sync SYS_SCHED_INFO : OK
Sync SYS_HELP : OK
Sync SYS_USERS : OK
Sync SYS_DENY_USERS : OK
Sync SYS_TABLE_PRIV : OK
Sync SYS_PROC_PRIV : OK
Sync SYS_PM_CONF : OK
- recover start
=== RECOVER OPTION =====
option HARD          : False
option ALLNODE       : False
option THREAD COUNT  : 2
```

```

option MASTER ACCOUNT : None
option USE-SCP          : False
option CLEAN            : True
option FAST             : False
=====
Sync SYS_NODE_INFO : OK
Sync SYS_TABLE_INFO : OK
Sync SYS_INDEX_INFO : OK
Sync SYS_RECYCLEBIN : OK
Sync SYS_PROC_INFO  : OK
Sync SYS_SCHED_INFO : OK
Sync SYS_HELP       : OK
Sync SYS_USERS      : OK
Sync SYS_DENY_USERS : OK
Sync SYS_TABLE_PRIV : OK
Sync SYS_PROC_PRIV  : OK
Sync SYS_PM_CONF    : OK
Target Table List : ['SYS_CPU_INFO', 'SYS_DISK_INFO', 'SYS_ERROR_LOG_INFO',
'SYS_NETWORK_INFO', 'SYS_OS_INFO', 'SYS_RAM_INFO', 'SYS_SESSION_DATA',
'SYS_TABLE_SIZE_INFO']

Estimated Total Time : 00:00:00

Table Recovering Start.

Nothing to Recover : SYS_CPU_INFO
Nothing to Recover : SYS_DISK_INFO
Nothing to Recover : SYS_ERROR_LOG_INFO
Nothing to Recover : SYS_NETWORK_INFO
Nothing to Recover : SYS_OS_INFO
Nothing to Recover : SYS_RAM_INFO
Nothing to Recover : SYS_SESSION_DATA
Nothing to Recover : SYS_TABLE_SIZE_INFO

=== SUMMARY =====
File Count : Total 0, Skip 0, Success 0, Fail 0, Delete 0,
             Found-D 0, Remain-D 0
Run Time   : 00:00:00
Start Time : Tue Jun  2 06:27:18 2015
End Time   : Tue Jun  2 06:27:19 2015
=====
=== RECOVER OPTION =====

```

```

option HARD          : False
option ALLNODE       : False
option THREAD COUNT  : 2
option MASTER ACCOUNT : None
option USE-SCP        : False
option CLEAN         : True
option FAST          : False
=====
Recover is DONE.
- run all IRIS daemon.

- please check IRIS daemon.
-----
ABN  MID  NAME      DESC          GROUP  MODE  PID  CMD  STA  ...
-----
<begin>
OK | 27100 | UDM      | userquery dist. | Slave | NML | 3249 | ACT | ACT | ...
OK | 27110 | DSD      | data selector   | Slave | NML | 3266 | ACT | ACT | ...
OK | 27120 | DCD      | data changer    | Slave | NML | 3253 | ACT | ACT | ...
OK | 27610 | LSM      | local session   | Slave | NML | 3257 | ACT | ACT | ...
OK | 27903 | PL       | partition locator | Slave | NML | 3256 | ACT | ACT | ...
OK | 27904 | PR       | partition remover | Slave | NML | 3251 | ACT | ACT | ...
OK | 27799 | DTD      | data transfer   | Common | NML | 3252 | ACT | ACT | ...
OK | 97800 | CMDD     | command daemon  | Common | NML | 3264 | ACT | ACT | ...
OK | 97902 | SM       | status monitor  | Common | NML | 3255 | ACT | ACT | ...
OK | 97999 | LogDaemon | log daemon      | Common | NML | 3258 | ACT | ACT | ...
<end>
[iris@slave1 ~]$

```

위의 작업이 완료 된 다음에 마스터 노드 설치에서와 같이 mps 명령을 이용하여 모든 데몬이 정상적인지 확인합니다.

2.2.2.3.4 노드 등록 모든 작업이 정상적으로 완료 되었다면 마지막 과정을 수행합니다. 마지막 과정은 슬레이브 노드를 마스터에 등록하는 과정으로 아래의 명령을 수행합니다.

```
[iris@slave1 ~]$ ~/IRIS/bin/Admin/NodeAdd
NODE is Added. (ID:1, IP:192.168.111.201)
[iris@slave1 ~]$
```

2.2.2.3.5 반복 작업 모든 슬레이브 노드에 동일한 작업을 수행합니다.

2.2.2.4 최종 확인 최종적으로 확인을 위해서 마스터 노드로 접속합니다. 마스터 노드로 접속해서 아래와 같은 명령을 내리면 주기적으로 노드 전체 상태를 확인할 수 있는 화면이 출력됩니다.

```
[iris@master ~]$ ntop

  NODE_ID,  SYS_STATUS,  ADM_STATUS,  UPDATE_TIME,  IP      , ...
=====
NODE:    0,      VALID,      ENABLE,  20150602065738, 192.168.111.100, ...
NODE:    1,      VALID,      ENABLE,  20150602065740, 192.168.111.201, ...
NODE:    2,      VALID,      ENABLE,  20150602065740, 192.168.111.202, ...
```

NODE ID 는 각각의 노드에 할당된 노드 ID를 의미합니다. 여기서 주의 깊게 봐야 하는 부분은 SYS_STATUS 부분입니다. 이 부분이 모두 VALID는 모두 정상적으로 연결되었다는 의미입니다.

2.3 Client Package 설치하기

2.3.1 Overview

IRIS에서는 어느 컴퓨터에서나 설치해서 사용할 수 있는 Client Package 를 제공하고 있습니다. 이를 다운로드 받아서 설치시 원하시는 IRIS 서버에 연결할 수 있습니다.

2.3.2 Client Package Overview

Client Package는 크게 두가지로 구성되어 있습니다.

- IRIS CLI Tool : iplus
- IRIS Python API

2.3.3 Client Package Install

IRIS Client Package는 IRIS 설치 프로그램과 독립적으로 제공됩니다.

```
IRISClient_[Build Date].tgz
```

위의 형태를 지니고 있는 파일을 받은 후에 압축을 풀어줍니다.

```
[anyone@localhost ~]$ ls
  IRISClient_[Build Date].tgz
[anyone@localhost ~]$ tar xzvf IRISClient_[Build Date].tgz
.
.
.
[anyone@localhost ~]$ ls
IRISClient  IRISClient_[Build Date].tgz
[anyone@localhost ~]$
```

압축을 해제하신 후에 폴더로 들어가보시면 setup.sh 라는 실행 파일이 있습니다. 이를 실행하면 환경 변수가 자동으로 세팅됩니다. 처음 실행시에는 바로 적용이 되지 않습니다. 다시 로그인을 하시거나 아래처럼 수동으로 적용합니다.

```
[anyone@localhost ~]$ cd IRISClient
[anyone@localhost IRISClient]$ ./setup.sh
[anyone@localhost IRISClient]$ source env.sh
[anyone@localhost IRISClient]$
```

2.3.4 IRIS CLI 실행하기

IRIS 클라이언트 패키지 설치 완료 후 IRIS에서 제공하는 CLI를 통해서 IRIS에 접속이 가능합니다. IRIS CLI는 아래 위치에 있습니다.

```
IRISclient/bin/iplus
```

다음과 같은 옵션으로 실행 가능합니다.

```
IRISclient/bin/iplus [ID]@[IRIS MASTER_IP]:[IRIS LISTENER PORT]
```

아래는 실제 실행하는 모습입니다. 아래에서 사용한 접속 정보는 다음과 같습니다.

접속 정보 항목	접속에 사용한 값
ID	test [IRIS에서 기본적으로 제공하는 값]
PASSWORD	test [IRIS에서 기본적으로 제공하는 값]
IRIS_MASTER_IP	192.168.111.100
IRIS_LISTENER_PORT	5050 [IRIS에서 기본값]

```
[anyone@localhost ~]$ IRISclient/bin/iplus test@192.168.111.100:5050
Password: test
Connecting to IRIS(test@192.168.111.100:5050).
Connected to IRIS.
Enter ".help" for instructions
iplus>
```


2.4 How to use IRIS CLI

2.4.1 iplus

iplus는 IRIS 자체적으로 제공하는 CLI Tool 입니다. 본 챕터에서는 IRIS Client에 포함되어 있는 iplus 를 사용하여 실제 IRIS에 접속, 여러 쿼리를 실행하는 방법에 대해서 설명합니다.

2.4.2 iplus 실행하기

iplus 실행은 아래와 같은 명령어로 실행이 가능합니다.

```
[IRISClient 압축 해제된 폴더]/bin/iplus [USER_ID]@[HOST]:[PORT]
```

여기서 사용하는 USER_ID는 IRIS 설치시에 기본적으로 제공하는 아이디를 사용합니다. 설치시 기본 제공하는 아이디와 패스워드는 다음과 같습니다.

```
test / test
```

접속 정보인 HOST:PORT는 다음과 같습니다.

```
[IRIS MASTER IP ADDRESS]:[PORT]
```

본 문서에서는 IRIS 마스터 노드 IP 주소는 192.168.111.100, 포트 정보는 기본 값인 5050으로 설명합니다. 위의 명령어로 실행을 하면 다음과 같은 콘솔이 나타납니다.

```
[client@client ~]$ cd [IRISclinet DIR]/bin
[client@client bin]$ ./iplus test@192.168.111.100:5050
iplus test@127.0.0.1:5050
Password:
Connecting to IRIS(test@127.0.0.1:5050).
Connected to IRIS.
Enter ".help" for instructions
iplus>
```

iplus> 콘솔이 나오면 정상입니다.

2.4.3 쿼리로 데이터 베이스 사용하기

2.4.3.1 테이블 생성 하기 테이블을 생성하기 위해서 기본적으로 확인해야 되는 것은 “테이블 생성 쿼리문” 과 “옵션” 입니다. 기본적으로 테이블 생성 쿼리문의 경우 다른 sql 문법과 동일합니다. 아래는 간단한 테스트 테이블 쿼리문입니다.

```
create table LOCAL_TEST_TABLE (
  k TEXT,
  p TEXT,
  a TEXT,
)
```

옵션 이름	값	설명
datascope	LOCAL GLOBAL	테이블 타입
ramexpire	min	Ram 에서 저장되는 시간. 분 단위로 지정가능
diskexpre	min	Disk 에서 저장되는 시간. 분 단위로 지정가능
partitionkey	column name	컬럼 중에서 key, 키 값으로 사용되는 컬럼 이름
partitiondate	column name	컬럼 중에서 partition, 파티션 값으로 사용되는 컬럼 이름
partitionrange	min	파티션을 나눌 시간 범위

이 외에 추가로 붙는 값이 바로 “옵션” 입니다. 옵션 값과 간단한 설명은 위의 표와 같습니다. 먼저 `datascope` 는 LOCAL, GLOBAL로 나뉘어집니다. LOCAL 테이블은 분산 되어서 저장되는 데이터로 대용량 데이터에 적합합니다. GLOBAL 테이블은 모든 노드에서 동일한 데이터를 유지하는 테이블로 소용량, 잦은 변경이 없는 데이터에 적합합니다. `ramexpire`, `diskexpire`는 각각 램 혹은 디스크에 데이터가 저장되는 시간입니다. 분 단위로 설정할 수 있습니다. `partition` 부분은 데이터를 분산하여 저장할 때 저장할 위치를 지정하는 옵션입니다.

“테이블 생성 쿼리문”, “옵션” 값을 더하면 다음과 같은 쿼리가 완성됩니다.

```
create table LOCAL_TEST_TABLE (
  k TEXT,
  p TEXT,
  a TEXT
)
datascope LOCAL
ramexpire 60
diskexpire 1440
partitionkey k
partitiondate p
partitionrange 10
;
```

위의 쿼리를 분석하면

`k`, `p`, `a` 컬럼을 가지고 있으며 모든 컬럼은 텍스트 데이터를 저장한다.
 LOCAL `datascope`를 가지고 있으며 분산 저장되고,
 데이터는 램에서 60분간 저장되며 디스크에서는 1440분, 하루 동안 저장된다.
 Partition Key는 `k` 컬럼의 값을 사용하고 Partition Date 는 `p` 값을 사용한다.
 파티션은 10분 단위로 생성된다.

로 이해할 수 있습니다. `iplus`에서 위의 쿼리를 실행합니다.

```

iplus> create table LOCAL_TEST_TABLE (
  > k TEXT,
  > p TEXT,
  > a TEXT
  > )
  > datascope LOCAL
  > ramexpire 60
  > diskexpire 1440
  > partitionkey k
  > partitiondate p
  > partitionrange 10
  > ;
Ret : +OK Create table

0.1945 sec

iplus>

```

실행이 정상적으로 실행되면 +OK 메시지와 함께 자세한 메시지가 나타납니다. 만약 에러가 발생한 경우에는 -ERR 메시지가 나타납니다.

2.4.3.2 테이블 조회 하기 테이블을 생성한 후에 +OK 메시지를 확인하여 테이블이 정상 생성된 것을 확인했지만 현재 테이블 리스트를 통하여 한 번 더 확인 하는 과정입니다. 테이블 리스트를 확인 하는 명령어는 먼저 아래와 같습니다.

```
.table list
```

위의 명령어를 이용하면 테이블 리스트를 확인할 수 있습니다. 그런데 명령어를 살펴보면 .table로 시작한다는 것이 보입니다. 즉, 다른 추가 명령이 있습니다. 이를 확인하기 위해서 .table 을 이용한 명령어를 실행해보면 아래와 같은 결과를 얻을 수 있습니다.

```

iplus> .table
Ret : +OK Success

HELP
=====
table command help
  list : show table list
         ex) .table list [options]
         you needs option help, .table list -h
  info : show detail table info
         ex) .table info [table name]
  schema : show tables schema
          ex) .table schema [table name]
=====

8 row in set
0.0807 sec

iplus> .table list
Ret : +OK Success

TABLE_NAME   SCOPE      RAM_EXP_TIME  DSK_EXP_TIME  KEY_STRING  PARTITION_STRING ...
=====
LOCAL_TEST_TABLE LOCAL      60            1440          k           p             ...
=====

1 row in set
0.2841 sec

iplus> .table info LOCAL_TEST_TABLE
Ret : +OK Success

TABLE_NAME   SCOPE      RAM_EXP_TIME  DSK_EXP_TIME  KEY_STRING  PARTITION_STRING ...
=====
LOCAL_TEST_TABLE LOCAL      60            1440          k           p             ...
=====

1 row in set
0.0407 sec

iplus> .table schema LOCAL_TEST_TABLE
Ret : +OK Success

```

```

SQL_SCRIPT
=====
CREATE TABLE LOCAL_TEST_TABLE ( K TEXT , P TEXT , A TEXT );
=====
1 row in set
0.0386 sec

iplus>

```

먼저 .table을 입력하면 HELP 메시지가 출력됩니다. 이곳에서 명령어 리스트와 옵션을 확인할 수 있습니다. 가장 처음에 나오는 명령어 순대로 살펴보면, list는 테이블 리스트를 출력하고, info [테이블 이름]은 테이블의 상세 정보를 보여주며, schema [테이블 이름]은 테이블 스키마를 보여줍니다.

2.4.3.3 테이블에 데이터 입력하기 테이블 생성하기에서 생성한 테이블에 데이터를 삽입하는 것을 설명합니다. 입력할 데이터는 아래와 같습니다.

```

k1,20150602000000,test_data_k1
k1,20150603000000,test_data_k1
k2,20150603000000,test_data_k2

```

위의 데이터를 이용하여 INSERT 쿼리를 만들면 아래와 같습니다.

```

insert into LOCAL_TEST_TABLE (k, p, a) values ('k1', '20150602000000', 'test_data_k1');
insert into LOCAL_TEST_TABLE (k, p, a) values ('k1', '20150603000000', 'test_data_k1');
insert into LOCAL_TEST_TABLE (k, p, a) values ('k2', '20150603000000', 'test_data_k2');

```

각각의 쿼리를 실행한 결과는 다음과 같습니다.

```

iplus> insert into LOCAL_TEST_TABLE (k, p, a)
> values ('k1', '20150602000000', 'test_data_k1');
Ret : +OK

```

```

0.1409 sec

iplus> insert into LOCAL_TEST_TABLE (k, p, a)
> values ('k1', '20150603000000', 'test_data_k1');
Ret : +OK

0.1091 sec

iplus> insert into LOCAL_TEST_TABLE (k, p, a)
> values ('k2', '20150603000000', 'test_data_k2');
Ret : +OK

0.1065 sec

iplus>

```

2.4.3.4 테이블에 있는 데이터 조회 하기 이전 챕터에서 삽입한 데이터를 조회합니다. 조회하는 쿼리는 아래와 같습니다.

```

SELECT * FROM LOCAL_TEST_TABLE;
SELECT COUNT(*) FROM LOCAL_TEST_TABLE;

```

첫번째 쿼리는 전체 데이터를 조회하는 쿼리이고 두번째 쿼리는 데이터 수를 조회하는 쿼리입니다. 각각의 쿼리 실행 결과는 다음과 같습니다.

```

iplus> SELECT * FROM LOCAL_TEST_TABLE;
Ret : +OK Success

  K           P           A
=====
k1      20150603000000 test_data_k1
k2      20150603000000 test_data_k2
=====

2 row in set
0.0827 sec

```

```

iplus> SELECT COUNT(*) FROM LOCAL_TEST_TABLE;
Ret : +OK Success

COUNT(*)
=====
2
=====
1 row in set
0.0823 sec

iplus>

```

결과를 보면 이전 챕터에서 분명히 세 개의 데이터를 입력했는데 왜 두 개만 남았지라는 의문이 들 수 있습니다. 테이블 생성시에 준 옵션을 다시 확인해보면 아래와 같습니다.

```
diskexpire 1440
```

위의 말은 1440분, 즉 하루가 지난 데이터를 삭제한다는 의미입니다. 제일 처음 입력한 데이터의 파티션은 20150602000000, 다음 데이터의 파티션은 20150603000000 이었습니다. 즉, 하루가 지난 데이터는 삭제를 수행했기에 하나가 삭제 되고 두 개의 데이터가 출력된 것입니다.

2.4.3.5 테이블에 있는 데이터 부분 조회 하기 IRIS에서는 데이터 조회에 있어서 “힌트”라는 기능을 제공합니다. 이 기능은 대용량 데이터를 저장했을 때 더 빠른 조회를 위한 기능입니다. 기본적으로 IRIS에서 데이터를 조회할 때 모든 데이터를 조회하게 되는데, 이 기능을 이용할 경우 일부분에 대해서만 조회가 가능합니다. 부분 조회를 할 수 있는 범위는 처음 테이블을 생성할 때 옵션으로 추가한 partition, 파티션 부분입니다.


```
partitionkey k
partitiondate p
partitionrange 10
```

partitionkey 는 편의상 key, 키라고 부르며 partitiondate 는 편의상 partition, 파티션이라고 부릅니다. 다음은 키 값이 “k2” 인 데이터에서만 조회를 하는 쿼리입니다.

```
/** LOCATION ( KEY = 'k2' ) */ SELECT * FROM LOCAL_TEST_TABLE;
```

쿼리에 대해서 자세하게 설명하기 전에 먼저 결과를 살펴보면 아래와 같습니다.

```
iplus> /** LOCATION ( KEY = 'k2' ) */ SELECT * FROM LOCAL_TEST_TABLE;
Ret : +OK Success

K          P          A
=====
k2          20150603000000 test_data_k2
=====
1 row in set
0.0454 sec

iplus>
```

결과를 보면 분명 K 컬럼의 값이 'k1' 인 데이터가 하나 'k2' 인 데이터가 하나가 있었는데 'k2' 값만 출력되었습니다. 이는 힌트 때문인데, 쿼리에서 힌트 부분만 빼어서 살펴보면 아래와 같습니다.

```
/** LOCATION ( KEY = 'k2' ) */
```

“/*+” 로 시작하고 “*/” 로 힌트의 종료를 알립니다. 그 사이 값을 보면 LOCATION (...) 를 확인 할 수 있는데 이 LOCATION 안에 있는 데이터만 조회하겠다는 의미입니다. 괄호 내부에 있는 값이 바로 범위를 준것입니다. 이 쿼리를 조금 변화 시킨 결과는 아래와 같습니다.

```

iplus> /*+ LOCATION ( KEY = 'k1' ) */ SELECT * FROM LOCAL_TEST_TABLE;
Ret : +OK Success

  K          P          A
=====
k1          20150603000000 test_data_k1
=====

1 row in set
0.0453 sec

iplus> /*+ LOCATION ( PARTITION > '20150604000000' ) */ SELECT * FROM LOCAL_TEST_TABLE;
Ret : +OK Success

  K          P          A
=====
=====

0 row in set
0.0374 sec

iplus>

```

첫 번째 쿼리는 키 값이 'k1'인 값에 대한 조회를 한 것입니다. 결과는 예상했듯이 하나의 결과가 나옵니다. 두 번째 쿼리는 조건이 조금 다릅니다. 바로 PARTITION > '20150604000000' 입니다. 이것은 파티션을 범위로 줘서 조회를 시도한 경우입니다. 내용을 살펴 보면 파티션 값이 2015년 06월 05일 00시 00분 00초 이후의 데이터를 이용하라는 의미입니다. 결과 2015년 06월 05일 이후의 데이터는 없으므로 0건이 출력됩니다.

2.4.3.6 테이블에 있는 데이터 업데이트, 삭제 그리고 테이블 삭제 이 챕터에서는 데이터를 업데이트, 삭제, 그리고 마지막으로 테이블 삭제하는 방법에 대해서 다룹니다. IRIS에서는 대부분의 SQL 쿼리를 그대로 지원합니다. 아래 쿼리를 보시면

```
UPDATE LOCAL_TEST_TABLE SET A = 'k1 updated' where K = 'k1';
DELETE FROM LOCAL_TEST_TABLE where K = 'k2';
drop table LOCAL_TEST_TABLE;
```

첫 번째 쿼리는 UPDATE 쿼리입니다. LOCAL_TEST_TABLE 에서 K 값이 'k1' 인 데이터의 A 값을 'k1 updated' 로 변경하는 쿼리입니다.

```
iplus> select * from LOCAL_TEST_TABLE;
Ret : +OK Success

  K          P          A
=====
  k1        20150603000000 test_data_k1
  k2        20150603000000 test_data_k2
=====

2 row in set
0.0913 sec

iplus> UPDATE LOCAL_TEST_TABLE SET A = 'k1 updated' where K = 'k1';
Ret : +OK

0.0635 sec

iplus> select * from LOCAL_TEST_TABLE;
Ret : +OK Success

  K          P          A
=====
  k1        20150603000000 k1 updated
  k2        20150603000000 test_data_k2
=====

2 row in set
```

```
0.0492 sec
```

```
iplus>
```

두 번째 쿼리는 DELETE 쿼리입니다. LOCAL_TEST_TABLE 에서 K 값이 'k2' 인 데이터를 삭제하는 쿼리입니다.

```
iplus> select * from LOCAL_TEST_TABLE;
Ret : +OK Success
```

K	P	A
k1	20150603000000	k1 updated
k2	20150603000000	test_data_k2

```
2 row in set
```

```
0.0476 sec
```

```
iplus> DELETE FROM LOCAL_TEST_TABLE where K = 'k2';
```

```
Ret : +OK
```

```
0.0844 sec
```

```
iplus> select * from LOCAL_TEST_TABLE;
```

```
Ret : +OK Success
```

K	P	A
k1	20150603000000	k1 updated

```
1 row in set
```

```
0.0446 sec
```

```
iplus>
```

마지막 쿼리는 테이블을 삭제하는 쿼리입니다.

```
iplus> .table list
Ret : +OK Success

TABLE_NAME    SCOPE          RAM_EXP_TIME  DSK_EXP_TIME  KEY_STRING    PARTITION_STRING ...
=====
LOCAL_TEST_TABLE LOCAL          60            1440          k             p             ...
=====

1 row in set
0.2801 sec

iplus> drop table LOCAL_TEST_TABLE;
Ret : +OK Drop Table

0.2321 sec

iplus> .table list
Ret : +OK Success

TABLE_NAME    SCOPE          RAM_EXP_TIME  DSK_EXP_TIME  KEY_STRING    PARTITION_STRING ...
=====
=====

0 row in set
0.2724 sec

iplus >
```

위와 같이 대부분의 SQL 문은 사용이 가능합니다.