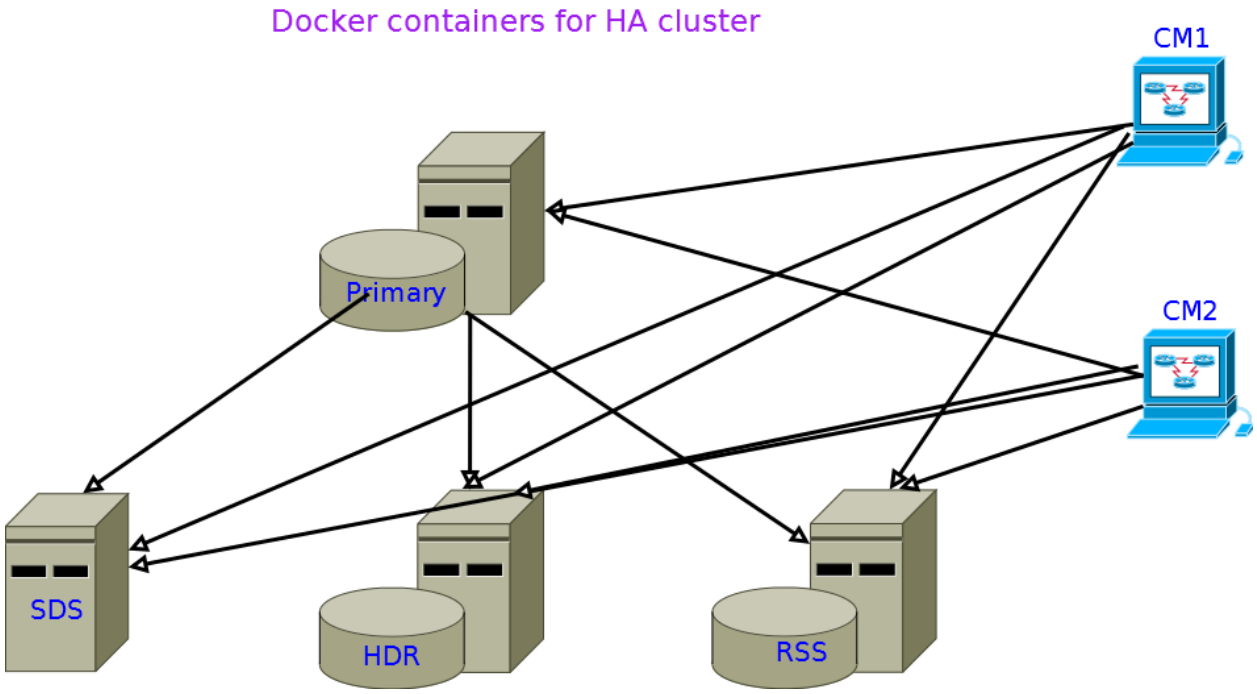
Docker Hands on Lab Instructions



Server related docker files at c:\docker\server

Connection manager docker files at c:\docker\cm

Step 1: Verify Docker installation.

Start Docker quick start terminal by clicking Docker icon on the desktop.

Run following command to verify Docker installation

$ docker run hello-world

First two lines of the output should be similar to

Hello from Docker.

This message shows that your installation appears to be working correctly.

Step 2: Build Docker image for Informix server

From Docker quickstart terminal window

$ cd /c/docker/server/

Files in /c/docker/server/:

Dockerfile : Commands to build Informix Server Docker image. **Review commands in this file.**

Boot.sh: Container startup script to manage Informix server. Briefly review logic in this script.

bundle.properties: Configuration file for Informix silent install.

iif.12.10.tar : Informix bundle tar file.

Command to build Informix server Docker image:

$ docker build -t iiug/informix .

Step 3: Build Docker image for Connection Manager.

From Docker quick start terminal window

$ cd /c/docker/cm

Files in /c/docker/cm:

Dockerfile : Commands to build Informix connection manager Docker image. **Review commands in this file.**

boot_cm.sh : Container startup script to manage connection manager. Briefly review logic in this script.

response.txt: Configuration file for client sdk silent install.

clientsdk.4.10.tar : Client SDK tar file.

Command to build connection manager Docker image:

$ docker build -t iiug/cm .

Step 4: Create private network for Informix containers.

$ docker network create --subnet=172.18.0.0/16 informix_nw

informix_nw is the private network name. This network to be used while creating Informix containers.

Step 5: Verify informix_nw details.

$ docker network inspect informix_nw

Step 6: Create Informix primary server container.

$ docker run --net=informix_nw --ip=172.18.0.10 -d -h primary --name primary iiug/informix

--net is for which network to use.

--ip is the static ip address to use.

-h is for hostname.

--name is for container name.

-d option: run container in background.

Docker run command creates and starts new container.

Step 7: Verify primary server state and wait for sysadmin database to be built.

$ docker exec -it primary /opt/ibm/boot.sh --shell onstat -m

Step 8: Create and start container for Informix HDR Secondary server

--initSec option initializes configuration files.

$ docker run --net=informix_nw --ip=172.18.0.11 -d -h hdr --name hdr iiug/informix --initSec hdr

Add HDR secondary host details to primary server

$ docker exec primary /opt/ibm/boot.sh --addHost 172.18.0.11 hdr

Now clone HDR secondary server from primary server. --initHDR option need primary server name and its ip address as input.

$ docker exec hdr /opt/ibm/boot.sh --initHDR primary 172.18.0.10

After previous command exits, wait for 10 seconds and check HDR secondary server status.

$ docker exec -it hdr /opt/ibm/boot.sh --shell onstat -g dri

*IBM Informix Dynamic Server Version 12.10.FC6 -- **Read-Only (Sec)** -- Up 00:11:11*

-- 156276 Kbytes

Data Replication at 0x45a4a028:

Type State Paired server Last DR CKPT (id/pg) Suppo

rts Proxy Writes

*HDR Secondary **on** primary 6 / 195 N*

Step 9: Create and start container for Informix RS Secondary server

--initSec option initializes configuration files.

$ docker run --net informix_nw --ip 172.18.0.12 -d -h rss --name rss iiug/informix --initSec rss

Add RS secondary host details to primary server

$ docker exec primary /opt/ibm/boot.sh --addHost 172.18.0.12 rss

Add RS secondary host details to HDR secondary server

$ docker exec hdr /opt/ibm/boot.sh --addHost 172.18.0.12 rss

Now clone RS secondary server from primary server. --initRSS option need primary server name and its ip address as input.

$ docker exec rss /opt/ibm/boot.sh --initRSS primary 172.18.0.10

Now check RSS server status.

$ docker exec -it rss /opt/ibm/boot.sh --shell onstat -

*IBM Informix Dynamic Server Version 12.10.FC6 -- **Read-Only (RSS)** -- Up 00:19:49*

-- 164468 Kbytes

Step 10: Create and start container for Informix Shared Disk Secondary server

First primary server as disk owner for shared disk.

$ docker exec primary /opt/ibm/boot.sh --setSDSPrim

Create and start SDS container. --initSec option initializes configuration files.

--volumes-from primary option mounts primary server container disk volumes to SDS container

$ docker run --net informix_nw --ip 172.18.0.13 -d --volumes-from primary -h sds --name sds iiug/informix --initSec sds

Add SDS host information to primary, HDR and RSS servers.

$ docker exec primary /opt/ibm/boot.sh --addHost 172.18.0.13 sds

$ docker exec hdr /opt/ibm/boot.sh --addHost 172.18.0.13 sds

$ docker exec rss /opt/ibm/boot.sh --addHost 172.18.0.13 sds

Start SDS server using –initSDS option. It needs primary server name and ip address as input.

$ docker exec sds /opt/ibm/boot.sh --initSDS primary 172.18.0.10

Now verify SDS server status.

$ docker exec -it sds /opt/ibm/boot.sh --shell onstat -

*IBM Informix Dynamic Server Version 12.10.FC6 -- **Read-Only (SDS)** -- Up 00:40:58*

-- 198736 Kbytes

Step 11: Verify cluster status at primary server.

$ docker exec -it primary /opt/ibm/boot.sh --shell onstat -g cluster

IBM Informix Dynamic Server Version 12.10.FC6 -- **On-Line (Prim)** -- Up 00:43:52 -

- 164468 Kbytes

Primary Server:primary

Current Log Page:6,710

Index page logging status: Enabled

Index page logging was enabled at: 2016/04/08 20:25:01

Server	ACKed Log (log, page)	Applied Log (log, page)	Supports Updates	Status
sds	6,710	6,710	No	**SYNC(SDS),Connected,Active**
hdr	6,710	6,710	No	**ASYNC(HDR),Connected,On**
rss	6,710	6,710	No	**ASYNC(RSS),Connected,Active**

Step 12: Create and start container for first connection manager (cm1)

Add connection manager 1 ip address and host name to primary server as a trusted host.

#Note: Primary server propagates this new trusted host record to its secondary servers.

$ docker exec primary /opt/ibm/boot.sh --addHost 172.18.0.14

$ docker exec primary /opt/ibm/boot.sh --addHost cm1

Create and start container for first connection manager (cm1)

#–initCM needs primary server name, ip address, connection manager name and priority number (used to determine failover arbitrator) as input.

$ docker run --net informix_nw --ip 172.18.0.14 -d -h cm1 --name cm1 iiug/cm --initCM primary 172.18.0.10 cm1 1

Step 13: Create and start container for second connection manager (cm2)

Add connection manager 2 ip address and host name to primary server as a trusted host.

#Note: Primary server propagates this new trusted host record to its secondary servers.

$ docker exec primary /opt/ibm/boot.sh --addHost 172.18.0.15

$ docker exec primary /opt/ibm/boot.sh --addHost cm2

Create and start container for first connection manager (cm2)

#–initCM needs primary server name, ip address, connection manager name and priority number (used to determine failover arbitrator) as input.

$ docker run --net informix_nw --ip 172.18.0.15 -d -h cm2 --name cm2 iiug/cm --initCM primary 172.18.0.10 cm2 2

Step 14: Connect to primary server and verify connection manager status.

$ docker exec -it primary /opt/ibm/boot.sh --shell onstat -g cmsm

IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -- Up 03:06:36 -

- 164468 Kbytes

Unified Connection Manager: cm1 *Hostname: cm1.informix_nw*

CLUSTER g_cluster LOCAL

Informix Servers: primary,hdr,rss,sds

SLA Connections Service/Protocol Rule

oltp 0 60000/onsoctcp DBSERVERS=primar

y

report 0 60001/onsoctcp DBSERVERS=(HDR,S

DS,RSS) POLICY=ROUNDROBIN

Failover Arbitrator: Active Arbitrator, Primary is up

ORDER=SDS,HDR,RSS PRIORITY=1 TIMEOUT=1

Unified Connection Manager: cm2 *Hostname: cm2.informix_nw*

CLUSTER g_cluster LOCAL

Informix Servers: primary,hdr,rss,sds

SLA Connections Service/Protocol Rule

oltp 0 60000/onsoctcp DBSERVERS=primary

report 0 60001/onsoctcp DBSERVERS=(HDR,S

DS,RSS) POLICY=ROUNDROBIN

Failover Arbitrator: Primary is up

ORDER=SDS,HDR,RSS PRIORITY=2 TIMEOUT=1

Step 15: Test failover

```
#Check primary server status.

$ docker exec -it primary /opt/ibm/boot.sh --shell onstat -

      IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -
      - Up 00:06:28 -
      - 164468 Kbytes


# Shutdown primary docker image

$ docker stop primary
      primary

# Wait for 30 seconds and check SDS server status. SDS should get
promoted to primary server by connection manager.

$ docker exec -it sds /opt/ibm/boot.sh --shell onstat -


      IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -
      - Up 00:08:06 -
      - 198736 Kbytes


# Now check cluster status

$ docker exec -it sds /opt/ibm/boot.sh --shell onstat -g cluster

      IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -
      - Up 00:08:32 -
      - 198736 Kbytes

      Primary Server:sds
      Current Log Page:4,280
      Index page logging status: Enabled
      Index page logging was enabled at: 2016/04/13 19:12:18


      Server ACKed Log    Applied Log  Supports     Status
             (log, page)  (log, page)  Updates
      hdr    4,280        4,280        No
      ASYNC(HDR),Connected,On
      rss    4,280        4,280        No
      ASYNC(RSS),Connected,Active
```

Step 16: Shutdown and restart RSS container and make sure RSS server restarts.

```
# Check RSS server status:

$ docker exec -it rss /opt/ibm/boot.sh --shell onstat -g rss

      IBM Informix Dynamic Server Version 12.10.FC6 -- Read-Only (RSS)
      -- Up 00:10:44
      -- 206928 Kbytes

      Local server type: RSS
      Server Status : Active
      Source server name: sds
      Connection status: Connected
      Last log page received(log id,page): 4,280



# Shutdown RSS container

$ docker stop rss
      rss

# Check cluster status at primary server. RSS should be in
disconnected state.

$ docker exec -it sds /opt/ibm/boot.sh --shell onstat -g cluster

      IBM Informix Dynamic Server Version 12.10.FC6 -- On-Line (Prim) -
      - Up 00:11:08 -
      - 206928 Kbytes

      Primary Server:sds
      Current Log Page:4,286
      Index page logging status: Enabled
      Index page logging was enabled at: 2016/04/13 19:12:18


      Server ACKed Log    Applied Log  Supports     Status
             (log, page)  (log, page)  Updates
      hdr    4,286        4,286        No
      ASYNC(HDR),Connected,On
      rss    0,0          0,0          No
      ASYNC(RSS),Disconnected,Inactive

# Restart RSS container.

$ docker start rss
```

```
      rss

# Check RSS server status:

$ docker exec -it rss /opt/ibm/boot.sh --shell onstat -g rss

      IBM Informix Dynamic Server Version 12.10.FC6 -- Read-Only (RSS)
      -- Up 00:00:20
      -- 198736 Kbytes

      Local server type: RSS
      Server Status : Active
      Source server name: sds
      Connection status: Connected
      Last log page received(log id,page): 4,288
```

Thanks for completing lab instructions.