XTIVIA

www.xtivia.com
888.685.3101 ext. 2

Docker Cookbook: Practical Recipes for Running Db2

Agenda

- Intro to Containers and Docker
- Recipe #1: Simplify Db2 Fixpacks & Upgrades
- Recipe #2: Extending Existing Containers
- Recipe #3: Automating Maintenance
- Recipe #4: Handling Db2 Authentication
- Questions

XTIVIA

What are Containers?

- Definition

> *An Application Container is a construct designed to package and run an application or its components running on a shared Operating System.*

- Container Technologies
  - Linux: Containers ("LXC")
  - AIX: Workload Partitions ("WPAR")
  - Solaris: Zones

Containers vs. Virtual Machines

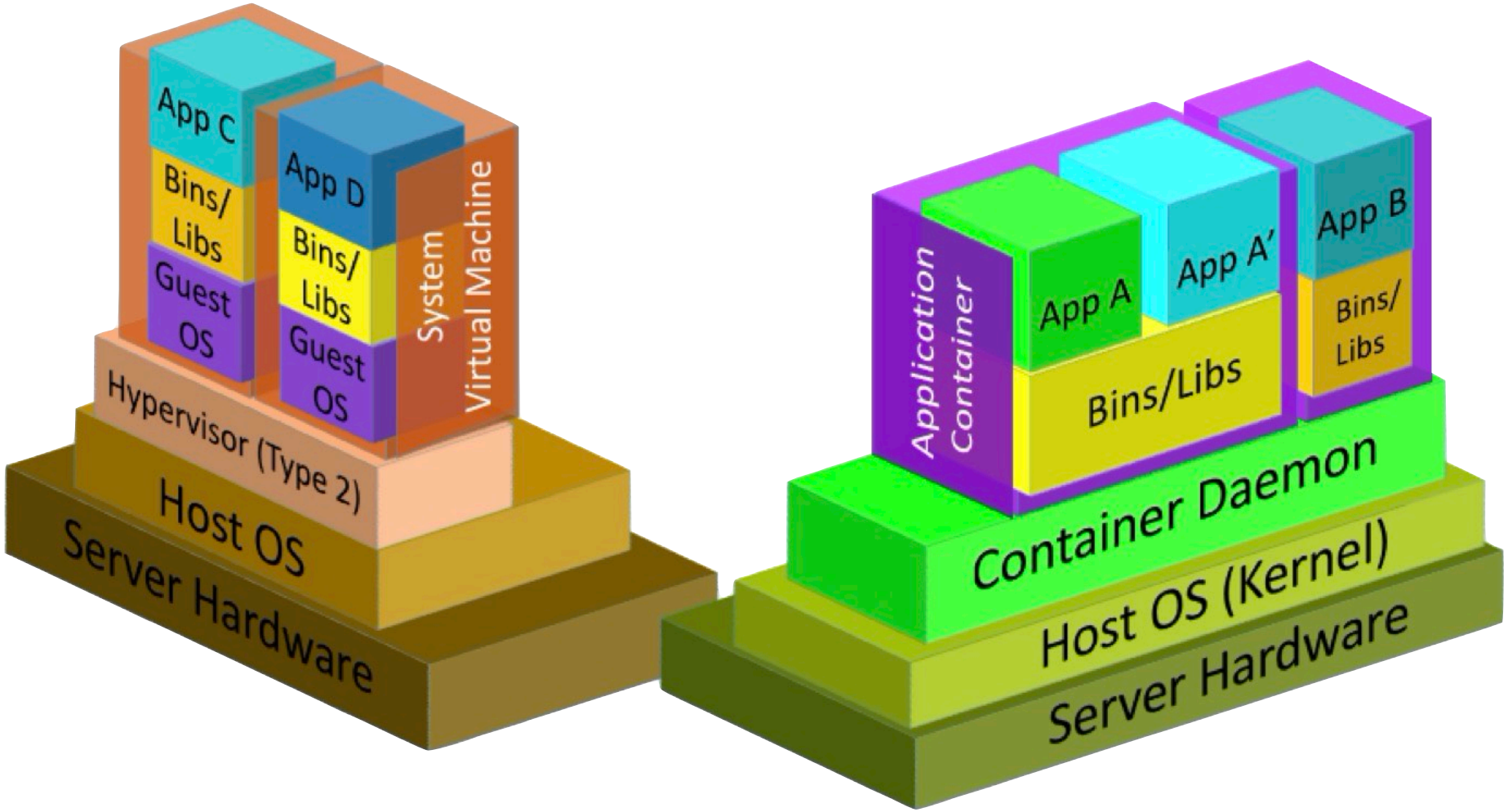


Image Credit: NIST Publication 180-800

XTIVIA

Containers vs. Virtual Machines

XTIVIA

Containers vs. Virtual Machines

Virtual Machine : House

- Dedicated Infrastructure
  - Foundation
  - Structure

- Single Family

Container : Apartment

XTIVIA

Containers vs. Virtual Machines

Virtual Machine : House

- Dedicated Infrastructure
  - Foundation
  - Structure
- Single Family

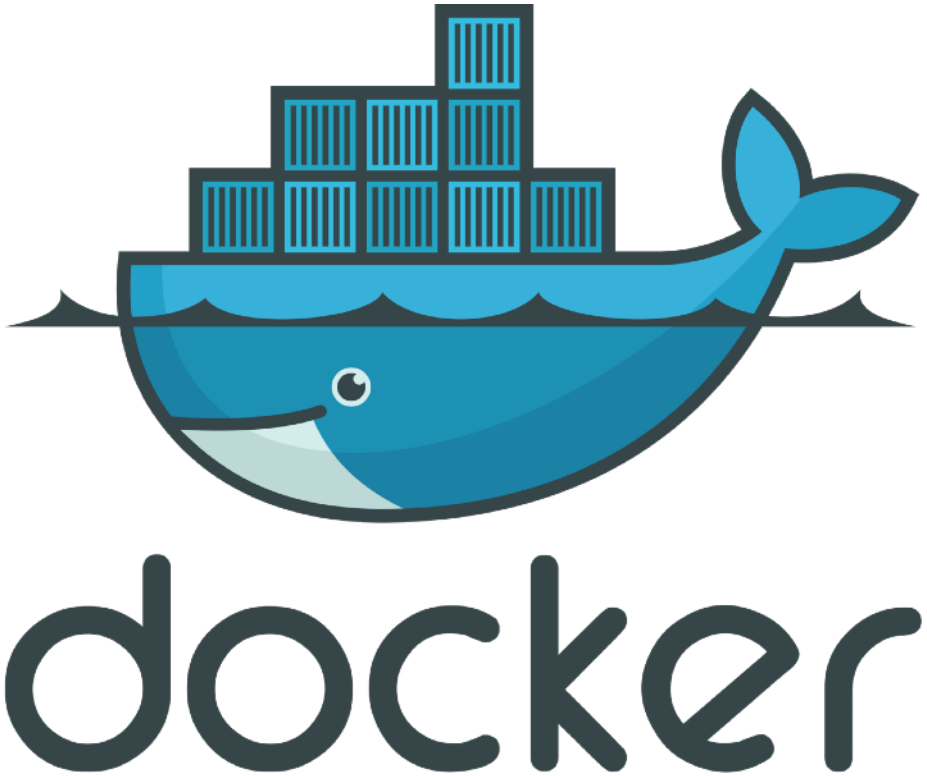
Container : Apartment

- Shared Infrastructure
  - Electric
  - Plumbing
  - Heating / Cooling
- High Density

XTIVIA

Introduction to Docker

- Initial release in 2013
- Open Source
- Downloaded over 13,000,000,000 times
- Community and Enterprise Editions
- Uses Linux kernel virtualization features
- Platform Support:
  - Linux, Windows, macOS, even zOS (!)
  - Cloud providers offer container services

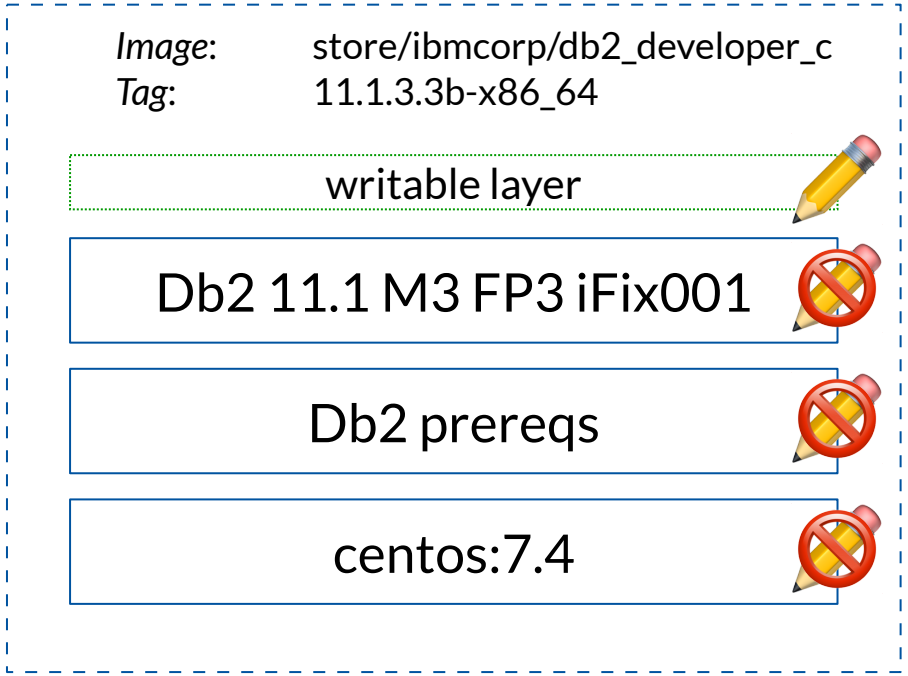


Storage in Containers: Union Filesystem

- An Image is built on a series of read-only layers
  - Each layer represents an instruction during the image build process

Storage in Containers: Union Filesystem

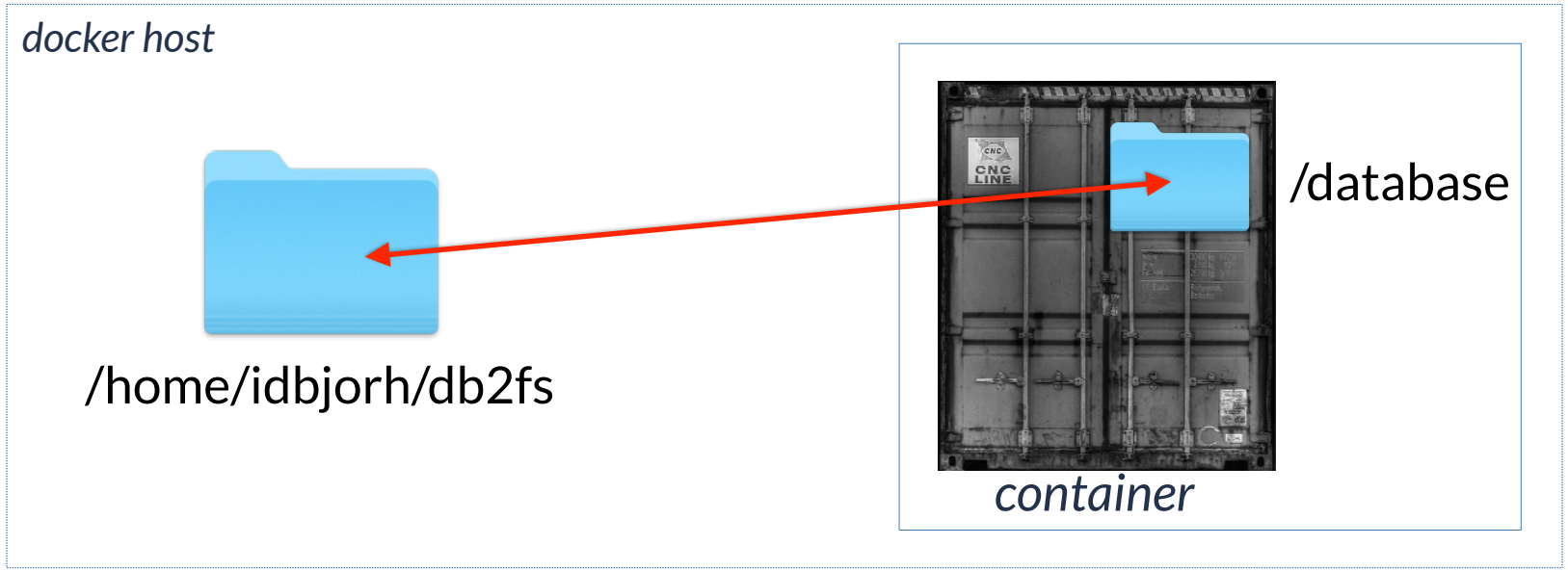
- An Image is built on a series of read-only layers
  - Each layer represents an instruction during the image build process
- When a container is started, a thin writable layer is allocated



XTIVIA

Storage in Containers: Bind Mounts

- "Share" a directory on the host to a directory in the container



XTIVIA

Agenda

✓Intro to Containers and Docker

- **Recipe #1: Simplify Db2 Fixpacks & Upgrades**
- Recipe #2: Extending Existing Containers
- Recipe #3: Automating Maintenance
- Recipe #4: Handling Db2 Authentication
- Questions

XTIVIA

Recipe #1: Simplifying Db2 Maintenance & Upgrades

- Planning
  - How long does it take to move to a new fixpack? A new version ?
  - Who needs to be involved?
  - How will you install the new code?
  - How will you clean up old code?
- Implementation
  - Required Outage
  - Complexity - Many steps to install

Can your developers do this by themselves?

XTIVIA

Container Rules of Engagement

- An ideal image contains only* required system binaries, libraries, application binaries, and dependencies


`/usr/bin`


`/usr/lib`


~~`/home/db2inst1`~~


`/opt/ibm/db2`


`/var/db2`

* a container may contain static data, but ideally all writable data will reside in volumes or bind-mounts

XTIVIA

Container Rules of Engagement

- An ideal image contains only* system binaries, application binaries, libraries and dependencies
- Containers must be considered ephemeral
- Software updates are not applied to containers
  - Updates are deployed in new images

Upgrading an Existing Container

```
$ docker ps
CONTAINER ID        IMAGE                                                COMMAND
d0dfc00745a1        store/ibmcorp/db2_developer_c:11.1.3.3a-x86_64   "/var/db

$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREAT
store/ibmcorp/db2_developer_c   11.1.4.4-x86_64     399b96dc7c1b        3 mor
store/ibmcorp/db2_developer_c   11.1.3.3b-x86_64    cf7f331251df        5 mor
store/ibmcorp/db2_developer_c   11.1.3.3a-x86_64    18585aea3e7b        9 mor
store/ibmcorp/db2_developer_c   11.1.3.3x-x86_64    439cb542a179        11 mo
store/ibmcorp/db2_developer_c   11.1.3.3-x86_64     c1aa24cafc56        12 mo
store/ibmcorp/db2_developer_c   11.1.2.2b-x86_64    9801629b153b        13 mo
```


XTIVIA

Check and Stop Existing Container

```
$ docker exec -it db2server su - db2inst1 -c db2level
DB21085I  This instance or install (instance name, where applicable:
"db2inst1") uses "64" bits and DB2 code release "SQL11013" with level
identifier "0204010F".
Informational tokens are "DB2 v11.1.3.3", "s1804271300", "DYN1804271300AMD64"
and Fix Pack "3a".
Product is installed at "/opt/ibm/db2/V11.1".

$ docker inspect db2server | grep -A2 \"bind\",
                "Type": "bind",
                "Source": "/home/idbjorh/db2fs",
                "Destination": "/database",

$ docker stop db2server
db2server
```


XTIVIA

Start New Container with Existing Storage

```
$ docker run -h db2server --name db2server_M4FP4 --detach \
     --privileged=true -p 50000:50000 --env-file env-list \
     -v /home/idbjorh/db2fs:/database \
     store/ibmcorp/db2_developer_c:11.1.4.4-x86_64
af5953a9feaaac759c05c90eade8cb6325f9644aecc4c98305d56bf4fc6857d3

$ docker exec -it db2server_M4FP4 su - db2inst1 -c db2level
DB21085I  This instance or install (instance name, where applicable:
"db2inst1") uses "64" bits and DB2 code release "SQL11014" with level
identifier "0205010F".
Informational tokens are "DB2 v11.1.4.4", "s1811091400", "DYN1811091400AMD64"
and Fix Pack "4".
Product is installed at "/opt/ibm/db2/V11.1".
```

Upgrade Effort:
< 2 minutes!

XTIVIA

Fixpack / Upgrade Gotchas

- Plan to handle container name conflicts
  - Use new container name (as shown in example)
  - Remove old container before starting new container

```
docker rm ...
```

- Don’t forget database backup
- Some post-upgrade tasks may still be required
  - Bind CLI utilities
  - db2updv111

Agenda

- ✓ Intro to Containers and Docker
- ✓ Recipe #1: Simplify Db2 Fixpacks & Upgrades
- **Recipe #2: Extending Existing Containers**
- Recipe #3: Automating Maintenance
- Recipe #4: Handling Db2 Authentication
- Questions

XTIVIA

Recipe #2: Extending Existing Containers

“This container is good, but…”

XTIVIA

Recipe #2: Extending Existing Containers

- The IBM images in Docker Hub are quite good

- Potential improvements:
  - Db2 Developer-C Image takes too long to start (~90 seconds)
  - Database is not automatically activated

- Solutions?
  - Create custom image from scratch
  - Extend existing image

Creating an Image from Scratch

Pros

- Complete control!

Cons

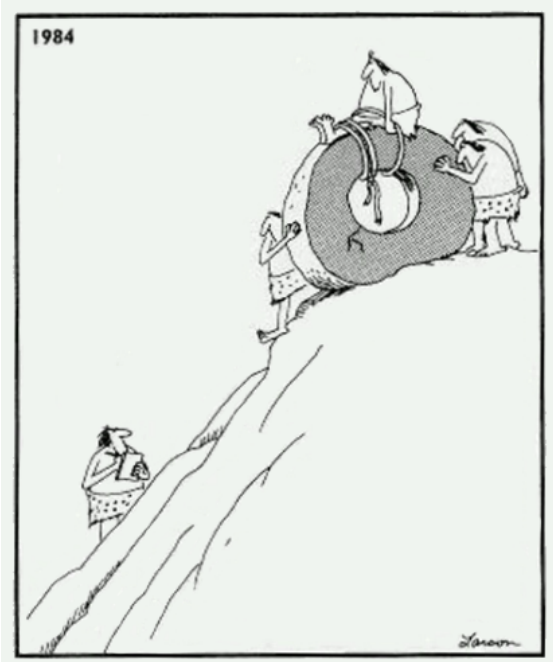
- Complicated
- Steep Learning Curve
- Lots of testing required

XTIVIA

Extending an Existing Image

Pros

- Change only what you need
- Not reinventing the wheel



Early experiments building Docker images

Cons

- Neet to build / maintain custom docker images
- Dependent on upstream image for updates
- How to deploy the image?

XTIVIA

Custom Script to Activate Database(s)

```
#!/bin/bash
#
#  activate-dbs.sh -- Activate databases known to this container
#  2019-02-18  Ian Bjorhovde <ibjorhovde@xtivia.com>
#

# Execute this script as instance owner
if [[ ${UID} -eq 0 ]] ; then
    exec su - ${DB2INSTANCE} -c ${0}
fi

. ${HOME}/sqllib/db2profile

for db in `db2fupdt -f /database/config/instance.cfg -s database -p list | sed -e 's/,/ /g'` ; do
   echo "(*) Activating database ${db}..."
   db2 activate database ${db}
done
```


XTIVIA

Modify ENTRYPOINT script for TEXT_SEARCH

- Add check for TEXT_SEARCH environment variable

```
...

# Check for the (custom) TEXT_SEARCH env var.
if [ ${TEXT_SEARCH:-true} = "true" ] ; then
    echo "(*) Starting TEXT SEARCH service ..."
    su - ${DB2INSTANCE?} -c "db2ts \"start for text\""
fi

...
```


XTIVIA

Build and Run Image

```
$ cat Dockerfile
FROM store/ibmcorp/db2_developer_c:11.1.4.4-x86_64

RUN  mkdir /var/custom

COPY db2_common_functions /var/db2_setup/include
COPY setup_db2_instance.sh /var/db2_setup/lib
COPY activate_dbs.sh /var/custom
```


XTIVIA

Build Custom Image (1|2)

```
$ docker build -t custom:11.1.4.4 .
Sending build context to Docker daemon   34.3kB
Step 1/5 : FROM store/ibmcorp/db2_developer_c:11.1.4.4-x86_64
 ---> 399b96dc7c1b
Step 2/5 : COPY db2_common_functions /var/db2_setup/include
 ---> 69c0b584ad5f
Step 3/5 : COPY setup_db2_instance.sh /var/db2_setup/lib
 ---> 4b9ffa7d7def
Step 4/5 : RUN mkdir /var/custom
 ---> Running in 7d5c54c0ed35
Removing intermediate container 7d5c54c0ed35
 ---> cf866ccb2e98
Step 5/5 : COPY activate_dbs.sh /var/custom
 ---> 31981f29225c
Successfully built 31981f29225c
Successfully tagged custom:11.1.4.4
```


XTIVIA

Build Custom Image (2|2)

```
$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREAT
custom                              11.1.4.4            584ca8432347        1 mir
store/ibmcorp/db2_developer_c       11.1.4.4-x86_64     399b96dc7c1b        3 mor
store/ibmcorp/db2_developer_c       11.1.3.3b-x86_64    cf7f331251df        6 mor
```


XTIVIA

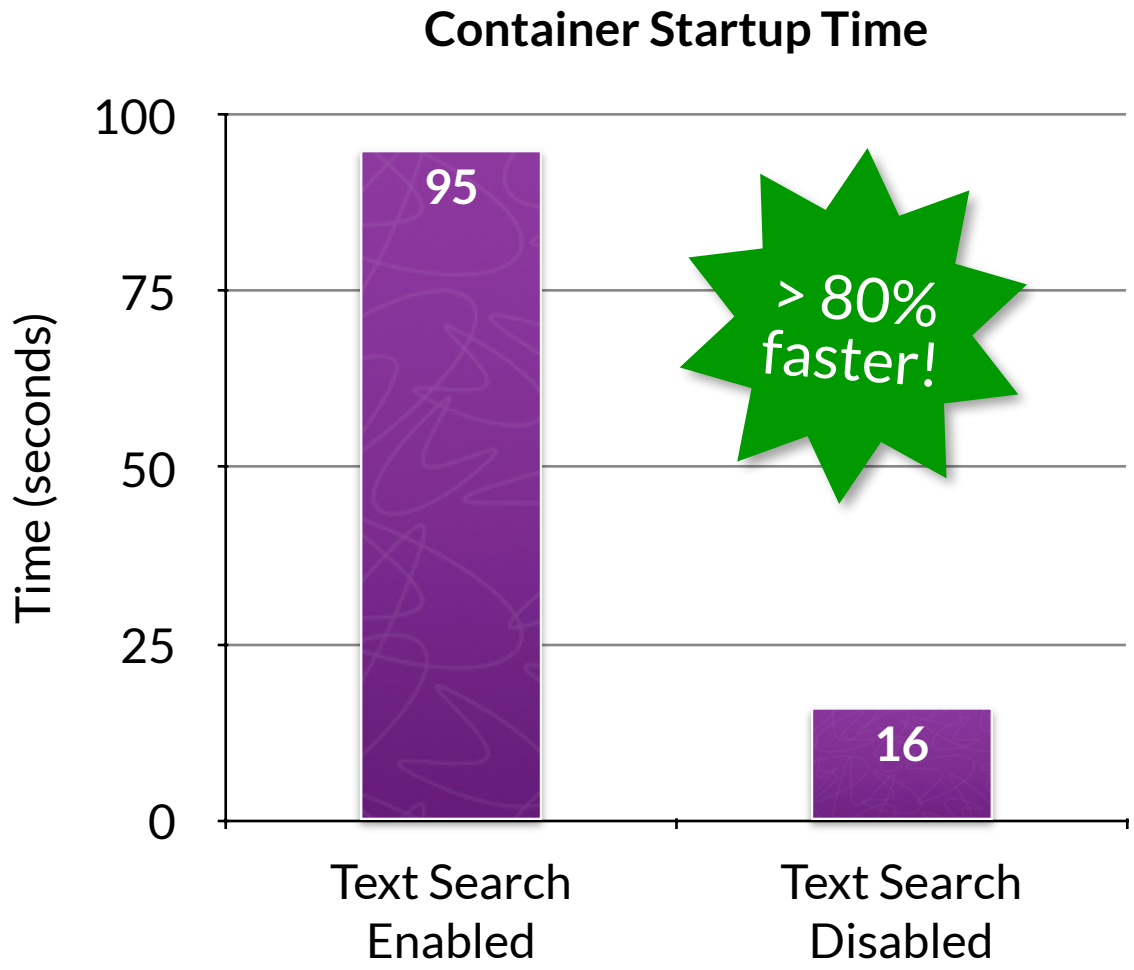
Run Container with new Image

```
$ docker run -h db2server --name custom_M4FP4 --detach \
     --privileged=true -p 50000:50000 -e TEXT_SEARCH=false \
     --env-file env-list -v /home/idbjorh/db2fs:/database custom:11.1.4.4
79a30e7a5391a73ee3f2dd9feaa2cae2de43c51a8769c1958313e38996caa0fe

$ docker exec -it custom_M4FP4 su - db2inst1 -c "db2 list active databases"

                                Active Databases

Database name                                = TESTDB
Applications connected currently             = 0
Database path                                = /database/data/db2inst1/NOD...
```


XTIVIA

Result

- Significantly Faster start time for containers
- Database(s) are activated when container starts



XTIVIA

Agenda

✓Intro to Containers and Docker

✓Recipe #1: Simplify Db2 Fixpacks & Upgrades

✓Recipe #2: Extending Existing Containers

- **Recipe #3: Automating Maintenance**
- Recipe #4: Handling Db2 Authentication
- Questions

XTIVIA

Recipe #3: Automating Maintenance

```
$ docker exec -it db2server su - db2inst1
Last login: Sun Mar 31 21:34:12 UTC 2019

[db2inst1@db2server ~]$ crontab -e
-bash: crontab: command not found

[db2inst1@db2server ~]$ rpm -qi cronie
package cronie is not installed
```

XTIVIA

Where is cron?

- Containers don't run system daemons (crond, ntpd, atd, etc.)
  - It's possible to install and run crond
  - Running daemons violates the spirit of containers

XTIVIA

Options for Job Automation

- Db2 Automatic Task Scheduler
- Cron jobs on DOCKER_HOST
- Cron and Docker CLI on remote host

Db2 Automatic Task Scheduler

- Enabled by default on IBM Db2 Developer-C Images

```
db2set DB2_ATS_ENABLE=YES
```

- Control job schedule:
  - ADMIN_TASK_ADD / ADMIN_TASK_UPDATE / ADMIN_TASK_REMOVE

- ATS can only execute Stored Procedures
- Limited notification options
- Jobs are only executed if database is activated

ATS Job Schedule - Add Job

```
call admin_task_add (
   'DAILY BACKUP',
   current timestamp,
   NULL,
   NULL,
   '5 13 * * *',  -- MINUTE HOUR DAY_OF_MONTH MONTH DAY_OF_WEEK
   'SYSPROC',
   'ADMIN_CMD',
   'VALUES(''BACKUP DATABASE TESTDB ONLINE TO /database/backup WITHOUT PROMPT
   NULL,
   'Backup Job Description'
);
```


XTIVIA

ATS Job Schedule - “crontab -l”

```
SELECT
   SCHEDULE ||
   '   call ' || rtrim(PROCEDURE_SCHEMA) || '.' || PROCEDURE_NAME ||
   ' (' || PROCEDURE_INPUT || '); ' ||
   '-- "' || NAME || '"'
FROM
   systools.admin_task_list;


--------------------------------------------------------------------------------
0 18 * * *   call SYSPROC.ADMIN_CMD (VALUES('BACKUP DATABASE TESTDB ONLINE
                TO /database/backup WITHOUT PROMPTING')); -- "DAILY BACKUP"
0 15 * * *   call DBA.RUNSTATS (VALUES('S','SYSIBM')); -- "RUNSTATS"
```


XTIVIA

ATS Job Status

```
$ db2 "select NAME, TASKID, STATUS, BEGIN_TIME, END_TIME, SQLCODE \
         from SYSTOOLS.ADMIN_TASK_STATUS"

NAME            TASKID   STATUS     BEGIN_TIME                 END_TIME
--------------- -------- ---------- -------------------------- --------------
DAILY BACKUP          41 COMPLETE   2019-04-01-13.05.00.464598 2019-04-01-13.0
RUNSTATS              43 COMPLETE   2019-04-01-15.00.00.445019 2019-04-01-15.0

  2 record(s) selected.
```


XTIVIA

Db2 ATS Pros & Cons

Pros

- Enabled by default

Cons

- Limited notification options
- Jobs are only executed if database is activated

XTIVIA

Schedule cron jobs on DOCKER_HOST

- Execute scripts and commands
- Use `crontab` to schedule jobs
- Use docker command to execute scripts

- Creates container → host dependency
- Potential issue with shell access to Docker host

Schedule cron jobs on DOCKER_HOST

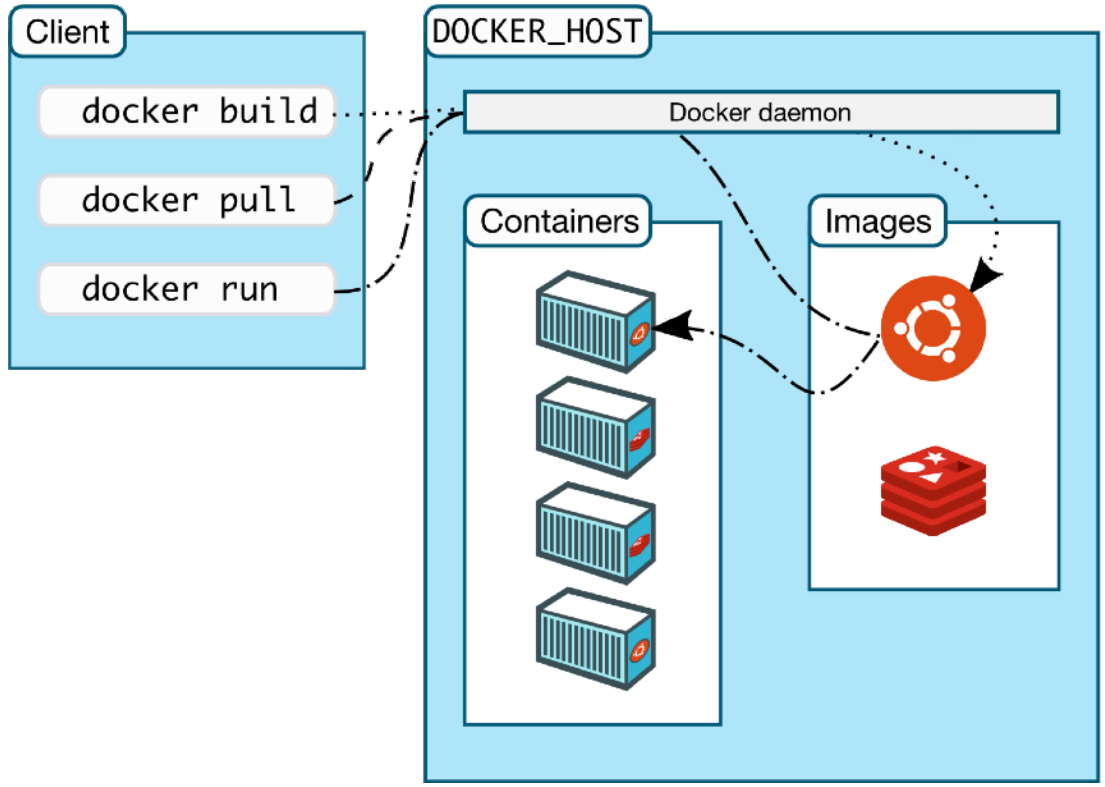
- Execute scripts and commands that exist in a container
- Use `docker exec`

```
/usr/bin/docker exec -it db2server_M4FP4 \
    su - db2inst1 -c "/database/config/db2inst1/runstats.sh -d TESTDB -s SYSIBM"
```

- Define a specific user for cron on host
- Create normal cron entries.

Use Docker CLI on remote host

- Docker is a client-server app
- Docker client can communicate with docker daemon via TCP/IP
- Designate special client machine for scheduling administrative tasks



XTIVIA

Enable Docker Daemon for Remote Access

```
$ sudo systemctl edit docker.service
```

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://192.168.174.12:2375
```

```
$ sudo systemctl daemon-reload

$ sudo systemctl restart docker.service
```

Unencrypted Channel

XTIVIA

Set up Remote Access for Docker Client

```
idbjorh@client $ docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED


idbjorh@client $ export DOCKER_HOST=192.168.174.12

idbjorh@client $ docker ps
CONTAINER ID        IMAGE                                                COMMAND
e0ed71dd1653        store/ibmcorp/db2_developer_c:11.1.3.3b-x86_64   "/var/db


idbjorh@client $ docker exec -it e0ed71dd1653 su - db2inst1
Last login: Tue Apr  2 01:56:52 UTC 2019
[db2inst1@db2server ~]$
```


XTIVIA

Cron Entries – Docker Client

```
idbjorh@client $ crontab -l
0 18 * * * DOCKER_HOST=192.168.174.12 /usr/bin/docker exec -it \
           db2server_M4FP4 \
           su - db2inst1 -c "/database/config/db2inst1/runstats.sh -d ..."

0 18 * * * DOCKER_HOST=192.168.174.14 /usr/bin/docker exec -it \
           db2server_M3FP3b \
           su - db2inst1 -c "/database/config/db2inst1/runstats.sh -d ..."
```


XTIVIA

Docker Automation Comparison

	Db2 ATS	Docker Host	Docker Client
Centralized Scheduling	❌	❌	✅
Use Shell Scripts	❌	✅	✅
Offline Backups	❌	✅	✅
Requires Host Access	❌	☑️	❌

XTIVIA

Agenda

✓Intro to Containers and Docker

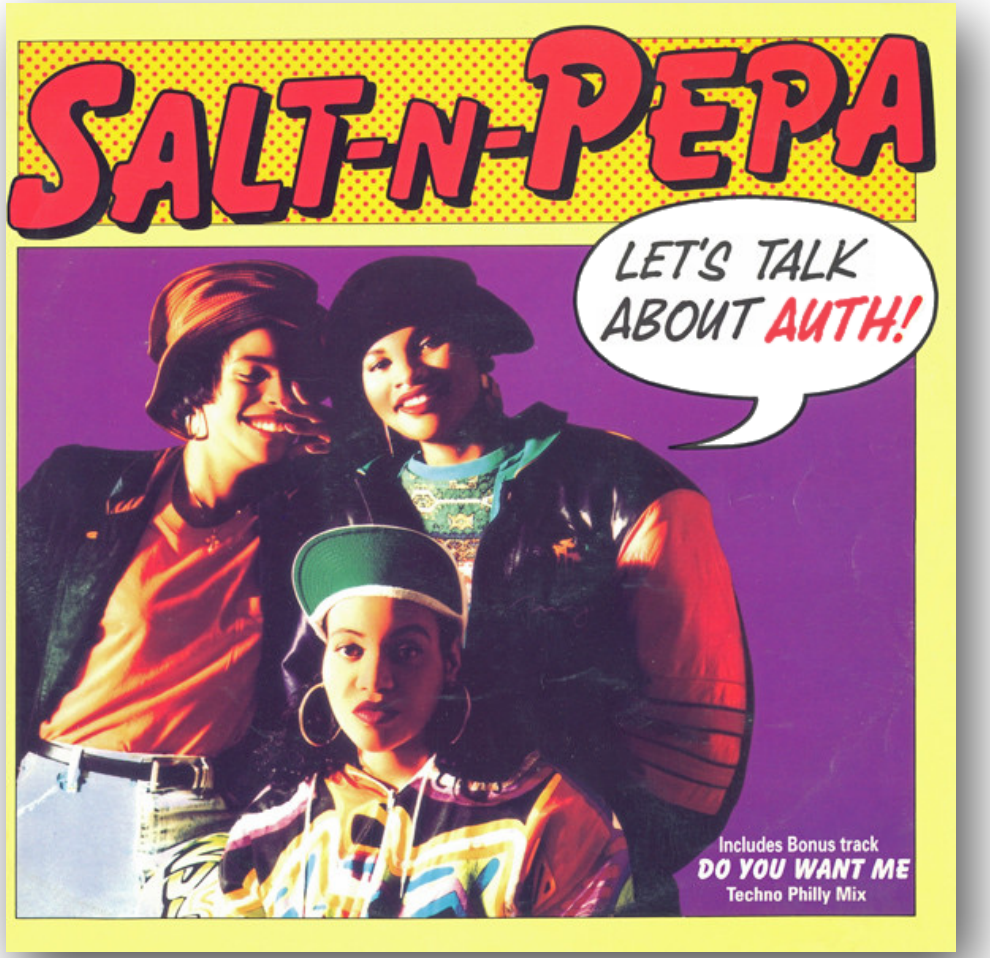
✓Recipe #1: Simplify Db2 Fixpacks & Upgrades

✓Recipe #2: Extending Existing Containers

✓Recipe #3: Automating Maintenance

- **Recipe #4: Handling Db2 Authentication**
- Questions

Recipe #4: Handling Db2 Authentication



XTIVIA

Default Users in your Db2 Container

- IBM's Db2 Developer-C Containers have 2 non-system users:

```
$ awk -F: '{ if ($3 > 500) { print $0 } }' /etc/passwd
db2inst1:x:1000:1000::/database/config/db2inst1:/bin/bash
db2fenc1:x:1001:1001::/database/config/db2fenc1:/bin/bash
```

XTIVIA

Authentication Options

- Add users manually to each *container*
  - Use `useradd` and `passwd`
- Define additional users in your *image*
  - Add `RUN` command(s) to your Dockerfile
  - Use `useradd` or use a custom script.
- Leverage external authentication (LDAP)

Adding Users to a Container

```
$ docker exec -it custom_M4FP4 useradd -M -N appuser

$ docker exec -it custom_M4FP4 passwd appuser
Changing password for user appuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```


XTIVIA

Adding Users to an Image

```
$ cat Dockerfile
FROM store/ibmcorp/db2_developer_c:11.1.4.4-x86_64

RUN  useradd -M -N appuser1 && \
     useradd -M -N appuser2 && \
     useradd -M -N appuser3

RUN  mkdir /var/custom

COPY db2_common_functions /var/db2_setup/include
COPY setup_db2_instance.sh /var/db2_setup/lib
COPY activate_dbs.sh /var/custom
```

XTIVIA

Leverage LDAP for Db2 Authentication

- Use the included LDAP security plugins:
  - `IBMLDAPauthserver` – User Authentication
  - `IBMLDAPgroups` – Group Membership

- What about Transparent LDAP?
  - Possible, but violates the principles for Docker containers

What about an LDAP Server?

- This *is* a Docker presentation!

```
$ docker pull osixia/openldap
Using default tag: latest
latest: Pulling from osixia/openldap
177e7ef0df69: Pull complete
2e9a343a17ed: Pull complete
47d97190f880: Pull complete
4410f2b3043e: Pull complete
71dfe666e2bd: Pull complete
0c1b8b11dfd7: Pull complete
f3478c8ba8bb: Pull complete
b223e9527c07: Pull complete
Digest: sha256:c2f52631643adde4212d776eb1f36ebb3dc8d69ac265a140ea6a5d0b5bf5
Status: Downloaded newer image for osixia/openldap:latest
```

Seeding LDAP Configuration - Organizational Units

```
$ cat ou.ldif
# Define Organizational Units
dn: ou=Users,dc=example,dc=org
objectClass: top
objectClass: organizationalUnit
ou: Users

dn: ou=Groups,dc=example,dc=org
objectClass: top
objectClass: organizationalUnit
ou: Groups
```


XTIVIA

Seeding LDAP Configuration – Users

```
$ cat users.ldif
# db2 users
dn:            cn=db2inst1,ou=Users,dc=example,dc=org
uid:           db2inst1
cn:            db2inst1
sn:            db2inst1
objectClass:   top
objectClass:   posixAccount
objectClass:   inetOrgPerson
uidNumber:     1000
gidNumber:     1000
loginShell:    /bin/bash
homeDirectory: /database/config/db2inst1
userPassword:  {SSHA}mUi1kby2pqmqM6a7XY/85lPwqGk7HeIo


dn:            cn=db2fenc1,ou=Users,dc=example,dc=org
```


XTIVIA

Seeding LDAP Configuration - Groups

```
$ cat groups.ldif
# db2 groups
dn:           cn=db2iadm1,ou=Groups,dc=example,dc=org
objectClass:  groupOfUniqueNames
cn:           db2iadm1
description:  Db2 SYSADM group
uniqueMember: cn=db2inst1,ou=Users,dc=example,dc=org
uniqueMember: cn=idbjorh,ou=Users,dc=example,dc=org

dn:           cn=db2fadm1,ou=Groups,dc=example,dc=org
objectClass:  groupOfUniqueNames
cn:           db2fadm1
description:  Db2 Fenced users group
uniqueMember: cn=db2fenc1,ou=Users,dc=example,dc=org
```


XTIVIA

Start OpenLDAP Container

```
$ cat ou.ldif users.ldif groups.ldif > db2.ldif

$ docker run --name ldap-service \
   --hostname ldap-service \
   --env LDAP_READONLY_USER="true" \
   --env LDAP_READONLY_USER_USERNAME="ldapquery" \
   --env LDAP_READONLY_USER_PASSWORD="ldap4db2" \
   -v $HOME/ldap/db2.ldif:\
/container/service/slapd/assets/config/bootstrap/ldif/50-bootstrap.ldif \
   --detach osixia/openldap --copy-service
c99faa2a0d74fefd7c07a174ca3db820815dab5f41fe3336306002966 9ac6fca
```


XTIVIA

Start Db2 Container – with Link to LDAP Container

```
$ docker run -h db2server \
    --name custom_M4FP4 \
    --link ldap-service:ldap-host \
    --detach \
    --privileged=true \
    -p 51000:50000 \
    -e TEXT_SEARCH=false \
    --env-file env-list \
    -v /home/idbjorh/db2fs2:/database \
    custom:11.1.4.4
80e48180ae32711669b125065abb5ac714d4c98072dc24bde2dc447a086b1155
```


XTIVIA

Configure Db2 to Talk with LDAP (1|2)

- Start with `sqllib/cfg/IBMLDAPSecurity.ini.sample`
- Modify `sqllib/cfg/IBMLDAPSecurity.ini`
  - `LDAP_HOST = ldap-host:389`
  - `USER_BASEDN = ou=Users,dc=example,dc=org`
  - `GROUP_OBJECTCLASS = groupOfUniqueNames`
  - `GROUP_BASEDN = ou=Groups,dc=example,dc=org`
  - `GROUP_LOOKUP_ATTRIBUTE  = uniqueMember`
  - `SEARCH_DN = cn=ldapquery,dc=example,dc=org`
  - `SEARCH_PW = ldap4db2`

XTIVIA

Configure Db2 to Talk with LDAP (2|2)

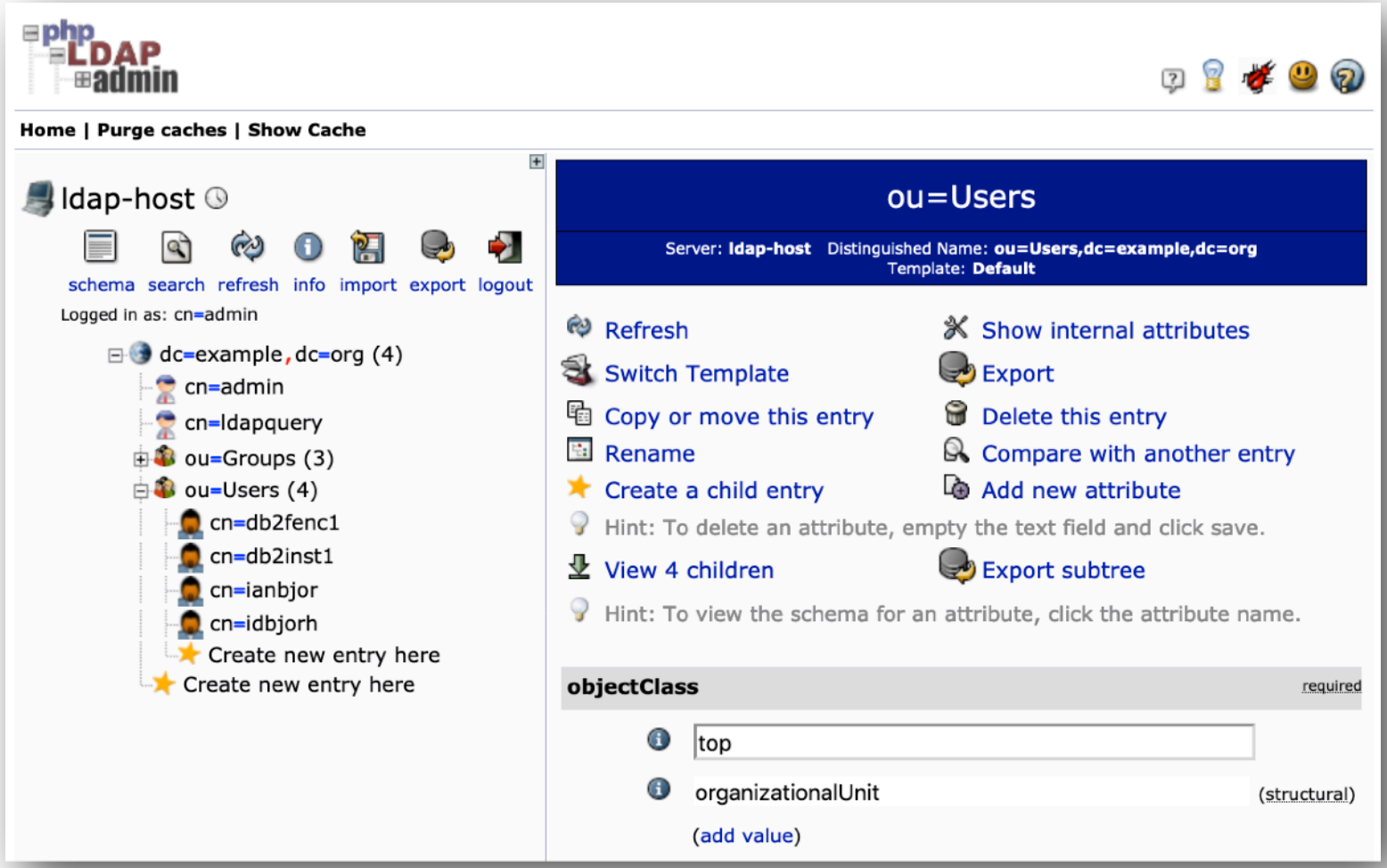
- Update Database Manager Configuration Parameters

```
$ db2 "update dbm cfg using \
       srvcon_pw_plugin IBMLDAPauthserver \
       group_plugin     IBMLDAPgroups"
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.

$ db2stop ; db2start
04/27/2019 09:45:38     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
04/27/2019 09:45:40     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
```


XTIVIA

Creating additional LDAP Users (1|3)

- Use *phpLDAPadmin*
  - Available as Docker container: osixia/phpldapadmin



XTIVIA

Creating additional LDAP Users (2|3)

- Commands: Use an LDIF file

```
$ cat user.ldif
dn:           cn=rbobby,ou=Users,dc=example,dc=org
uid:          rbobby
cn:           rbobby
cn:           Ricky Bobby
sn:           rbobby
objectClass:  top
objectClass:  inetOrgPerson
```

Creating additional LDAP Users (3|3)

- Use `slappasswd` to generate default password for LDIF file
- Use `ldapadd` to add users to LDAP server

```
$ docker exec ldap-service slappasswd -s mypassword | \
    awk '{print "userPassword: " $0 }' | tee -a user.ldif
userPassword: {SSHA}2HyGhWn9PxGUuXsrDuh582PXfx/8JDCP

$ docker cp user.ldif ldap-service:/tmp/

$ docker exec ldap-service ldapadd -x -H ldap://localhost:389 \
    -D "cn=admin,dc=example,dc=org" -w admin -f /tmp/user.ldif
adding new entry "cn=rbobby,ou=Users,dc=example,dc=org"
```

Agenda

✓Intro to Containers and Docker

✓Recipe #1: Simplify Db2 Fixpacks & Upgrades

✓Recipe #2: Extending Existing Containers

✓Recipe #3: Automating Maintenance

✓Recipe #4: Handling Db2 Authentication

- **Questions**

XTIVIA

Speaker: Ian Bjorhovde

Email Address: ibjorhovde@xtivia.com
Phone: 719-309-1127

XTIVIA