IBM WW Z Security Conference

October 6-9, 2020

z/OS Container Extensions
LDAP ITDS on z/OS

(2-parts session)

Part-2

- **Add Native Authentication**
- **Add TLS support**
- **Add MFA factor for multifactor authentication support**

Philippe RICHARD

IBM Systems LBS

Philippe_Richard@fr.ibm.com

Objectives

- This presentation describes how we set up and configured a zOS LDAP server (IBM Tivoli Directory Server) for zOS Container Extensions (**zCX**) and other open source applications (**DPP**, **docker**, …).
- **Part 1**
  - zCX user management and authentication
  - Configure zCX for LDBM backend ldap support
- **Part 2 (with demo)**
  - Add Native Authentication
  - Add TLS support
  - Add MFA factor for multifactor authentication
- **Goals for our solution**
  - Use zOS LDAP server with zOS Container Extensions (zCX) to provide centralized user/group management.
  - Use zOS LDAP server with other Open Source applications within our zCX environments.
  - Configure for RACF password/passphrase authentication
  - Enable multifactor authentication with MFA for z/OS
- **Docker and Open Source application/users should run transparently with zOS LDAP and leverage RACF authentication**

Part 2 (with demo)

- Add Native Authentication
- Add TLS support
- Add MFA factor for multifactor authentication
- LDAP misc.
  - Monitor and audit LDAP logins
  - Change LDAP passwords

zCX - Adding zOS LDAP Native Authentication support

- The z/OS LDAP server has the ability to authenticate to the Security Server through the LDBM or TDBM backends.
- The password or password phrase is stored in RACF. Verification is performed by the Security Server.
- The LDAP entry should contain either the **ibm-nativeId** or **uid** attribute to specify the Security Server ID that is associated with this entry.
- Configuration options that come into play are:
  - LDAP tree structure for zCX users.
    - Where users are defined
    - Which sub-trees are marked for native authentication.
  - zOS LDAP server DSCONFIG options for LDBM or TDBM, including:
    - nativeAuthSubtree
    - useNativeAuth
  - Use of ibm-nativeId attribute and/or the uid attribute in the LDBM entry

/etc/ldap/ldbm0/ds.config

nativeAuthSubtree "ou=zcx_users_racf,o=zCX,o=ibmmop,c=fr"
useNativeAuth: all

RACF requirements

- The RACF userid must already exist. LDAP will not create one.
- The RACF userid specified by either the **uid** or **ibm-nativeId** attributes must contain a valid OMVS segment with an OMVS UID value in RACF.
  - However, the OMVS UID value has no relation to the uid attribute defined in the posixAccount.
- While the password/passphrase will be compared against the one in RACF, the user will still need the posixAccount defined with the **uid** and various other attributes.
- If both the **uid** and **ibm-nativeID** are present in the entry, the **ibm-nativeID** will be used for authentication.
- Note: the SDBM backend is NOT required for this to work, the RACF database colocated on the same LPAR as the ldap server will be used for credentials validation (password/passphrase)

‘native’ id

nativeAuthSubtree

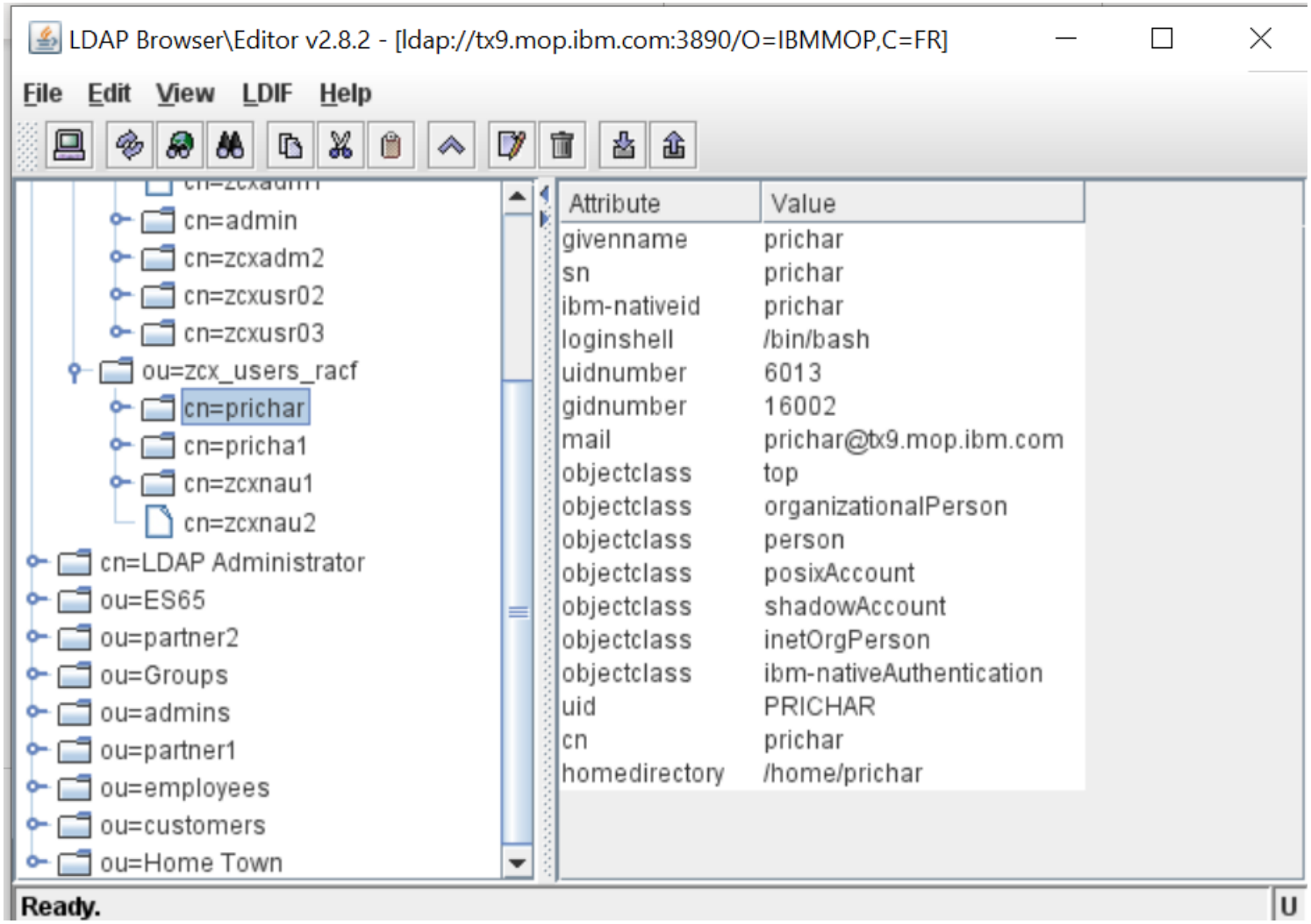
dn: cn=prichar,**ou=zcx_users_racf,o=zCX,o=ibmmop,c=fr**
objectclass: top
objectclass: organizationalPerson
objectclass: person
objectclass: posixAccount
objectclass: shadowAccount
objectclass: inetOrgPerson
objectclass: ibm-nativeAuthentication
cn: prichar
gidnumber: 16002
homedirectory: /home/prichar
sn: prichar
uid: prichar
ibm-nativeId: prichar
uidnumber: 6013
givenname: prichar
loginshell: /bin/bash
mail: prichar@tx9.mop.ibm.com

Add Native Authentication - Procedure

- add the new sub-tree for Native Authentication users

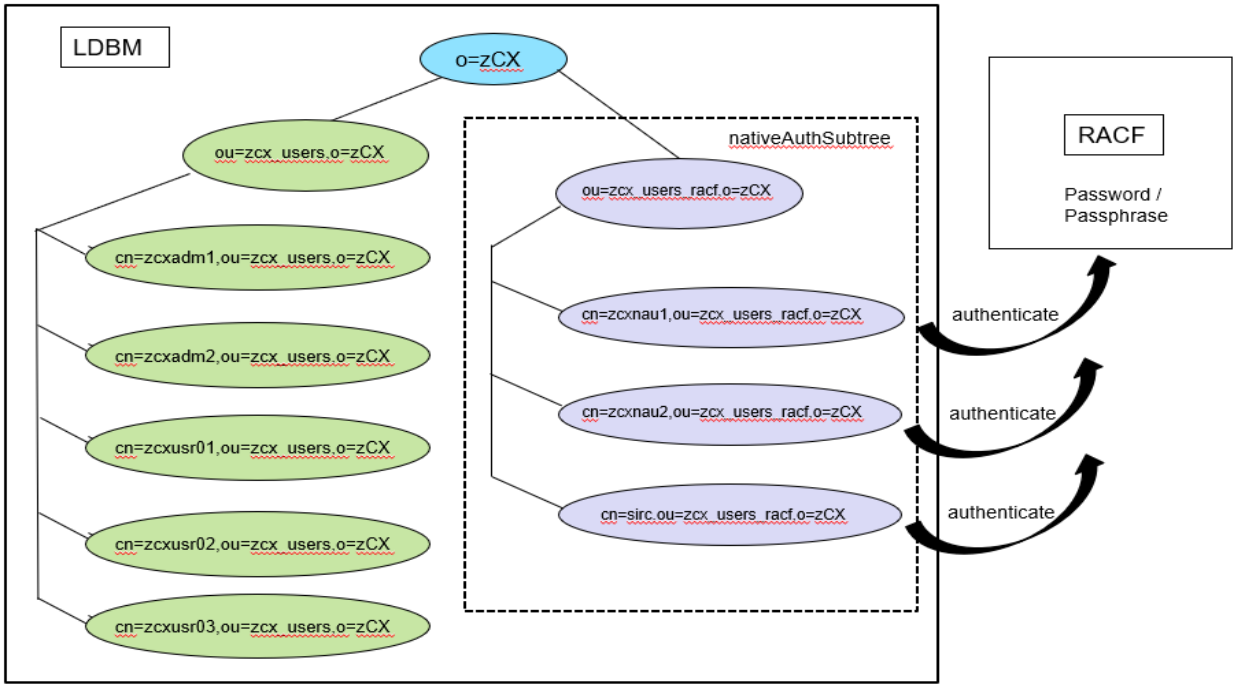
```
//S1       EXEC PGM=IKJEFT1B
//SYSTSPRT DD SYSOUT=*
//SYSEXEC  DD DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSTERM  DD DUMMY
//SYSTSIN  DD *
oshell +
ldapadd    -h 127.0.0.1 -p 3890 +
-D "cn=admin" -w xxxxxx +
-f /u/prichar/ES65/zCX_NativeAuth.ldif
/*
```

Nativeauth subtree: ou=zcx_users_racf,o=zCX,o=ibmmop,c=fr



Nativeauth passwords

- All users in the ***ou=zcx_users,o=zCX*** sub-tree have the passwords within LDBM backend.
- ***ou=zcx_users_racf,o=zCX*** sub-tree is configured for Native Authentication. All of the users in that sub-tree are authenticated by RACF (their password/passphrases are stored in RACF database).



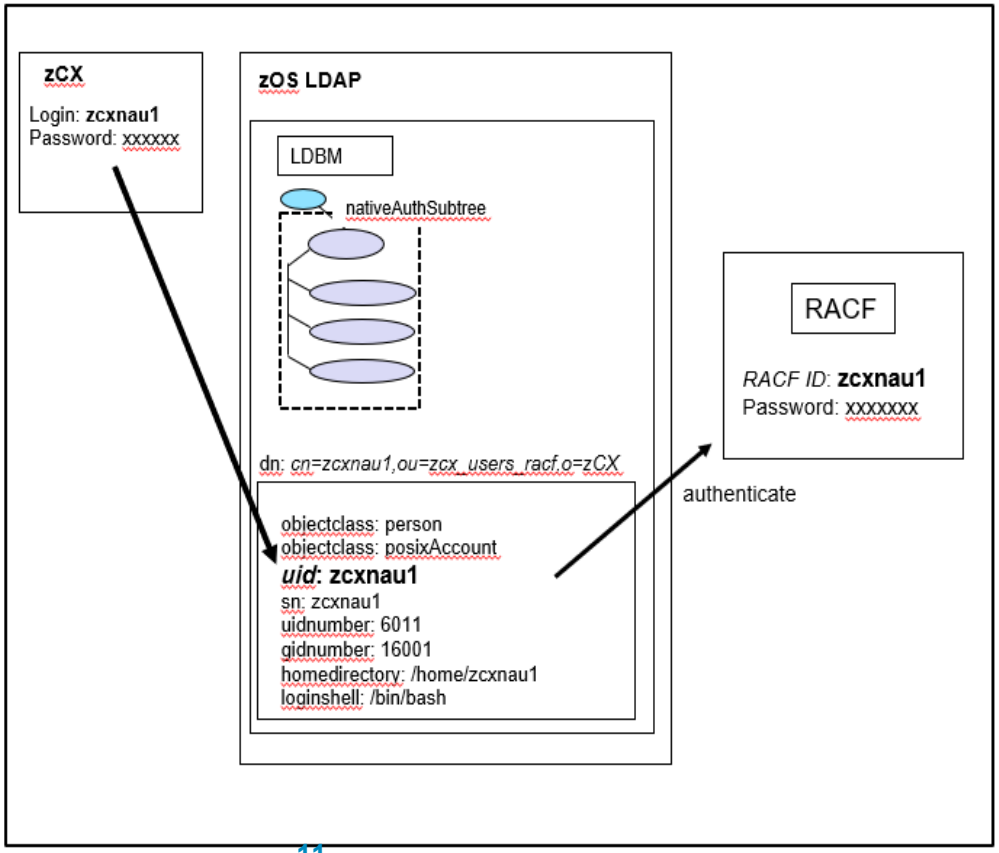
ssh logon with Nativeauth

- You should now be able to successfully login into the SSH CLI using either uid prichar zcxnau1 or zcxnau2, both using the password/passphrase defined in RACF.
- No changes are needed in the zCX Appliance instance setup.

- Notes:
  - Since "zcxnau1" does not have an "ibm-nativeId" attribute, the uid ("zcxnau1") is passed to RACF along with the password/passphrase for verification.
  - You will login to zCX using the "zcxnau2" id, the user is associated to RACF userid "zcxsafu2".
  - For our setups, there is no "zcxnau2" userid defined in RACF.

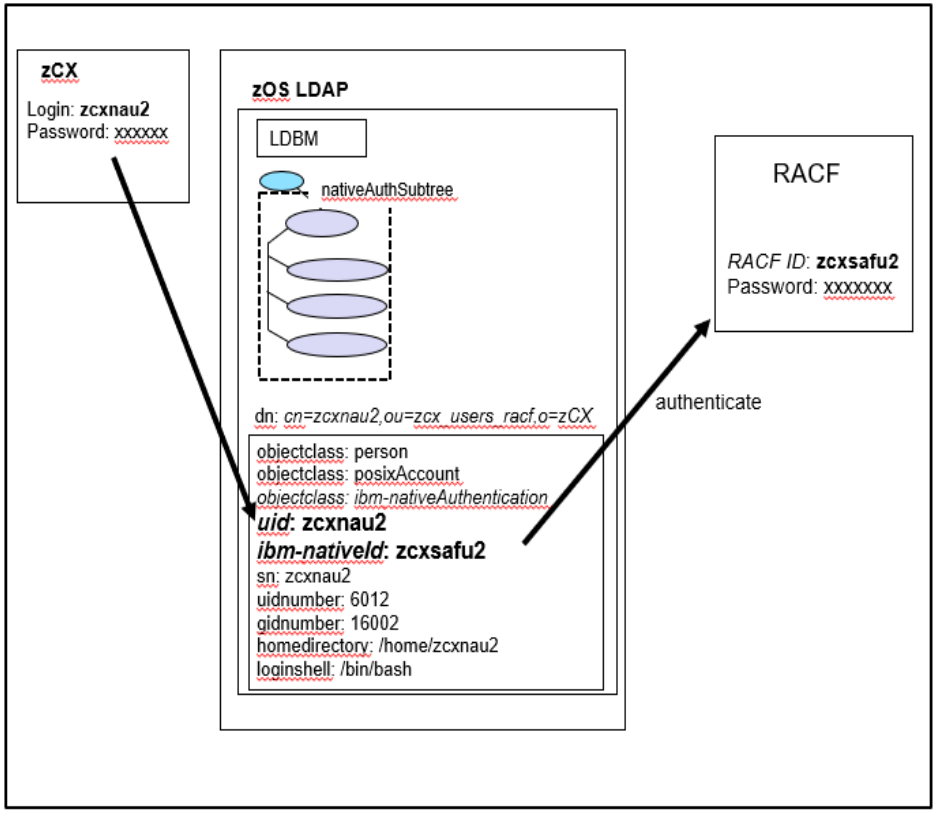
zOS LDAP – Native Authentication – uid attribute only

- zCX **uid *zcxnau1*** (dn: *cn=zcxnau1,ou=zcx_users_racf,o=zCX*) does not contain an **ibm-nativeId** attribute (does not include the **ibm-nativeAuthentication** objectClass), only the **uid**.
- The **uid** (***zcxnau1***) is used as the RACF ID.



zOS LDAP – Native Authentication – uid attribute and ibm-nativeId attribute

- zCX uid ***zcxnau2*** (dn: *cn=zcxnau2,ou=zcx_users_racf,o=zCX*) DOES contain an **ibm-nativeId** attribute (includes the **ibm-nativeAuthentication** objectClass).
- The uid (***zcxnau2***) is entered in the zCX Login, but the **ibm-nativeId** (***zcxsafu2***) is used to authenticate with RACF.
- (The **uid** and the **ibm-nativeId** can be the same value.)



ssh with native authentication

```
AzureAD+philippeRichard@LAPTOP-QP04D215 MINGW64 ~
$ ssh -p 8022  zcxnau2@9.212.128.231
zcxnau2@9.212.128.231's password:
Creating directory '/home/zcxnau2'.

Welcome to the IBM z/OS Container Extensions (IBM zCX) shell that provides access to
Docker commands.
For more information on how to use this shell to execute Docker commands refer to IBM
zcxnau2@TX9:~$ id
uid=6012(zcxnau2) gid=16002(zcx_group2)
groups=16002(zcx_group2),109(docker),16001(zcx_group1)
zcxnau2@TX9:~$
```

RACF pasword of zcxsafu2

```
lu zcxnau2
ICH30001I UNABLE TO LOCATE USER    ENTRY ZCXNAU2

Lu zcxsaf2
 USER=ZCXSAFU2  NAME=ZCX IBM-NATIVEID      OWNER=MVS       CREATED=20.121
  DEFAULT-GROUP=OMVSGRP  PASSDATE=20.121 PASS-INTERVAL= 60 PHRASEDATE=N/A
  ATTRIBUTES=OPERATIONS
  REVOKE DATE=NONE   RESUME DATE=NONE
  LAST-ACCESS=20.136/15:51:17
```

Ldap added TLS/SSL security:

- You have to implement TLS/SSL for your ldap server to encrypt the session when passing the passwords
- This is B.A.U
  - Use RACF keyrings and certificates for the ldap server
  - Or openssl to create the certificate and gskkyman to import it in a key database for use by your ldap server
- Update ldap environment variables (DSENVVAR)
  - Review the DSENVVAR file to ensure that the following values are added/set correctly:

```
GSK_PROTOCOL_TLSV1=ON
GSK_PROTOCOL_TLSV1_1=ON
GSK_PROTOCOL_TLSV1_2=ON
```

- Update ldap config file (DSCONFIG)

```
Listen ldaps://9.100.200.300:6890
sslAuth serverAuth
sslKeyRingFile /zcx/ldap/zcxldap.kdb
sslKeyRingPWStashFile
/zcx/ldap/zcxldap.sth
```

```
listen ldaps://:6890
listen ldap://:pc
#sslAuth serverClientAuth
sslAuth serverAuth
#sslCipherspecs any
#sslCipherspecs 12288
# sslCipherspec TRIPLE_DES_SHA_US+DES_SHA_US
sslKeyRingFile OMVSUSR/LDAPSRV_RACF_KEYRING
```

Update zCX instance to use secure communications to zOS LDAP server

- Run a reconfiguration workflow for zCX instance to update the ldap.conf file
  - Update the input ldap config file used to provision the zCX instance:

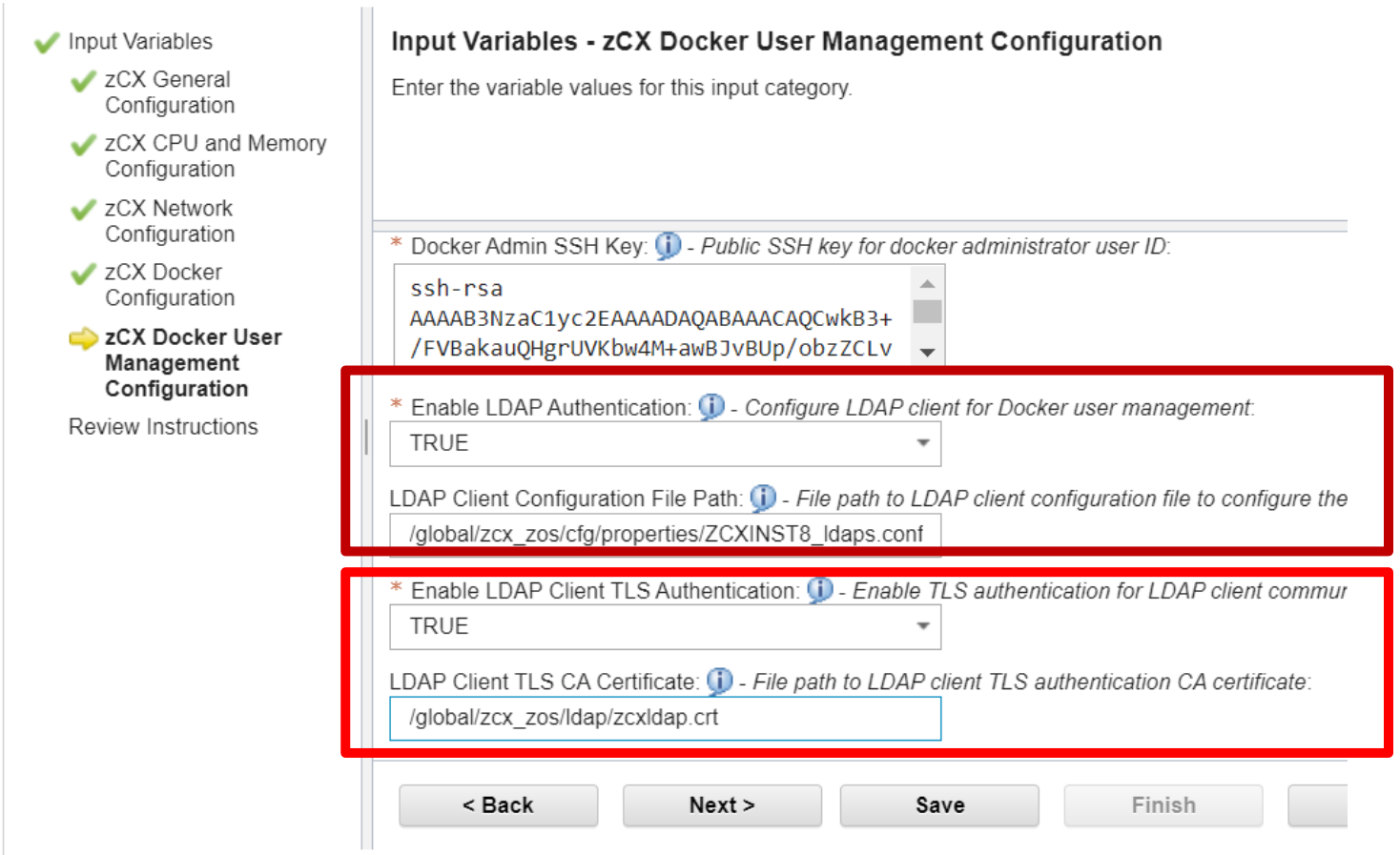

(care: not the ldap config file !)

/global/zcx_zos/cfg/properties/ZCXINST8_ldaps.conf

```
# Set uri to ldaps protocol and specify the zOS LDAP server's secure
port
# uri ldap://9.100.200.300:3890
uri ldaps:// 9.100.200.300:6890
 # set ssl on
ssl on
# The file that contains certificates for all of the certificate authorities
# During provisioning, zCX copies the certificate file to the directory
# /etc/ldap   and renames
# it to ldap-ca.crt. Therefore, the TLS_CACERT setting in the LDAP
 # configuration file  should be /etc/ldap/ldap-ca.crt.
tls_cacertfile /etc/ldap/ldap-ca.crt
```

Run a reconfiguration workflow for

- Run a reconfiguration workflow for zCX instance to update the ldap.conf file



User management reconfiguration

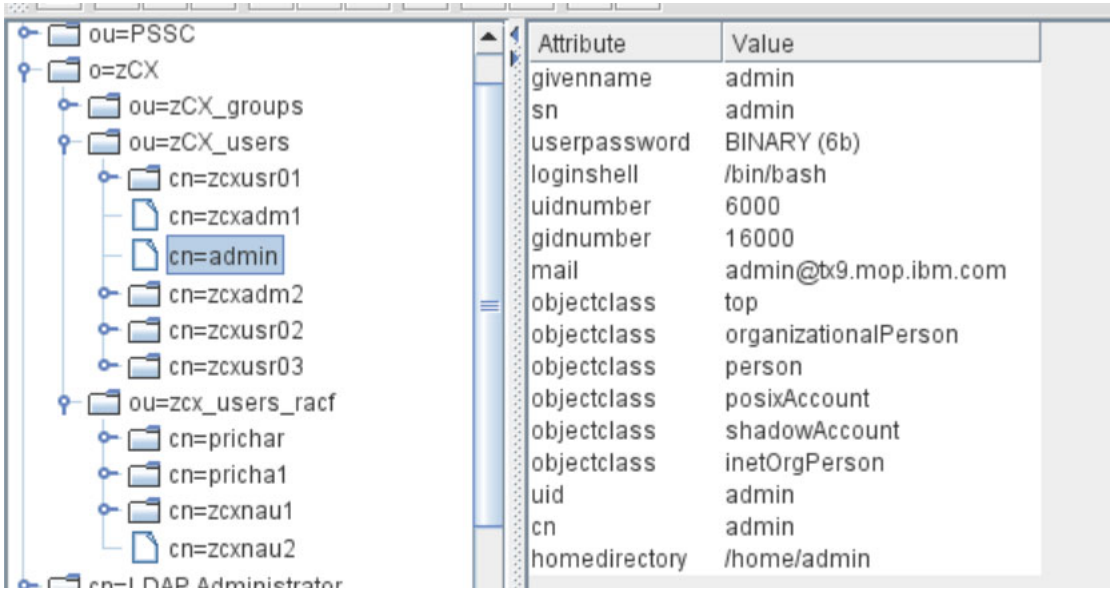
- You can change user management control between local and LDAP-based user management or from one LDAP server to another using the “Reconfiguration workflow”
- **Switching from local user management to LDAP-based user management**
  - The login access of the default administrator user will be removed when switching from local to LDAP based user management because the local administrator role and capabilities are not needed to manage users centrally.
  - **No sudo capability when ldap is enabled**
  - However, the home directory of the administrator will remain intact.
- If you want preserve the access and data of additional users and groups (including the default administrator), you should recreate them in the LDAP server with the same attributes they had locally.
  - (LDAP groups should have the same names and group IDs. LDAP users should have the same user names, LDAP *uidNumber*s corresponding to local *uid*s, primary group assignments, and home directory settings.)

Recreate the zCX ‘admin’ user in ldap

```
# ----  Add userids to zCX Group1  ----

dn: cn=zcx_admins,ou=zcx_groups,o=zCX,o=ibmmop,c=fr
changetype: modify
add: memberUid
memberUid: admin


dn: cn=admin,ou=zCX_users,o=zCX,o=ibmmop,c=fr
objectclass: top
objectclass: organizationalPerson
objectclass: person
objectclass: posixAccount
objectclass: shadowAccount
objectclass: inetOrgPerson
cn: admin
gidnumber: 16000
homedirectory: /home/admin
sn: admin
uid: admin
uidnumber: 6000
givenname: admin
loginshell: /bin/bash
mail: admin@tx9.mop.ibm.com
userpassword: secret
```



Docker users

- Local user management:
  - create users in group docker with command:

  sudo adduser --ingroup docker local-user
- LDAP users authorized to login to a zCX appliance will be part of the local docker group automatically.
  - No need for an admin user to manually add LDAP users to local docker users, for them to run docker commands.

before logon of zcxusr02 (command issued under user zcxadm1)

```
zcxadm1@TX9:~$ id zcxusr02
uid=7002(zcxusr02) gid=16002(zcx_group2)
groups=16002(zcx_group2),16001(zcx_group1)
```

after logon of zcxusr02:

```
zcxadm1@TX9:~$ su zcxusr02
Password:
Creating directory '/home/zcxusr02'.
zcxusr02@TX9:/home/zcxadm1$ id
uid=7002(zcxusr02) gid=16002(zcx_group2)
groups=16002(zcx_group2),109(docker),16001(zcx_group1)
```

MFA

- A multi-factor authentication system requires that multiple authentication factors be presented during logon to verify a user's identity. Each authentication factor must be from a separate category of credential types:
  - Something you know: A password / passphrase or security question
  - Something you have: An ID badge or cryptographic token device
  - Something you are: Fingerprint or other biometric data
- MFA deals with three components: What you know, What you have and What you are. “What you are” or “What you have” with traditional passwords can provide assurance that the right person is accessing critical systems.
- By requiring multiple authentication factors, a user's account can not be compromised if one of their factors is discovered.
- With Z MFA V2.1, z/OS sign ons are protected with a variety of factors. These include
  - Yubikey support, RSA SecurID ®, Gemalto SafeNet, IBM Security Access Manager (ISAM), IBM Cloud Identity Verify (CIV) via RADIUS, Generic RADP, LDAP, Native timed one time password (**TOTP**), and SmartCard usage.
  - Several of these factors include biometric support.

Enable multifactor authentication

- Add an MFA factor for a native-id RACF user
- Ex: IBM TouchToken AZFTOTP1

```
/* Enter the following command to set the IBM TouchToken registration*/
/* state for the user to OPEN. (Case is sensitive for OPEN.) */
ALU ZCXNAU1 MFA(FACTOR(AZFTOTP1) +
 TAGS(REGSTATE:OPEN))
ALU ZCXNAU1  MFA(FACTOR(AZFTOTP1) TAGS(ALG:SHA1
NUMDIGITS:6 PERIOD:30))
```

```
LISTUSER ZCXNAU1  MFA
```

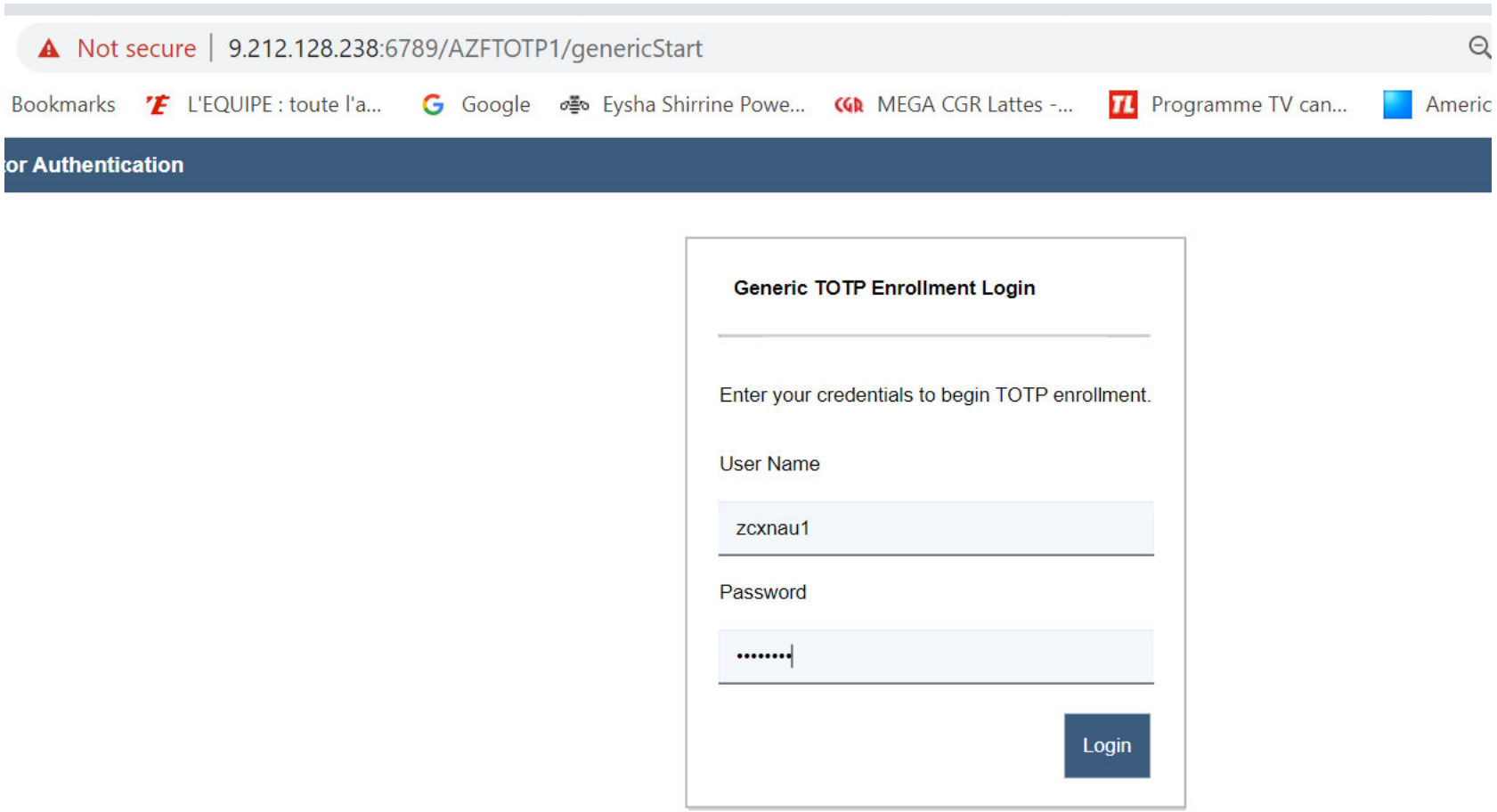
```
 LU ZCXNAU1 MFA

MULTIFACTOR AUTHENTICATION INFORMATION:
---------------------------------------
  PASSWORD FALLBACK IS ALLOWED
  FACTOR = AZFTOTP1
    STATUS = INACTIVE
    FACTOR TAGS =
      REGSTATE:OPEN
```

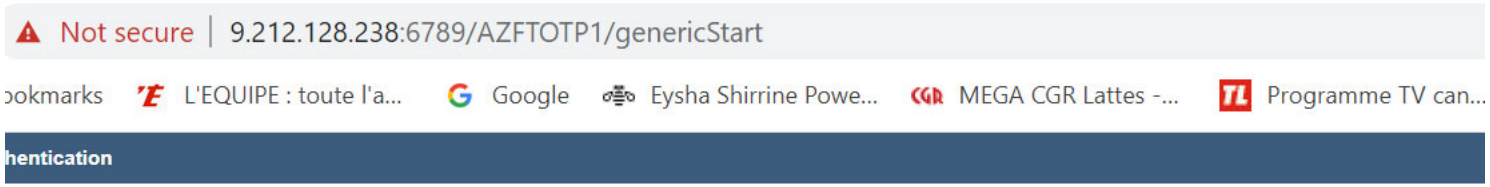
- Register your device
  - open the generic TOTP start page in a desktop web browse

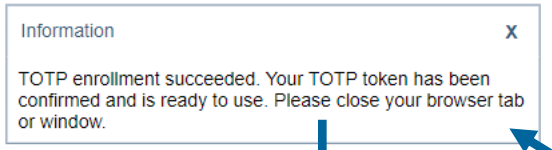
https://*hostname:6789*/AZFTOTP1/genericStart

MFA register your device for TOTP1 touchtoken



Use an App to scan the QR code to register (FreeOTP, Google Auth, IBM Verify,…)



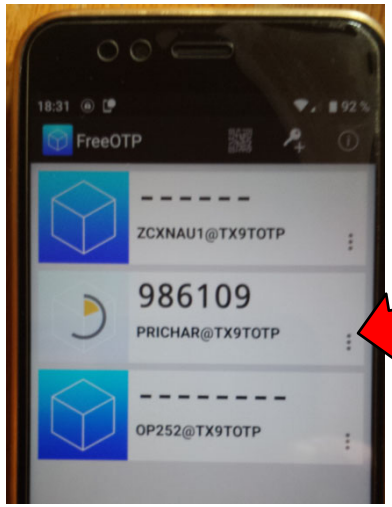


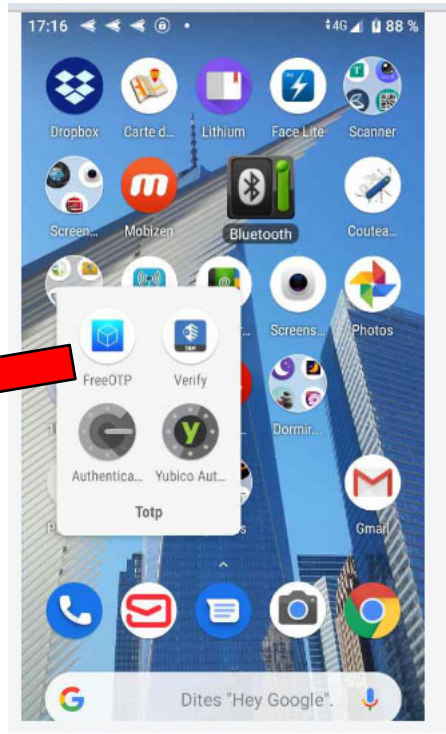
LISTUSER ZCXNAU1 MFA

```
MULTIFACTOR AUTHENTICATION INFORMATION:
---------------------------------------
  PASSWORD FALLBACK IS ALLOWED
  AUTHENTICATION POLICIES =
    CERTONLY
  FACTOR = AZFTOTP1
    STATUS = ACTIVE
FACTOR TAGS =
    REGSTATE:PROVISIONED
    ALG:SHA1
    NUMDIGITS:6
    PERIOD:30
    KEYLABEL:AZF.ZCXNAU1.D7DDE16DA9240B03
    CVALUE:52985141
    WINDOW:10
```

Generate a token

- Use any token generator app that you have used to register your device
  - FreeOTP or GoogleAuthenticator or IBM Verify
- Tap the selected entry to generate the TOTP token





- TOTP1 MFA factor is configured with:

```
Compound In-band Authentication
  Enable. . . . . .  Y  ( N or Y )
  Credential Order. . . . .  1  1. MFA Credential First
                                2. RACF Credential First
```

  - Then logon to zCX with a credentail in the form of:
    Token:racf-password

Logon to zCX with MFA

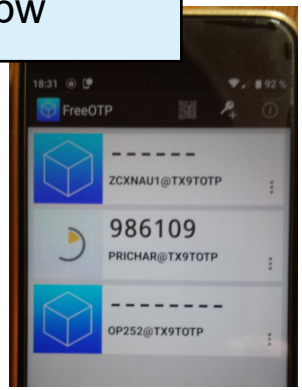
- Then logon to zCX with a multifactor credential in the form of:
  - " Something you have": a cryptographic token device authentication (TOTP token)
  - "Something you know." (The RACF password.)

```
AzureAD+philippeRichard@LAPTOP-QP04D215 MINGW64 ~
$ ssh -p 8022  zcxnau1@9.212.128.231
zcxnau1@9.212.128.231's password:
Permission denied, please try again.
zcxnau1@9.212.128.231's password:

Welcome to the IBM z/OS Container Extensions (IBM zCX) shell
that provides access to Docker commands.
For more information on how to use this shell to execute Docker
commands refer to IBM
Last login: Fri May 15 15:25:22 2020 from 9.145.162.117
```

RACF pasword-> fails !

986109:racfpw



Auditing ldap logins

- dsconfig:
  - audit error,modify+search+bind
  - audit all,add
  - audit on
- When auditing is on, an LDAP SMF type 83 subtype 3 audit record is generated

```
<event>
<eventType>*BIND</eventType>
<eventQual>FAILURE</eventQual>
<timeWritten>15:59:39.98</timeWritten>
<dateWritten>2020-05-29</dateWritten>
<systemSmfid>X9</systemSmfid>
<prodName>LDAP</prodName>
<details>
<violation>Y</violation>
…
<jobName>LDAPLD0</jobName>
…
<logRauditx>Y</logRauditx>
<d:reqTimestp>38</d:reqTimestp>
<d:serverUrl>ldaps://9.212.128.238:6890</d:ser
verUrl>
…
```

```
<d:eventCode>2</d:eventCode>
<d:mapcertOpt>00</d:mapcertOpt>
<d:flags>0x00000000</d:flags>
<d:protocolVer>V3</d:protocolVer>
<d:returnCode>0000000049</d:returnCode>
<d:errorMsg>
R004062 Credentials are not valid
(process_simple_bind:1268)
</d:errorMsg>
<d:entryNm>cn=zcxadm2,ou=zCX_users,o=
zCX,o=ibmmop,c=fr</d:entryNm>
<d:relocate205>Fri May 29 15:59:39
2020</d:relocate205>
<d:bindMech>SIMPLE</d:bindMech>
<d:policyUpdated>F</d:policyUpdated>
</details>
</event>
```

ldap misc:

- **Simple bind (**ldap_sasl_bind): ldapsearch -H ldap://9.212.128.238:3890 -x -b "o=zcx,o=ibmmop,c=fr" -D "cn=admin,ou=zcx_users,o=zcx,o=ibmmop,c=fr" -w xxxxxx

- **SSL/TLS**: LDAPTLS_CACERT=/etc/ldap/ldap-ca.crt ldapsearch -H ldaps://9.212.128.238:6890 -x -b " cn=admin,ou=zcx_users,o=zcx,o=ibmmop,c=fr " -D "cn=admin,ou=zcx_ users,o=zcx,o=ibmmop,c=fr" -w xxxxxx

```
# extended LDIF
#
# LDAPv3
# base < cn=admin,ou=zcx_users,o=zcx,o=ibmmop,c=fr > with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# admin, zCX_users, zCX, ibmmop, fr
dn: cn=admin,ou=zCX_users,o=zCX,o=ibmmop,c=fr
objectclass: top
objectclass: organizationalPerson
objectclass: person
objectclass: posixAccount
objectclass: shadowAccount
objectclass: inetOrgPerson
cn: admin
gidnumber: 16000
homedirectory: /home/admin
sn: admin
uid: admin
uidnumber: 6000
givenname: admin
loginshell: /bin/bash
mail: admin@tx9.mop.ibm.com
userpassword:: c2VjcmV0

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

ldap « own password » change

LDAPTLS_CACERT=/etc/ldap/ldap-ca.crt ldappasswd -H ldaps://9.212.128.238:6890 -x -D "cn=admin,ou=zcx_users,o=zcx,o=ibmmop,c=fr" -W -S -d 1
<R006010 Unsupported extended operation '1.3.6.1.4.1.4203.1.11.1'
(srv_process_extended_request:946

- **Change password (regular LDBM users under ou=zcx_users,o=zcx,o=ibmmop,c=fr)**

ldapmodify -H ldap://9.212.128.238:3890 -D "cn=admin,ou=zcx_users,o=zcx,o=ibmmop,c=fr" -w **secret** -f /home/admin/ldappwchg.ldif

```
dn: cn=admin,ou=zcx_users,o=zcx,o=ibmmop,c=fr
changetype: modify
replace: userPassword
userPassword: new1pass
-
```

- **Change password (RACF) with bind: (nativeauth users under ou=zcx_users_racf,o=zcx,o=ibmmop,c=fr)**
  - modify RACF password wih bind

```
ldapsearch -H ldap://9.212.128.238:3890 -V 3 -s base -D
racfid=zcxnau1,profiletype=user,sysplex=tx -w oldpwd/newpwd -b
"racfid=zcxnau1,profiletype=user,sysplex=tx" "objectclass=*"
```

Requires that the ldap server also has an SDBM backend !

Wrap-up: Why LDAP user management on z/OS ?

- Centralized users credentials management by z/OS security team
- Single source of credentials accross all platforms (z/OS, distributed)
- Compatibility with Open source RFC 2307,an approach for Using LDAP as a Network Information Service (https://tools.ietf.org/html/rfc2307)
- RACF proven security infrastructure
  - Encrypted password/passphrase with AES256
- z/OS LDAP ITDS leverages z/OS qualities of service (RAS Reliability and Availability)
  - An LDAP server running in a sysplex environment supports multiple instances of the same server within a cross-system coupling facility group.
  - WLM LDAP transaction priority management
- Enable multifactor authentication with MFA for z/OS for stronger multi-factor authentication

Summary

- We showed you how we set up and configured a zOS LDAP server (IBM Tivoli Directory Server) for zOS Container Extensions (**zCX**) and other open source applications (**DPP**, **docker**, …).
- **Part 1**
  - zCX user management and authentication
  - Configure zCX for LDBM backend ldap support
- **Part 2 (with demo)**
  - Add Native Authentication
  - Add TLS support
  - Add MFA factor for multifactor authentication
- **What we achieved:**
  - Use zOS LDAP server with zOS Container Extensions (zCX) to provide centralized user/group management.
  - Use zOS LDAP server with other Open Source applications within our zCX environments.
  - Configure for RACF password/passphrase authentication
  - Enable multifactor authentication with MFA for z/OS
- **Any Docker and Open Source application/users should run transparently with zOS LDAP and leverage RACF and MFA authentication**

zCX – Documentation Links

Modernize and Extend your z/OS® Applications with IBM z/OS® Container Extensions(zCX)

Resource	Link
Content Solutions Page	http://ibm.biz/zOSContainerExtensions
Open Z Systems Exchange	http://ibm.biz/openzsx
zCX FAQ	http://ibm.biz/zcx_FAQ
z/OS 2.4 Knowledge Center	https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.izso100/abstract.htm

Google Docker: Lots of excellent resources, documentation and self-paced learning courses are available

Getting Started videos:

Resource Planning for zCX:
https://www.youtube.com/watch?v=5o1r2EPMMUc

Provisioning zCX using z/OSMF workflows:
https://www.youtube.com/watch?v=CPeI5KmoAw0

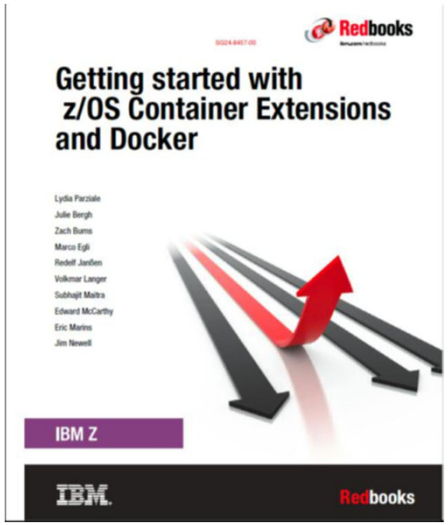
Getting started with Docker in zCX:
https://www.youtube.com/watch?v=9aYFzhvJVb

An Overview of IBM z/OS Container Extensions:
https://youtu.be/W0akd6fCHtE

zCX – IBM Redbook

Redbooks
ibm.com/redbooks

Getting started with z/OS Container Extensions and Docker



Available at:
http://www.redbooks.ibm.com/abstracts/sg248457.html

Chapters

1. Introduction
2. z/OS Container Extensions Planning
3. Security – Overview
4. Provisioning and managing your first z/OS Container
5. Your first running Docker container in zCX
6. Private Registry Implementation
7. Operation
8. Integrating container applications with other processes on z/OS
9. zCX User administration
10. Persistent data
11. Swarm on zCX