



IBM WW Z Security Conference

October 6-9, 2020

z/OS Container Extensions LDAP ITDS on z/OS

Part-1

**Configuring zCX with ldap on z/OS
(ITDS) with RACF and MFA support
(2-parts session)**

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 (On demand, 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**

Preamble

- Special thanks to:
 - Thomas Sirc, z/OS Integration Test – Networking, IBM Systems [Poughkeepsie, NY](#) for his work on creating the the schema changes and documenting his experience with zCX and ldap z/OS
 - Yuksel Gunal, zCX Development, IBM Systems, [Poughkeepsie, NY](#) for his continuous help and support

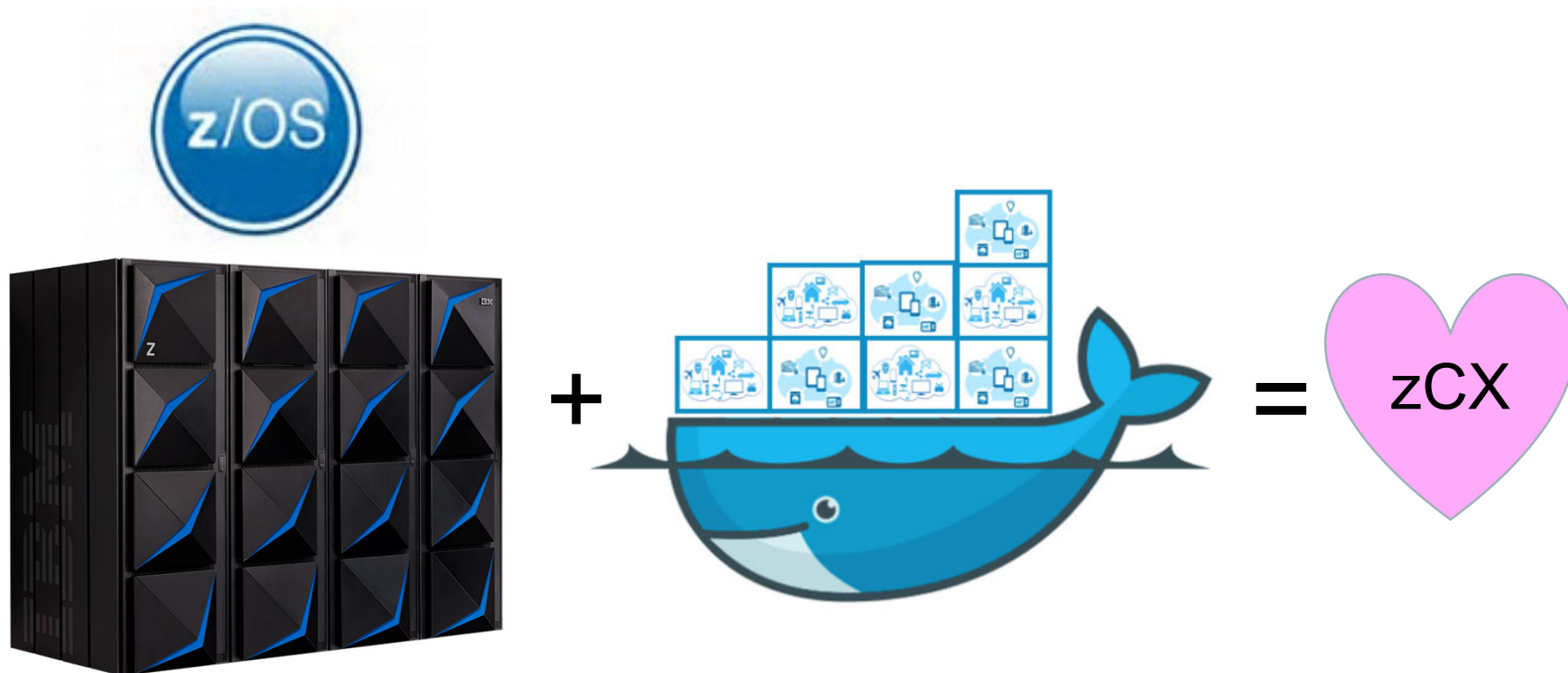
Why LDAP user management on z/OS ?

- 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

Part 1

- zCX user management and authentication**
- Configure zCX for LDBM backend LDAP support**

zCX: exciting new capabilities



zCX brings an exciting new capability to z/OS:
the ability to deploy s390x Docker Linux containers directly under z/OS

What Is IBM z/OS Container Extensions (zCX)?

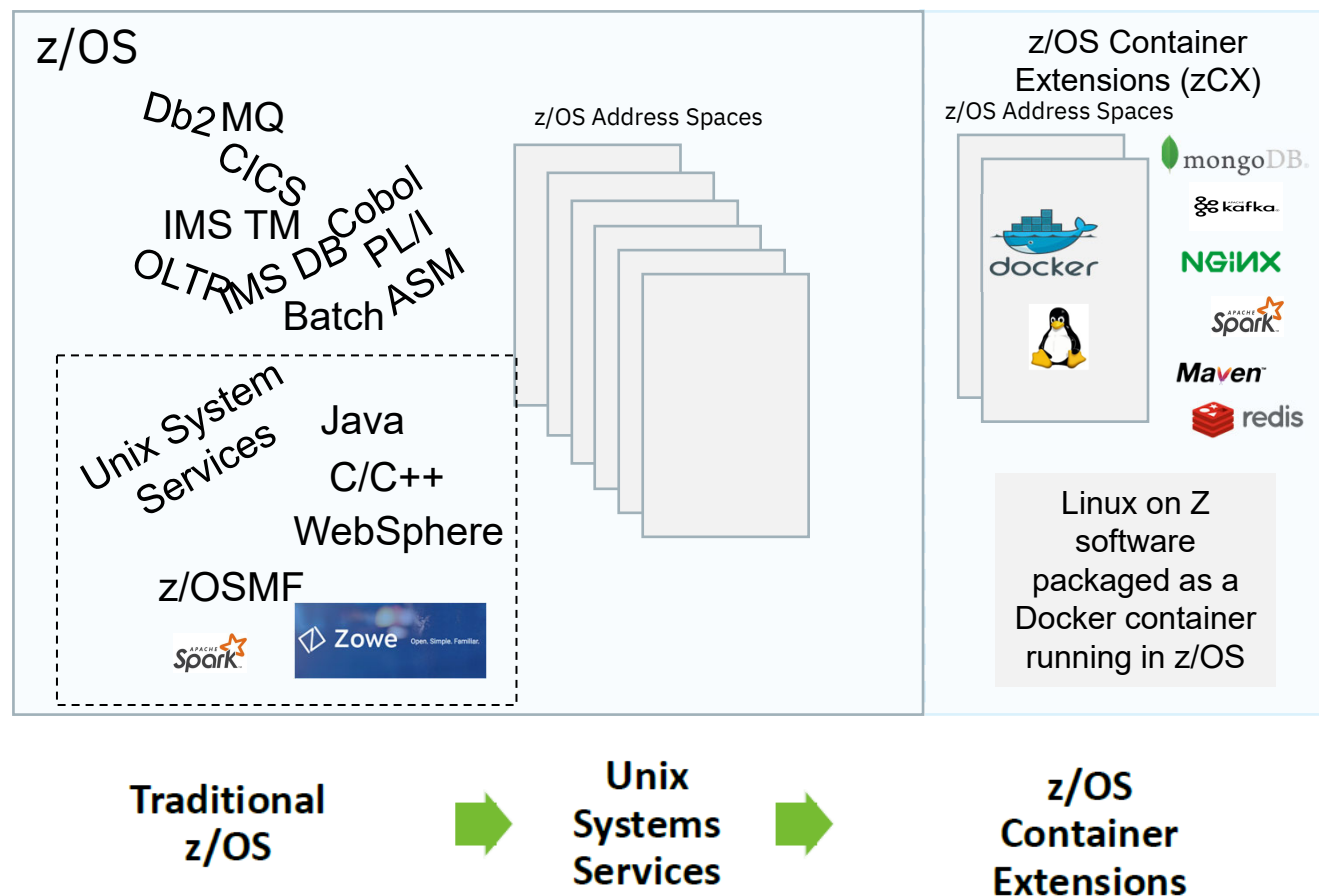
New function in z/OS 2.4 that enables clients to:

- ✓ Deploy Linux on Z software components as Docker Containers in a z/OS system, in direct support of z/OS workloads
- ✓ Without requiring a separately provisioned Linux server
- ✓ While maintaining overall solution operational control within z/OS and with z/OS Qualities of Service
- ✓ Requires IBM z14 (or later) based server with Container Hosting Foundation (feature code 0104)

Design Thinking Hill Statement:

A **solution architect** can **create a solution to be deployed on z/OS based on components available as Docker containers** in the Linux on Z ecosystem transparently exploiting z/OS QoS, **without requiring z/OS development skills**.

Expanding the z/OS Software Ecosystem



- Traditional z/OS workloads, middleware, subsystems and programming languages
- Unix System Services provided z/OS with a Unix personality enabling porting of Unix applications and new programming languages to the platform
- z/OS Container Extensions (zCX) provides the next big evolution – unmodified Linux on Z Docker images running inside z/OS

zCX architecture

- A zCX instance runs as a standard z/OS address space. Docker containers running inside a zCX instance have no way to access the memory or data contained in any other address space running in the z/OS LPAR.
- zCX is a feature of z/OS V2R4 providing a pre-packaged turn-key Docker environment that includes Linux and Docker Engine components supported directly by IBM
- The initial focus is on base Docker capabilities
- zCX workloads are zIIP eligible providing competitive price performance
- There is limited visibility into the Linux environment
 - No root access is allowed / no sudo capabilities
 - Access is as defined by Docker interfaces (CLI)
 - No direct access to underlying Linux kernel
 - No mechanism that allows any container — including the SSH container in the zCX instance —to access any data set, USS directory, or file that resides in z/OS.

SSH – planning

- When you connect to a zCX instance to logon, this connection is over ssh
 - To use ssh you need to set up ssh keys (default mode)
 - Setup of ssh keys can be done on z/OS or on a distributed server:
 - ssh-keygen -t rsa -b 4096 -C your_email@domain.com
 - Putty
 - ...
- If you plan to logon to the zCX instance from z/OS (OMVS, telnet,SSH), then set up ssh keys on z/OS
- If you plan to logon to the zCX instance from a distributed server, then set up ssh keys on that distributed server
- If you want to be able to logon to the zCX instance from both z/OS and a distributed server
 - Then setup ssh keys in one environment and copy to the other

Prerequisites – z/OSMF

- You need z/OSMF V2R4 for execution of different workflows for zCX:
 - Provisioning (for initial setup and local user management using SSH keys)
 - Backup configuration
 - Reconfiguration (for switching to LDAP user management)
 - Restore configuration
 - Upgrade (Service and maintenance)
 - Rollback (Service and maintenance)
 - Deprovisioning
 - Start instance
 - Stop instance
- The xml based workflow definition files are in
/usr/lpp/zcx_zos/workflows
- The corresponding workflow variable file is in
/usr/lpp/zcx_zos/properties
 - It should be used as a template for the workflows of your planned zCX instances

zCX z/OSMF Input Variables Properties file

- zCX provisioning workflow steps can be executed manually or automated in z/OSMF
 - **Step #1** is a manual step to gather/verify all input values for zCX provisioning workflow variables
 - From **Step #2 onwards**, you can automate rest of the workflow steps
 - Select the check box of Step #2, and select “Actions” button and select “Perform”
 - **Last step** of the zCX provisioning workflow provides the z/OS console start command
- Input values for the zCX provisioning workflow variables can be provided in two ways:
 - Using the workflow variables input properties file - When creating the zCX provisioning workflow
 - Using the workflow UI panel – When performing the zCX provisioning workflow step #1
- zCX provides a sample provisioning workflow variables input properties file `/usr/lpp/zcx_zos/properties/workflow_variables.properties`
 - Default values are provided via sample workflow variables properties file and workflow UI panel
 - Copy and customize according to your installation requirements

zCX – z/OSMF – Create Workflow

Create Workflow

Type or select a workflow definition file to use for creating a new workflow.
For a z/OS data set, specify a fully qualified name, with no quotes.

* Workflow definition file:

Type or select a variable input file to populate the new workflow. For a z/OS qualified name, with no quotes.

Workflow variable input file:

* System:

< Back

Next >

Finish

Load zCX provisioning workflow

- Supplied zCX Provisioning xml file location and name

Workflows
Simplifies tasks through guided step-based workflows, and provides administrative functions for assigning workflow responsibilities and tracking

Actions ▾

12 of 96

Workflow Name Filter

Reconfigure

Add Data Dis


Deprovision Z


Provision ZC

Deprovision Z

Create Workflow

* Location (system) of definition and variable input files:
PLEX75.SC74 (SC74_005) - Local

* Workflow definition file: 
/usr/lpp/zcx_zos/workflows/provision.xml

Workflow variable input file: 
/global/zcx/cfg/properties/zcxed02_workflow_vars.properties

< Back Next > Finish Cancel Help

Input field	Value	Description
Workflow definition file:	/usr/lpp/zcx_zos/workflows/provision.xml	Location of supplied zCX provisioning workflow
Workflow variable input file	/global/zcx/cfg/properties/zcxed01_workflow_vars.properties	Location of our customized property file
System	SC74	LPAR to run zCX instance on

customized zCX property file location and name

zCX – z/OSMF – Execute Workflow

Provision ZCXED01

Description: Provision a IBM zOS Container Extensions Appliance Instance.

Owner: zcxprv1 System: PLEX75.SC74 (SC74)

Percent complete: 0%

Steps complete: 0 of 38 Status: In Progress

Workflow Steps

Actions ▾

No filter applied

<input type="checkbox"/> State Filter	No. Filter	Title Filter
<input type="checkbox"/> Ready	1	
<input type="checkbox"/> Not Ready	2	
<input type="checkbox"/> Not Ready	3	

<input type="checkbox"/> Not Ready	12	
<input type="checkbox"/> Not Ready	13	

Dependencies Notes **Perform** Status Input Variables Feedback

Review Instructions

Review and confirm the instructions provided below have been performed on **PLEX75.SC74 (SC74)**, then click complete.

Instructions:

Start the zCX appliance instance with the following start z/OS console command:

S GLZ,JOBNAME=ZCXED01,CONF='/global/zcx/instances/ZCXED01/start.json'

Step 1 – Docker User Management

✓ Input Variables

- ✓ zCX General Configuration
- ✓ zCX CPU and Memory Configuration
- ✓ zCX Network Configuration
- ✓ zCX Root and Config Storage Configuration
- ✓ zCX Instance Directory Storage Configuration
- ✓ zCX Swap Data Storage Configuration
- ✓ zCX User Data Storage Configuration
- ✓ zCX Diagnostics Data Storage Configuration
- ✓ zCX Docker Configuration
- ✓ zCX Proxy Configuration

➡ zCX Docker User Management Configuration

Input Variables - zCX Docker User Management Configuration

Enter the variable values for this input category.

* Docker Admin User ID: ⓘ - The administrator of local Docker users in the IBM z/OS.

admin

* Docker Admin SSH Key: ⓘ - Public SSH key for docker administrator user ID:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDHu/
Hky1Vmc2s8Phk3Mmq+4x70jti9rZtOUoeD
1DqPIWfnJAf2wn1BQz9TMHp+MKOzGReaP3
```

* Enable LDAP Authentication: ⓘ - Configure LDAP client for Docker user management.

FALSE

LDAP Client Configuration File Path: ⓘ - File path to LDAP client configuration file for LDAP client.

/global/zcx/cfg/racflDap/sc74ldap.conf

* Enable LDAP Client TLS Authentication: ⓘ - Enable TLS authentication for LDAP client.

< Back

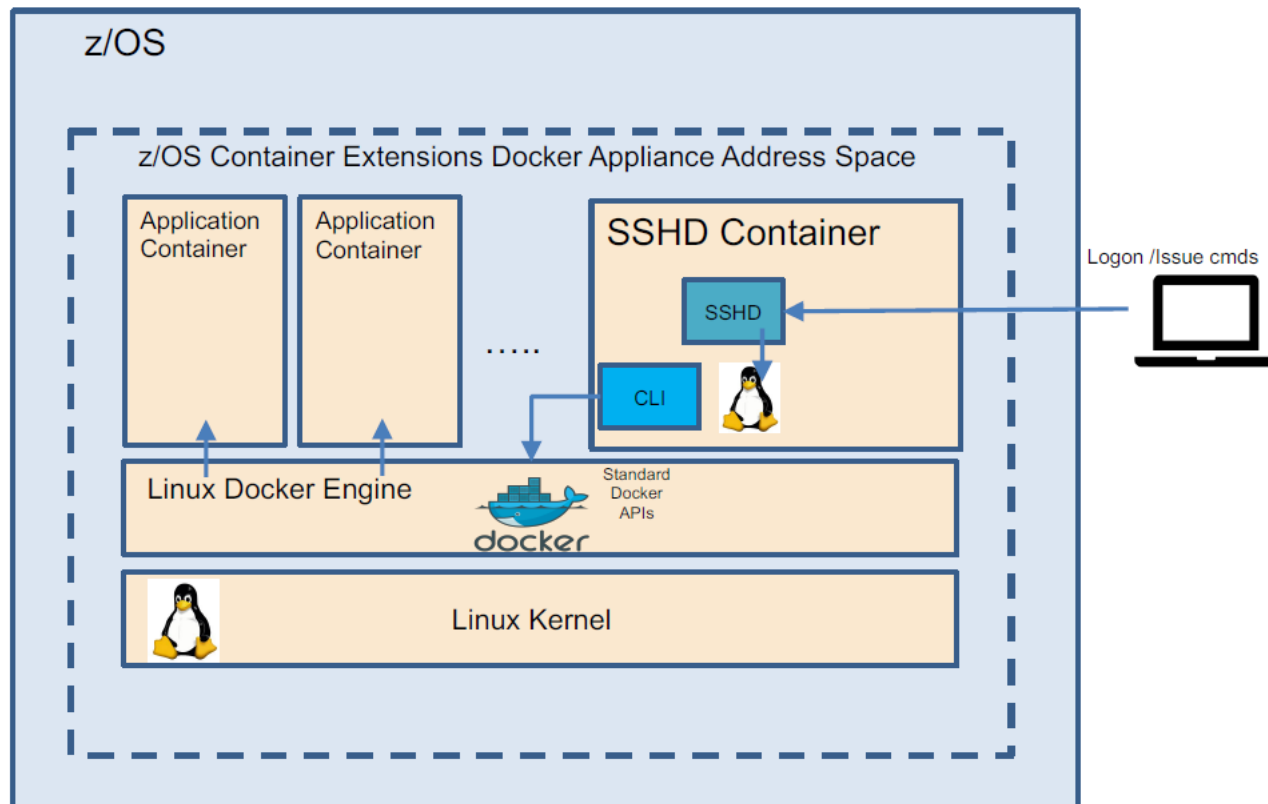
Next >

Save

Finish

- If not using LDAP, this is first User ID that will be defined in the zCX instance
- ssh public key must be supplied here
- Set to TRUE to enable LDAP
- File contains settings on how to connect to LDAP

Deploy Linux on Z docker container using standard Docker Command-Line Interface(CLI)



Logon to zCX Instance

```
ssh admin@sc74cn01.pbm.ihost.com -p 8022
```

- Logon to USS on z/OS

- Using ZCXADM1

```
wtsc74oe.pbm.ihost.com - PuTTY
login as: zcxadm1
zcxadm1@wtsc74oe.pbm.ihost.com's password:
ZCXADM1 @ SC74:/u/zcxadm1>
ZCXADM1 @ SC74:/u/zcxadm1>ssh admin@sc74cn01.pbm.ihost.com -p 8022
```

- Issue ssh command to connect and logon to zCX instance

- On first connection get expected ssh message like this:

```
The authenticity of host '[sc74cn01.pbm.ihost.com]:8022 ([129.40.23.68]:8022)'
can't be established.
ECDSA key fingerprint is SHA256:IA5Sw4oNK7w0DhFGkH4ayyJNvT73mluvQprHjqQC41M.
Are you sure you want to continue connecting (yes/no)?
```

- Successful logon:

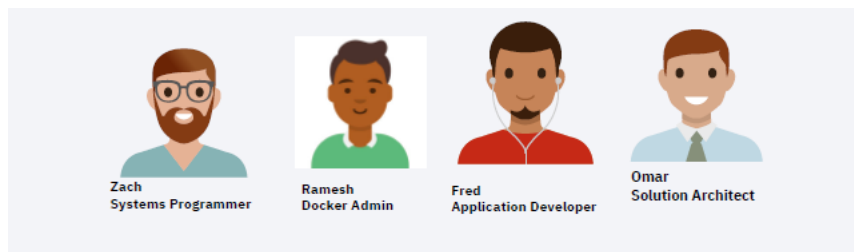
```
FOTS2274 Warning: Permanently added
'[sc74cn01.pbm.ihost.com]:8022,[129.40.23.68]:8022' (ECDSA) to the list of known
hosts.
```

```
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 z/OS Container Extensions Guide".
Sudo only permits specific User Management functions. See additional
documentation for details.
```

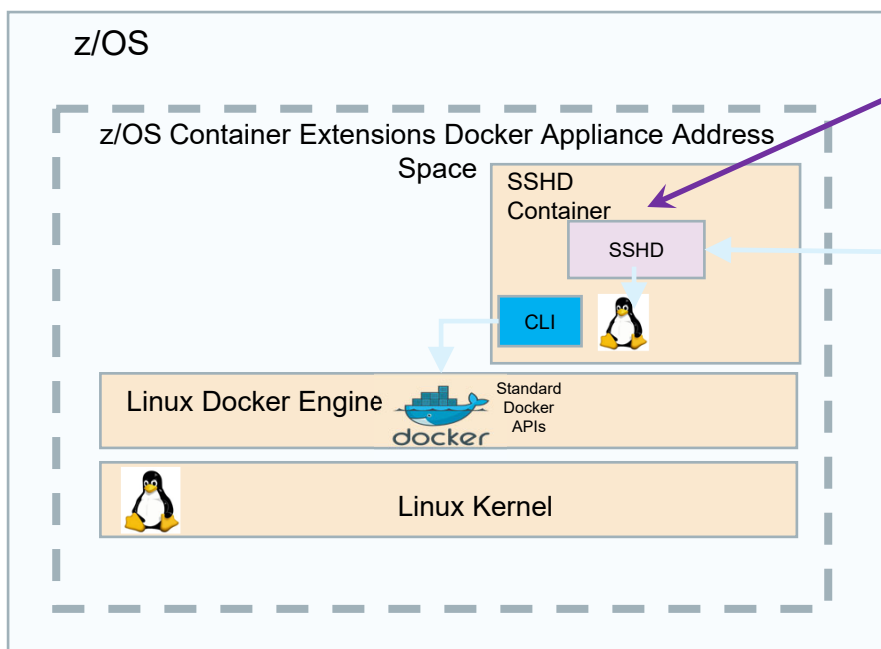
```
admin@d377e12aa94b:~$ id
uid=1001(admin) gid=1001(admin) groups=1001(admin),27(sudo),109(docker)
```

Docker administrators and Docker users

- Docker administrators and permitted Docker users can deploy any Linux on Z docker container image using standard Docker CLI
 - Docker administrator is responsible for managing the Docker daemon and Docker users within the zCX appliance instance.
 - The user information for the Docker administrator must be given during provisioning so that the user ID can be configured to manage the Docker daemon in the zCX instance
 - The Docker administrator does not necessarily need to be a z/OS user or zCX appliance administrator
 - Docker users have the ability to run Docker commands only (members of the Docker user group)
 - Docker users do not have privileges in Sudo, which is the program that allows users to use elevated privileges.
 - Therefore, they cannot create or modify other Docker users.
 - **Sudo is restricted**, only permits specific User Management functions, only by the zCX admin user



SSH Container



- SSH CLI container starts automatically when zCX instance starts

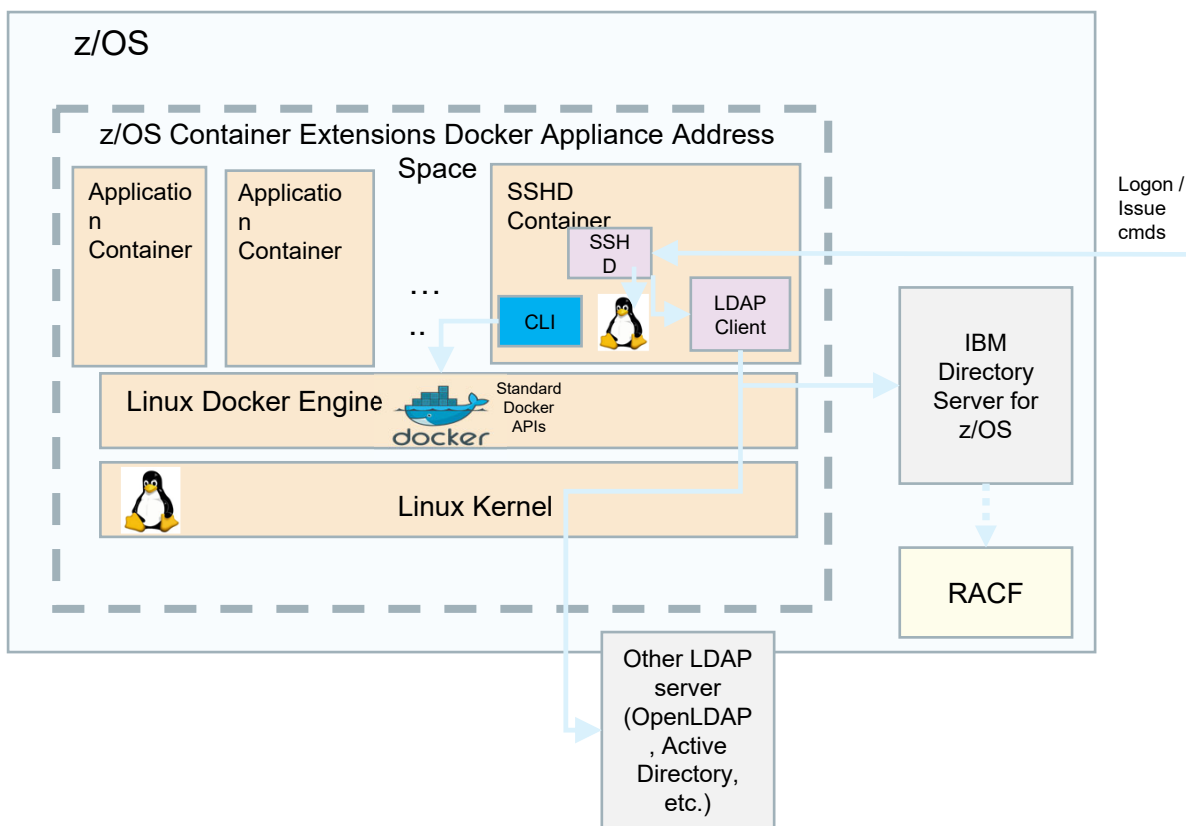


- Logon to zCX Instance over ssh connection port 8022
- Once logged on, can enter Docker commands
- You can use the zCX admin credentials to create additional zCX users with the password option.
- You can also upload the public SSH keys so the zCX user can use them to authenticate into the zCX SSH CLI container and retrieve the data.

zCX – Administration of Docker CLI

- Access to the Docker CLI managed using either:
 - local user management
 - central LDAP server-based user management
- LDAP authentication can be preferable to local user authentication because it provides an easy way to create identical user and group configurations across multiple instances.
 - Modifications to users and groups are also much simpler when you use LDAP because changes to the LDAP architecture are immediately propagated to the authentication scheme of the instances..
- When LDAP authentication is enabled for an instance, local user management and public key authentication are disabled.
- This link from the z/OS 2.4 Knowledge Center describes these options in further detail:
 - https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.izso100/izso100_usermanagementchapterintro.htm

User Management and Authentication: Local Registry or LDAP

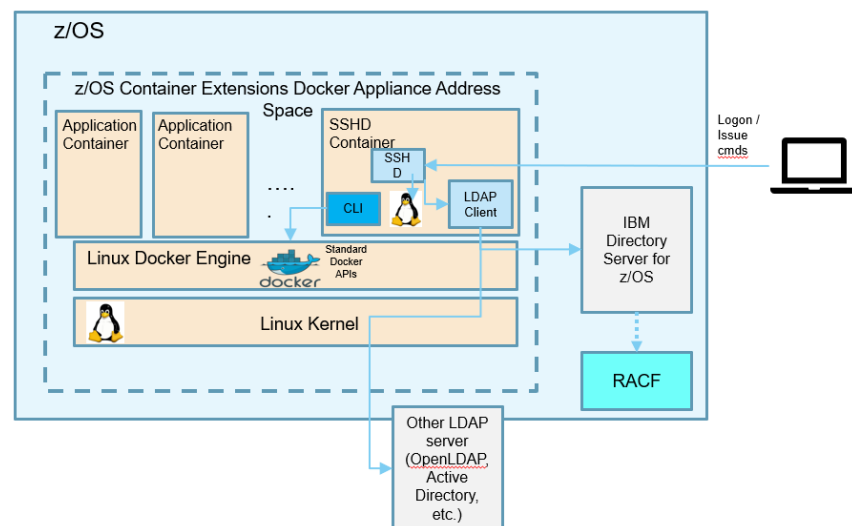


There are several options for user management and authentication for both the SSHD Container and Application Containers:

1. Local appliance registry
2. z/OS LDAP Server (IBM Directory Server for z/OS, included with base z/OS), with a choice of self-contained or native SAF authentication
3. “Remote” LDAP server (e.g. OpenLDAP, Active Directory, etc. – could be running on zCX or on Linux on Z)

zCX – LDAP User administration

- In zCX provisioning workflow specify target LDAP server
 - LDAP will then be used for authorization and authentication of the zCX Docker CLI users
- Benefit:
 - Any number of zCX instances have consolidated user management
- LDAP Options:
 - IBM® Tivoli® Directory Server for z/OS®
 - Ships with z/OS
 - No charge
 - Provides optional integration with RACF or other compliant security manager products using **LDBM or TDBM with native-authentication support**
 - Result is logon to zCX with RACF user ID and password
 - Any supported LDAP server
 - OpenLDAP
 - Active Directory



zCX – Local User Management

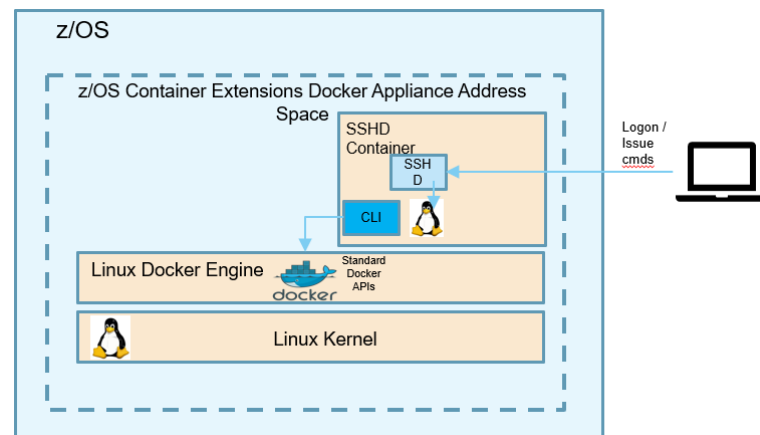
• Local User Management

- Using this method:
 - a Docker administrator user ID is specified during provisioning of a zCX instance
 - This user ID has access to the Docker CLI
 - Can also define and delete additional zCX users in that zCX instance

```
sudo adduser --ingroup docker username
```

- Simple approach, useful for:
 - initial testing in zCX
 - limited number of zCX instances are deployed
- Requires that all authorized users are defined and maintained on each individual zCX instance
 - no sharing of user access across zCX instances.

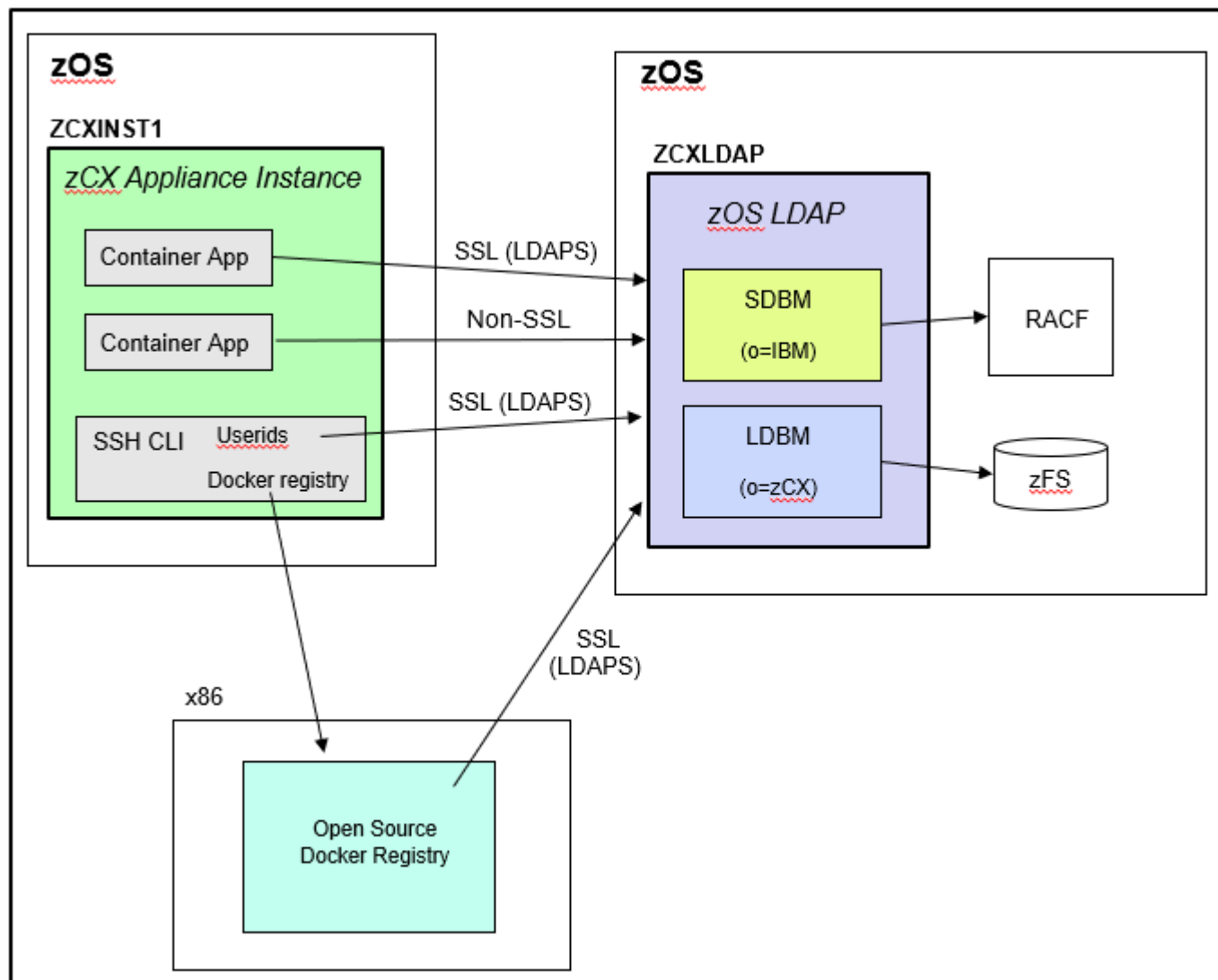
Once created, the new Docker user can SSH into the container and use Docker with the command:
`ssh username@ip_address -p 8022`



zCX and LDAP ITDS on z/OS

- *Can I use a zOS LDAP server with zCX?*
 - Yes! - zCX CAN use a zOS LDAP server for user/group management and authentications, but some customizations are needed.
- *Can the zOS LDAP server be configured ONLY with SDBM (RACF) backend for zCX?*
 - No - The zOS LDAP server can NOT be configured ONLY for SDBM (RACF) backend. Additional schema definitions are needed for zCX and they can not be added to SDBM.
 - The zOS LDAP server MUST be configured for a LDBM (filesystem) or TDBM (database) backend. This is where the zCX users/groups entries (posixAccount/posixGroup) and attributes will exist.
- *Can zCX users be authenticated by RACF?*
 - Yes and No ! - The zOS LDAP server CAN be configured so users (all or some) are authenticated by RACF for their credentials or passwords. This will be through the Native Authentication feature provided by the zOS LDAP server.
- You cannot use the SDBM (RACF) backend alone and need to add an LDBM (or TDBM) backend to hold the entries needed for zCX.

zOS LDAP server in our zCX environment group



The issue with ldap ITDS on z/OS

- The zOS LDAP (IBM Tivoli Directory Server or ITDS) is a full featured LDAP server that can be used for many solutions.
- However, for use with zCX (and other Open Source applications), it needs some additional setup that can be a bit tricky and complicated.
- Need to **tweak the schema** and the LDBM backend needed for zCX and other open source applications.

LDAP schema dependencies for zCX

- Basic support for LDAP in zCX (through the Linux environment it provides) is implemented by modules added to the system components:
 - Pluggable Authentication Module (PAM), in particular, **pam_ldap**
 - The Name Server Switch (NSS), a component of the GNU C Library, in particular, **nss_ldap**
- Both **pam_ldap** and **nss_ldap** assume that the LDAP schema in use is compatible with the definition contained in RFC 2307.
- It is actually the **posixAccount** object that contains the attributes that **pam_ldap** and **nss_ldap** need.
- Adding the objects and attributes from the NIS schema is necessary to use the z/OS LDAP servers to authenticate users.
- **Problem:** zOS LDAP does not provide the schema needed to implement these. For this part, users are left on their own.
- Making it more complicated, various incompatibilities exist.
- *Help us change this and vote for zOS LDAP to include the schema needed. Go to the IBM RFE (Request For Enhancements) web site and "Vote" for RFE ID 137082. Add schema from RFC 2307 -*
https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=137082

RFC 2307

RFC 2307 (<https://tools.ietf.org/html/rfc2307>) includes many schema attributes and classes that are not found in z/OS LDAP

- Examples of the **attributes** needed are:
 - uidNumber
 - gidNumber
 - homeDirectory
 - memberUid
 - nisNetgroupTriple
 - ipServicePort
 - ipServiceProtocol
 - macAddress
 - bootParameter
 - bootFile
- Examples of **object classes** needed are:
 - posixAccount
 - shadowAccount
 - posixGroup
 - ipService
 - ipProtocol
 - ipHost
 - ipNetwork
 - nisNetgroup
 - nisObject
 - bootableDevice

nis_schema.ldif

```
BROWSE /u/prichar/ES65/nis_schema.ldif      Line 0000000000 Col 001
Command ==>                                Scroll ==> C
```

```
***** Top of Data *****
```

```
dn: cn=schema
```

```
changetype: modify
```

```
add:attributetypes
```

```
attributetypes: ( 1.3.6.1.1.1.0 NAME 'uidNumber'
```

```
DESC 'An integer uniquely identifying a user in an administrative domain'
```

```
EQUALITY integerMatch
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

```
attributetypes: ( 1.3.6.1.1.1.1 NAME 'gidNumber'
```

```
DESC 'An integer uniquely identifying a group in an administrative domain'
```

```
EQUALITY integerMatch
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

```
# attributetypes: ( 1.3.6.1.1.1.1.2 NAME 'gecos'
```

```
# DESC 'The GECOS field; the common name'
```

```
# EQUALITY caseIgnoreIA5Match
```

```
# SUBSTR caseIgnoreIA5SubstringsMatch
```

```
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

```
# attributetypes: ( 1.3.6.1.1.1.1.3 NAME 'homeDirectory'
```

```
# DESC 'The absolute path to the home directory'
```

```
# EQUALITY caseExactIA5Match
```

```
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

```
# attributetypes: ( 1.3.6.1.1.1.1.4 NAME 'loginShell'
```

```
# DESC 'The path to the login shell'
```

```
# EQUALITY caseExactIA5Match
```

```
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

```
...
```

zCX LDAP users

#----- zCX Users -----#

zCX_Base1.Idif

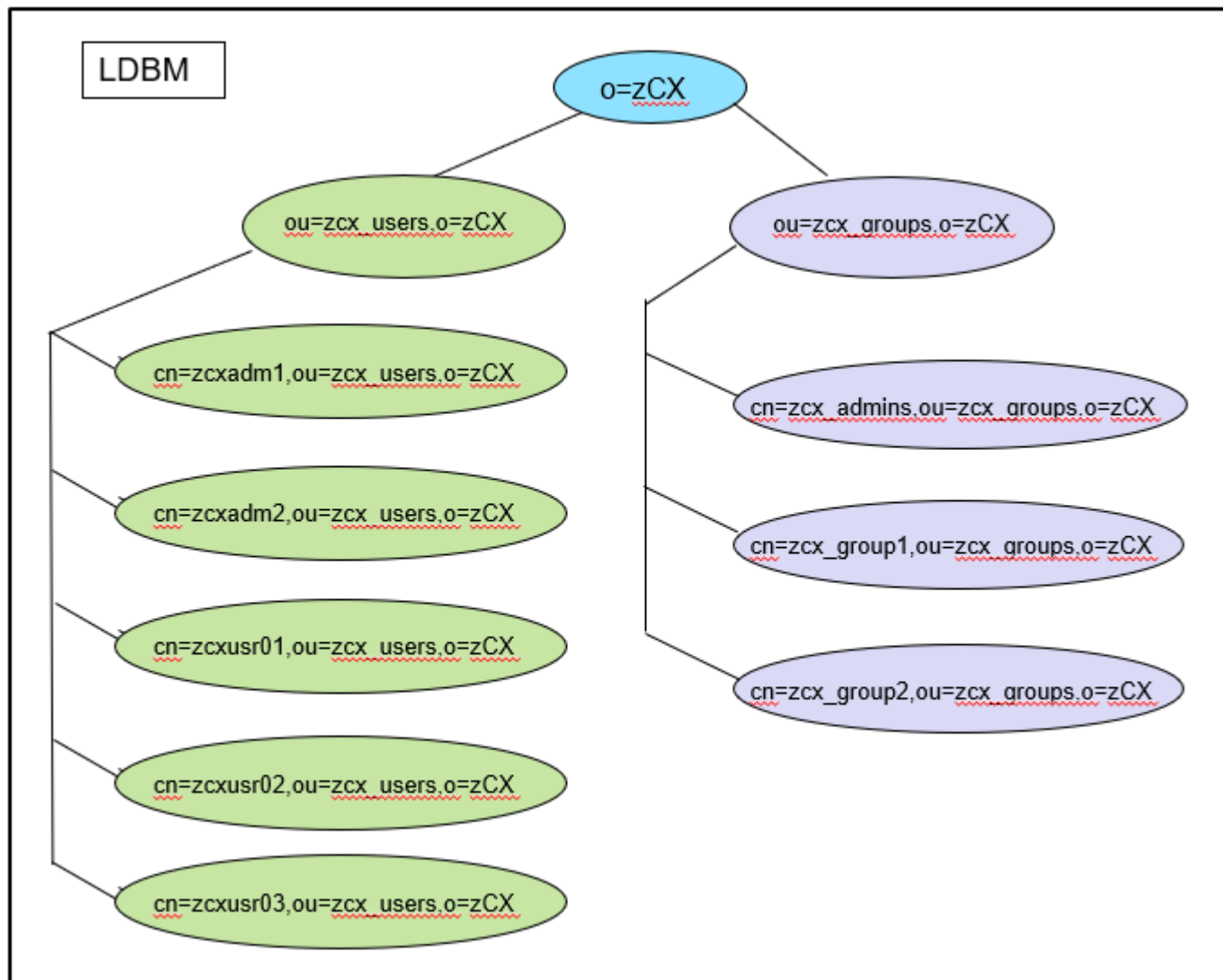
dn: ou=zCX_users,o=zCX,o=ibmmop,c=fr
objectclass: top
objectclass: organizationalUnit
ou: People
ou: zcx_users

dn: cn=zcxadm1,ou=zCX_users,o=zCX,o=ibmmop,c=fr
objectclass: top
objectclass: organizationalPerson
objectclass: person
objectclass: posixAccount
objectclass: shadowAccount
objectclass: inetOrgPerson
cn: zcxadm1
gidnumber: 16000
homedirectory: /home/zcxadm1
sn: zcxadm1
uid: zcxadm1
uidnumber: 6001
givenname: zcxadm1
loginshell: /bin/bash
mail: zcxadm1@tx9.mop.ibm.com
userpassword: secret

add the zCX users/groups to the LDBM backend

```
/*      -PURPOSE -run ldapmodify command_____
/*
/*      _____
/*      _____
/*
/*
//S1    EXEC PGM=IKJEFT1B
//SYSTSPRT DD SYSOUT=*
//SYSEXEC DD DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSTEM DD DUMMY
//SYSTSIN DD *
oshell +
ldapadd -v -h 127.0.0.1 -p 3890 +
-D cn=Admin -w xxxxxx -f /u/prichar/ES65/zCX_Base1.ldif
//
```

Directory structure for our zCX users and groups, provided in the **zCX_Base.ldif** file



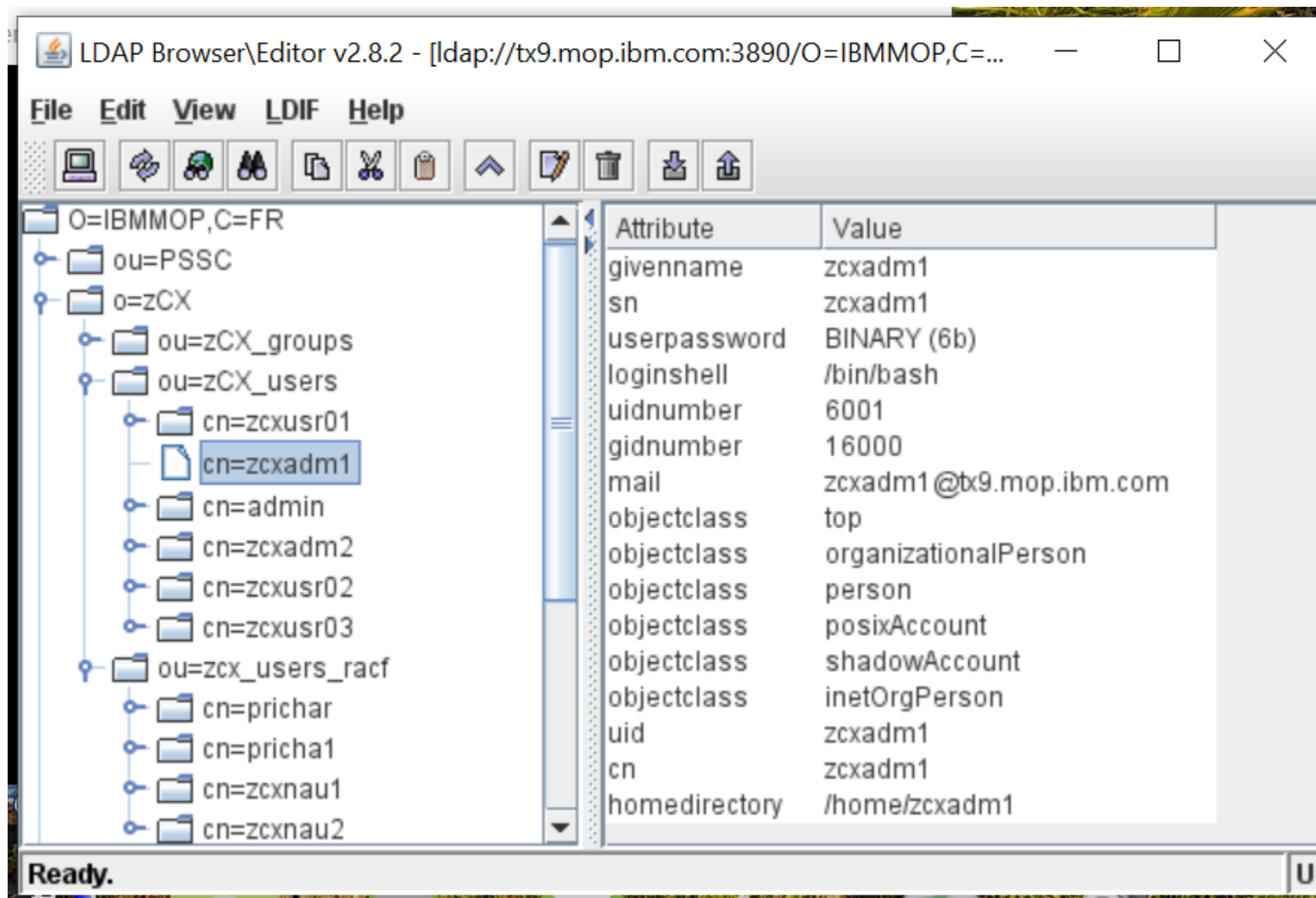
zCX LDAP posixAccount attribute settings

- for use with zCX, some of the posixAccount attribute values must be present and within specific ranges.

attribute	Required	Valid values	Description
uid	Yes	Valid string	Linux login id
uidNumber	Yes	Integer in range 1000 to 59999 Must be unique for each user	User's Linux UID
homeDirectory	No	/home/<uid>	User's home directory.
loginshell	No	/bin/bash	User's login shell

- Some of the errors seen when not configured properly:
 - homedirectory set to "/u/xxx" caused user to fail to log in successfully (/u does not exist in zCX)
 - loginshell not set to "/bin/bash" caused user to fail to log in successfully
 - Error message for 'groups: cannot find name for group ID 10000" due to missing "posixGroup" with gidNumber of "10000"
 - Multiple users with same uid may result in wrong userid used in zCX.
 - Changing homedirectory will result in a new directory created (if it doesn't exist). The existing homedirectory will NOT be deleted/renamed.

zCX ldap users



Configure zCX for ldap

- Update the **ldap.conf** used for the zCX instance. Set the "base" option to point to the zCX root in the LDBM backend:

```
# Here is a sample ldap.conf file to allow a zCX appliance to authenticate users
```

```
uri ldap://9.100.200.300:3890
```

```
# uri ldaps://9.100.200.300:3896
```

```
base o=ibmmop,c=fr
```

```
ldap_version 3
```

```
# LDAP searches will be performed using the reader distinguished name defined be  
binddn cn=zcxadm1,ou=zCX_users,o=zCX,o=ibmmop,c=fr  
bindpw secret
```

- Run a zCX **Provision or Reconfiguration workflow** against the target zCX instance to apply/update the ldap.conf file.
/usr/lpp/zcx_zos/workflows/reconfigure.xml
 - *Note: If running a Reconfiguration workflow against an existing zCX instance and only the ldap.conf file is changed, then you will still have to step through all of the workflow panels. The updated ldap.conf file will be applied to the zCX instance during another step in the workflow.*
- Recycle the zCX instance to pick up the change

zCX Docker User Management Configuration for provision workflow

- For the reconfigure workflow, navigate to the “zCX Docker User Management Configuration” substep as shown

- ✓ Input Variables
 - ✓ zCX General Configuration
 - ✓ zCX CPU and Memory Configuration
 - ✓ zCX Network Configuration
 - ✓ zCX Docker Configuration
 - ➡ **zCX Docker User Management Configuration**
- Review Instructions

Input Variables - zCX Docker User Management Configuration

Enter the variable values for this input category.

* Docker Admin User ID: ⓘ - The administrator of local Docker users in the IBM zCX appliance instance:

* Docker Admin SSH Key: ⓘ - Public SSH key for docker administrator user ID:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACjtcg/x7
kSbHaNcWuT
/v2w+ZBMEp9XtsWwUADobDeDSV8kLZK+XuvLMz
```


* Enable LDAP Authentication: ⓘ - Configure LDAP client for Docker user management:

LDAP Client Configuration File Path: ⓘ - File path to LDAP client configuration file to configure the IBM zCX appliance instance LDAP client:

* Enable LDAP Client TLS Authentication: ⓘ - Enable TLS authentication for LDAP client communication with remote LDAP server:

LDAP Client TLS CA Certificate: ⓘ - File path to LDAP client TLS authentication CA certificate:

Time to ssh using ldap !

- SSH to zCX instance using one of userids configured in the LDBM "o=zCX" base.
 - zcxadm1
 - zcxadm2
 - zcxusr01
 - zcxusr02
 - zcxusr03
- **Note:** a decision was made by development not to create any local users (like the default **admin** account) when LDAP-based user management is configured.
- At this point, the **admin** account that was configured during provisioning no longer exists.
 - You can confirm this fact by attempting to ssh into the admin account 
- See **Part-2** for more details on how to recreate it

ssh into the zCX instance with ldap authentication: no ssh keys used any longer

AzureAD+philippeRichard@LAPTOP-QP04D215 MINGW64 ~

\$ ssh -p 8022 zcxadm1@9.212.128.231

zcxadm1@9.212.128.231's password:

LDAP userpassword attribute

secret

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: Thu May 7 10:32:16 2020 from 9.145.38.171

zcxadm1@TX9:~\$ id

uid=6001(zcxadm1) gid=16000(zcx_admins)

groups=16000(zcx_admins),109(docker),16001(zcx_group1),16002(zcx_group2)

zcxadm1@TX9:~\$ ls -al

total 16

drwxr-xr-x 1 zcxadm1 zcx_admins 92 Apr 30 13:23 .

drwxr-xr-x 1 root root 104 May 4 09:23 ..

-rw----- 1 zcxadm1 zcx_admins 174 May 7 11:25 .bash_history

-rw-r--r-- 1 zcxadm1 zcx_admins 220 Apr 30 13:21 .bash_logout

-rw-r--r-- 1 zcxadm1 zcx_admins 3771 Apr 30 13:21 .bashrc

drwx----- 1 zcxadm1 zcx_admins 40 Apr 30 13:21 .cache

-rw-r--r-- 1 zcxadm1 zcx_admins 807 Apr 30 13:21 .profile

On-demand recorded Part-2...

- **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
- LDAP misc
 - Audit and monitor ldap logins
 - Change own LDAP passwords

- **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

- **link for Part-2 On-demand video:**

<https://ibm.box.com/s/qeo2krp2s5a2ehvjcr0qtuyzo8a73wjd>

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>



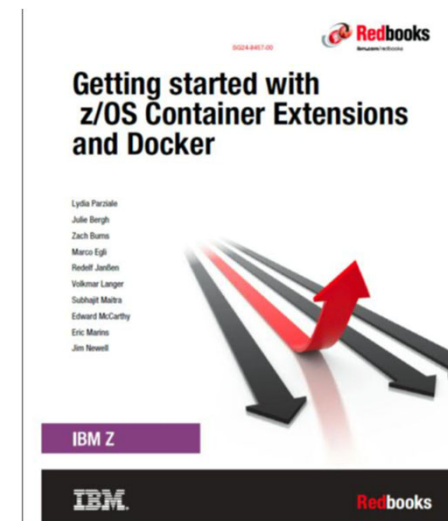
zCX – IBM Redbook



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

Appendix

Extending the schema of the z/OS LDAP server:

RFC 2307

nis_schema.ldif file

LDAP Schema changes

nis_schema.ldif file

- **Extending the schema of the z/OS LDAP server**

- As mentioned above, while the z/OS LDAP server is **missing certain syntax definitions and matching rules** that OpenLDAP provides (and that many schemas use), the z/OS LDAP server provides an interface to extend the schema,

- **Obtaining schema definitions**

- OpenLDAP is a project that develops Open Source LDAP applications and development tools. The project produces the OpenLDAP Suite, which includes the most common LDAP server implementation found on Linux systems, slapd.
- The Web site for the OpenLDAP Suite is: <http://www.openldap.org/>
- OpenLDAP supplies the schemas that implement the object and attribute types described in Internet RFCs (such as RFC 2307 and RFC 2798), and other schemas such as those for Samba.
- Among them is **the NIS schema**, which provides the **object classes and attributes usually used by UNIX and Linux systems for authentication**.
- While there are many other sources for the schema definitions needed, we are going to start by using the **nis.schema** that is provided with OpenLDAP v2.4.47 (Obtainable from <http://www.openldap.org/>).

- **Converting OpenLDAP schema files for the z/OS LDAP server**

- The Directory Information Tree (DIT) that is used for the schema in OpenLDAP differs from that used for the z/OS LDAP server, so taking an OpenLDAP schema LDIF file and applying it to the z/OS LDAP server directly is not possible.
- **In addition, the missing syntax definitions and matching rules also prevent OpenLDAP schema LDIF files from being used with the z/OS LDAP server.**

LDIF differences between OpenLDAP OLC format and z/OS LDAP format

- The process of converting OpenLDAP schema files can require some effort.
- **LDIF formats**
 - Even though schema files in OpenLDAP's LDIF cannot be directly applied to the z/OS LDAP server, they **can be converted** to the format which is acceptable to z/OS LDAP.
 - The following table shows the differences we found between the OpenLDAP LDIF schema and the z/OS LDAP LDIF schema.

	OpenLDAP OLC	z/OS LDAP
DN of schema updates	cn=schema,cn=config	cn=schema
Name of object class for attribute definitions	olcAttributeTypes	attributeTypes
Name of object class for object class definitions	olcObjectClasses	objectClasses

Matching rules

- The schema needed may be using LDAP syntaxes or matching rules that are not present in the z/OS LDAP server.
- The set of matching rules which are supported by the z/OS LDAP server cannot be modified, added to, obsoleted, or deleted by users.
- Matching rule "caseIgnoreIA5SubstringsMatch" is only supported when the LDAP server is running at server compatibility level 6 or higher.
- The following table shows rules that are not defined in the z/OS LDAP server and the values we replaced them with that do exist.

Rule not found	Replaced with	Objects using in NIS schema
caseIgnoreIA5SubstringsMatch	caseIgnoreSubstringsMatch	gecos
caseExactIA5SubstringsMatch	caseExactSubstringsMatch	memberUid memberNisNetgroup nisMapEntry

Missing Syntax

Missing Syntax

- The SYNTAX for 'nisNetgroupTriple' and 'bootParameter' do not exist on zOS LDAP.

attributetypes: (1.3.6.1.1.1.1.14 NAME 'nisNetgroupTriple'
DESC 'Netgroup triple'
SYNTAX 1.3.6.1.1.1.0.0)

attributetypes: (1.3.6.1.1.1.1.23 NAME 'bootParameter'
DESC 'rpc.bootparamd parameter'
SYNTAX 1.3.6.1.1.1.0.1)

- Instead, since we don't anticipate strict usage of these in our environment, **Syntax 1.3.6.1.4.1.1466.115.121.1.26** (IA5 String, IA5 characters, commonly known as 7-bit ASCII) was used.

Previously defined attributetypes

- **Previously defined attributetypes**
- The following attributetypes were already defined from /usr/lpp/ldap/schema.IBM.ldif, but with slightly different characteristics.
 - attributetypes NAME 'gecos'
 - attributetypes '1.3.6.1.1.1.1.3' (homeDirectory)
 - attributetypes '1.3.6.1.1.1.1.4' (loginShell)
 - attributetypes '1.3.6.1.1.1.1.15' (ipServicePort)
- We decided to use the definitions supplied in the /usr/lpp/ldap/schema.IBM.ldif file and commented them out of our nis_schema.ldif file.

The full process to convert the nis schema:

- Copy nis.schema to nis_schema.ldif
- Edit nis_schema.ldif
- Remove comment and blank lines
- Replace the opening LDIF syntax with corrected syntax. Add following lines before first attributetype:
dn: cn=schema
changetype: modify
add:attributetypes
- Add following lines before first objectclasses (Note: The dash ("-") is included and on a single line. It closes the "add:attributetypes".)
-
add: objectclasses
- Change:
olcAttributeTypes to attributetypes:
olcObjectClasses to objectclasses:
- Change SYNTAX for 'nisNetgroupTriple' and 'bootParameter'
SYNTAX 1.3.6.1.1.1.0.0 to SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SYNTAX 1.3.6.1.1.1.0.1 to SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
- Comment out
 - attributetypes NAME 'gecos'
 - attributetypes '1.3.6.1.1.1.1.3' (homeDirectory)
 - attributetypes '1.3.6.1.1.1.1.4' (loginShell)
 - attributetypes '1.3.6.1.1.1.1.15' (ipServicePort)

Step 1: update schema of Idbm backend **nis_schema.ldif**

```
/* update schema for zcx openldap support
/*      -PURPOSE -run ldapmodify command_____
/*      Idbm server: ldapld0 port 3890_____
/*      _____
//S1      EXEC PGM=IKJEFT1B
//SYSTSPRT DD SYSOUT=*
//SYSEXEC DD DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSTEM DD DUMMY
//SYSTSIN DD *
oshell +
ldapmodify -h 127.0.0.1 -p 3890 +
-D cn=Admin -w xxxxxx -f /u/prichar/ES65/nis_schema.ldif
//
```

zCX – A turn-key Virtual Docker Server Software Appliance

Pre-packaged Linux Docker appliance

- Provided and maintained by IBM
- Provisioned using z/OSMF workflows

Provides standard Docker interfaces

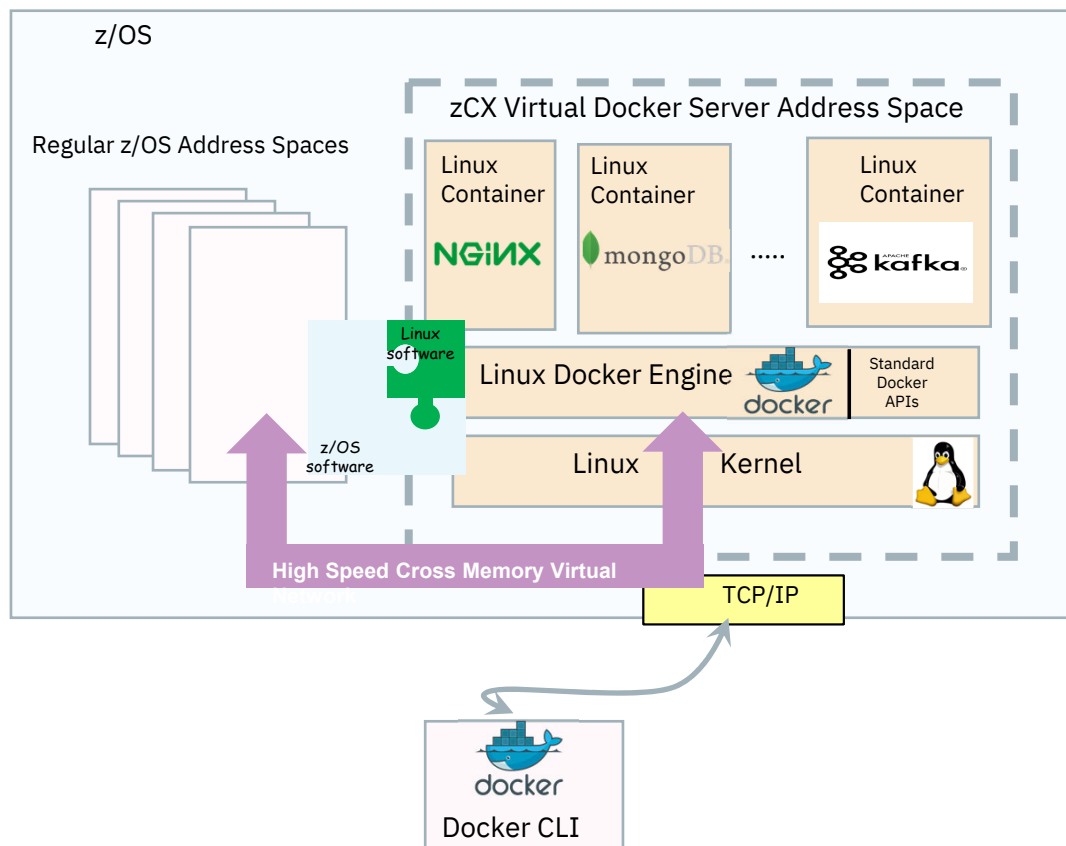
- Supports deployment of any software available as a Docker image for Linux on Z
- Communications with native z/OS applications over high speed virtual IP network
- No z/OS skills required to develop and deploy Docker Containers

No Linux system administration skills required

- Interfaces limited to Docker CLI
- No direct access to underlying Linux kernel

Managed as a z/OS process

- Multiple instances can be deployed in a z/OS system
- Managed using z/OS Operational Procedures
- zCX workloads are zIIP eligible

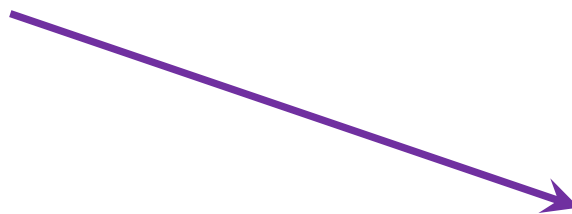


zCX – Property file

Supplied sample in:

`/usr/lpp/zcx_zos/properties/workflow_variables.properties`

Customize then use as input to the z/OSMF workflow



Property	Value	Description
ZCX_REGISTRY_DIR	/global/zcx/instances	Uncommented this line to use default location
ZCX_INSTNAME	ZCXED01	Name of zCX instance to be created
ZCX_SAVE_PROPERTIES	/global/zcx/cfg/properties	Uncomment this line and set to directory where workflow will save copy of properties file
ZCX_HOSTNAME	sc74cn01.pbm.ihost.com	DNS name of TCPIP address that zCX instance will use
ZCX_GUESTIPV4	129.40.23.68	TCPIP address that zCX instance will use
ZCX_HOSTDNS1	129.40.106.1	TCPIP address of DBNS server
ZCX_HLQ	ZCX.REDB	High level qualifier to use for datasets that will be created by workflow