Standalone Enterprise-ready LDAP - 389 Directory Server

https://www.port389.org/docs/389ds/documentation.html

Installing (also installs optional Cockpit dashboard for a nice web browser frontend)
dnf install 389-ds cockpit cockpit-389-ds sssd

Installed Files:

/etc/dirsrv/	Main directory for 389-ds configuration files.
/etc/dirsrv/slapd-config.conf	Main configuration file defining the LDAP server instance.
/etc/dirsrv/slapd-*.conf	Additional key configuration files (e.g., slapd-database.conf, slapd-access.conf)
/etc/dirsrv/schema/	Contains schema files for LDAP entries (core.schema, inetorgperson.schema)
/usr/lib/dirsrv/	Directory containing libraries used by the server
/usr/lib/systemd/system/dirsrv@slapd.service	Systemd service file for managing the 389-ds server process

Executables:

/usr/sbin/ns-slapd	Main server executable for the 389 Directory Server. (more below)
/usr/bin/dsctl	Tool to control and manage instances of 389 Directory Server.
/usr/bin/dsconf	Tool for configuring the server and managing server instances.
/usr/bin/dscreate	Tool to create a new instance of the 389 Directory Server.
/usr/bin/dsidm	Tool for managing identities (users, groups) in 389 Directory Server.
/usr/bin/dsconf-import	Tool for importing LDIF data into the directory server.
/usr/bin/dsconf-export	Tool for exporting directory server data to an LDIF file.
/usr/bin/dslogpipe.py	Tool for processing and managing server logs.
/usr/bin/ldapsearch	Command-line tool for searching LDAP directories.
/usr/bin/ldapmodify	Command-line tool for modifying LDAP entries.
/usr/bin/slaptest	Tool for testing the syntax of slapd configuration files
/usr/bin/slappasswd	Tool for generating hashed passwords for LDAP authentication.

Configuration Files:

/etc/dirsrv/slapd-instance/dse.ldif	Main configuration file for the 389 Directory Server instance.
/etc/dirsrv/slapd-instance/schema/99user.ldif	User-defined schema file for customizing directory attributes and object classes.
/etc/dirsrv/slapd-instance/ldif/template.ldif	Template LDIF file used during the initial server setup.
/etc/dirsrv/slapd-instance/password.conf	Configuration for password policies and storage.
/etc/dirsrv/slapd-instance/certmap.conf	Configuration for certificate mappings and SSL/TLS settings.
/etc/dirsrv/admin-serv/adm.conf	Configuration file for the Directory Server Admin Service.
/etc/dirsrv/admin-serv/console.conf	Admin console configuration file.
/etc/sysconfig/dirsrv	Environment variables for the Directory Server services.

Configuration Directories (files and scripts)

/etc/dirsrv/slapd-instance/schema	Schema files that define the structure and data types stored in the directory.
/etc/dirsrv/slapd-instance/ldif	Directory for LDIF files used for data import/export and initial configuration.
/etc/dirsrv/admin-serv	Configuration files for the administration server, managing the web-based console.
/etc/dirsrv/config	Contains configuration files and scripts for the server's operational settings.

Libraries and Systemd Services:

/usr/lib/dirsrv	Main directory for 389 Directory Server libraries and plugins.
dirsrv@instance.service	Manages a specific instance of the 389 Directory Server.
dirsrv-admin.service	Manages the administration service for the 389 Directory Server.

Commands and Options:

dsctl:		
dsctl instance start	Start the server instance	--debug (Run in debug mode)
dsctl instance stop	Stop the server instance	--force (Force stop the server)
dsctl instance status	Status of the server instance	--verbose (Provide detailed output)
dsconf:		
dsconf instance backend create	Create new backend for the server	--suffix (Specify the suffix DN for the backend)
dsconf instance replication enable	Enable replication for server	--role (Specify role i.e., master, consumer)
dsconf instance config replace	Modify configuration settings	--attr (Specify the attribute to replace)
dscreate:		
dscreate from-file config.inf	Create new server instance using config file	--force (Overwrite existing instance)
dscreate interactive	Create a new server instance interactively	--accept-license (Auto-accept license)
dsidm:		
dsidm instance user create	Create a new user in the directory	--uid (Specify the user ID)
dsidm instance group add-member	Add a user to a group.	--uid (Specify the user ID to add)
dsidm instance account status	Check the status of a user account.	--uid (Specify the user ID)

ns-slapd

ns-slapd is the primary executable used to start and manage the 389 Directory Server. Those familiar with other 'flavors' of LDAP servers (OpenLDAP) should be familiar with slapd. In 389 Directory Server, ns-slapd essentially serves the same role. It serves as both a server process daemon and can be used as an executable utility to do administrative tasks. Below, some options/flags are similar for both modes but have a different meaning for each- in this case ns-slapd's determination of which mode to execute them in is based on context.

Running ns-slapd as a process/daemon:

Quick example: /usr/sbin/ns-slapd -d 1 -r /var/lib/dirsrv/slapd-instance

-D, --daemon	Run as a background daemon (default).	/usr/sbin/ns-slapd -D
-d or --debug LEVEL	Run in foreground, set 1-9, (higher gives more detail)	/usr/sbin/ns-slapd -d 1
-f or --config DIR	Specify the directory containing the server's config files.	/usr/sbin/ns-slapd -f /path/to/config
-r, --read-only	Start the server in read-only mode (for maintenance)	/usr/sbin/ns-slapd -r
-w, --writenolog	Disable writing to the changelog	/usr/sbin/ns-slapd -w
-n, --no-clean	Do not remove temporary files on exit (for debugging).	/usr/sbin/ns-slapd -n
-F, --no-fsync	Disable fsync for performance (can risk data loss).	/usr/sbin/ns-slapd -F
-i or --instancename NAME	Give name of the server instance to manage.	/usr/sbin/ns-slapd -i instance
-p or --port PORT	TCP port the server listens on (default 389)	/usr/sbin/ns-slapd -p 1389

Running ns-slapd as a executable utility:

Quick example: ns-slapd -D "cn=admin,dc=example,dc=com" -W -a -f users.ldif -r -f delete_entries.ldif -n

-D or --binddn DN	Specify the DN to login with for LDAP/LDIF tasks	ns-slapd -D "cn=admin,dc=example,dc=com"
-W, --prompt	Prompt for password for the bind DN.	ns-slapd -W
-x, --simple	Use simple authentication instead of SASL.	ns-slapd -x
-a, --add	Add entries (useful for importing initial data).	ns-slapd -a
-r, --remove	Remove entries from the directory.	ns-slapd -r
-c, --continue	Continue processing despite errors.	ns-slapd -c
-n, --dry-run	Simulate task without making th task's changes.	ns-slapd -n

Using ldapmodify and ldapsearch

Common to Both ldapsearch and ldapmodify

-x	Use simple authentication (not SASL).	-x
-Y	Specify the SASL mechanism to use.	-Y GSSAPI
-D	Bind DN. Specifies the (admin) DN to login with to make changes, etc.	-D "cn=admin,dc=example,dc=com"
-W	Prompt for password.	-W
-H	Specify the LDAP URI to connect to.	-H "ldap://ldap.example.com"
-b	Base DN. The starting point for the search or modification	-b "dc=example,dc=com"
-LLL	Remove LDIF version lines/comments (e.g., trims off anything but the query answer in search results, etc.)	

Exclusive to ldapsearch

-s	Search scope: base, one, sub.	-s sub
-l	Time limit for the search in seconds.	-l 10
-z	Size limit for the number of entries returned.	-z 500
-E	Enable LDAP extensions.	-E pr=1000/noprompt
-A	Return attribute names only, not values.	-A
-T	Write results to a specified file.	-T /tmp/results.ldif
filter	Search filter expression to match entries.	(uid=jdoe)
attributes	Specify which attributes to return.	cn mail uid
subtree	Scope for subtree search (matches at and below the base DN).	subtree
one	Scope for one-level search (matches one level below the base DN).	one
base	Scope for base object search (limit matches to the base DN).	base

---- *LDAP Search Filter Operators*

`	OR operator for combining multiple search conditions.	
&	AND operator for combining multiple search conditions.	(&(objectClass=posixAccount)(uid=jdoe))
!	NOT operator for negating search conditions.	(!(objectClass=posixAccount))
=	Equality operator for matching attribute values.	(uid=jdoe)
>=	Greater than or equal to for numeric or ordered attribute values.	(uidNumber>=1000)
<=	Less than or equal to for numeric or ordered attribute values.	(uidNumber<=500)
~=	Approximate match operator for attribute values.	(cn~=John)
*	Wildcard operator for matching any attribute value.	(mail=*)

Exclusive to ldapmodify

-a	Add new entries to the directory.	ldapmodify -a -f new_entries.ldif
-c	Continue processing even if errors are encountered.	ldapmodify -c -f update.ldif
-r	Remove entries specified in the LDIF file.	ldapmodify -r -f delete_entries.ldif
-n	Show proposed changes, without applying them.	ldapmodify -n -f changes.ldif
changetype	Specifies type of change: add, modify, delete, or modrdn	changetype: modify
add	Add a new attribute value.	add: mail
delete	Delete an attribute or value.	delete: description
replace	Replace an attribute value.	replace: cn
modrdn	Modify (rename) the relative distinguished name (RDN).	modrdn: newcn
newrdn	New RDN for an entry.	newrdn: cn=Jane Doe
newsuperior	New superior (parent) entry for moving an entry	newsuperior: ou=newdept,dc=example,dc=com
deleteoldrdn	Flag to delete the old RDN value after renaming.	deleteoldrdn: 1

Example of ldapmodify

> ldapmodify -x -D "cn=admin,dc=example,dc=com" -W << EOF dn: uid=user,dc=example,dc=com changetype: modify replace: mail mail: new-email@example.com EOF

This can be better explained breaking it down like this:

ldapmodify -x -D "cn=admin,dc=example,dc=com" -W << EOF

The -D to 'bind' a DN (cn=admin,dc=example,dc=com) of the admin starting the session to do the rest

The -W says to ask for the admin password interactively, the -x says to just use simple authentication (not SASL for this)

The "<< EOF" is simply standard Linux "here document" syntax to send input until the terminating EOF at the end.

Next is the LDIF content:

dn: uid=user,dc=example,dc=com

the DN of the entry to be modified (user entry with uid=user)

changetype: modify

replace: mail

Says we are modifying an existing record, specifically replacing the mail attribute

mail: new-email@example.com

EOF

Lastly we give the replacement value for the mail attribute and close the here doc block with an EOF

Other Commands in 389 DS

slaptest -f	Test server configuration file for errors	slaptest -f /etc/dirsrv/slapd-instance/slapd.conf Use -F to force
slappasswd -s	Generate hashed passwords for server	slappasswd -s secret
dsconf-import	Import LDIF data into the server	dsconf-import -c data.ldif -c Continue on error/ skip bad entries, --dry-run: trial run with no changes.
dsconf-export	Export server data to an LDIF file	dsconf-export --base-dn "dc=example,dc=com" -f export.ldif -b, --base_dn: Specify base DN to export. -f, --file: filename
dslogpipe.py	Process and manage server logs	dslogpipe.py -i /var/log/dirsrv/slapd-instance/access -i, --input filename. -f, --filter: Apply filters to log entries

Finally, just about all commands support

-h, --help	Show help message with available options and usage.
-v, --version	Print the server version and exit.

LDAP Attributes and objectClasses

uid	User ID	uid=jdoe
sn	Surname (Last Name)	sn=Doe
givenname	Given Name (First Name)	givenname=John
cn	Common Name	cn=jdoe,ou=Users,dc=example,dc=com
dn	Distinguished Name	dn=cn=jdoe,ou=Users,dc=example,dc=com
mail	Mail	mail=jdoe@example.com
ou	Organizational Unit	ou=Programmers
department	departmentName	department=IT
title	title	title=Software Engineer
telephonenumber	Telephone Number	telephonenumber=555-123-4567
mobile	mobile	mobile=123-456-7890
o	Organization Name	o=Roxxon
postaladdress	postalAddress	postaladdress=123 Main St, Los Angeles, CA 90028
postalcode	postalCode	postalcode=90028
st	localityName	st=CA
description	description	description=Java and .Net Programming
dc	Domain Component	dc=com
dnsHostName	DNS Host Name	dnsHostName=dns1.example.com

ipHostNumber	IP Address	ipHostNumber=192.168.1.10
macAddress	MAC Address	macAddress=00:11:22:33:44:55
createTimestamp	Creation Timestamp	createTimestamp=20240625120000Z
modifyTimestamp	Modification Timestamp	modifyTimestamp=20240625120000Z
objectclass	Object Class	objectclass=inetOrgPerson,posixAccount,top
domainServer	hypothetical dn for a server	cn=server1,ou=Servers,dc=example,dc=com
inetOrgPerson	Represents a person within an organization	cn=jdoe,ou=Users,dc=example,dc=com mail=jdoe@example.com department=IT
groupOfNames	Container for group members	cn=Network Admins,ou=Groups,dc=example,dc=com member=uid=jdoe,ou=Users,dc=example,dc=com member=uid=jsmith,ou=Users,dc=example,dc=com

The objectClasses are grouping of other attributes and objectClasses, many are premade to chose from, or you can make your own

389 Directory Service quick-setup

Install preliminary packages:
dnf install 389-ds cockpit cockpit-389-ds sssd

Open Firewall Ports
sudo firewall-cmd --permanent --add-port={389/tcp,636/tcp}
sudo firewall-cmd --reload
sudo dnf install cockpit cockpit-389-ds sssd

Enable and start Cockpit (optional, for web-based management):
sudo systemctl enable --now cockpit.socket
sudo firewall-cmd --add-service=cockpit --permanent
sudo firewall-cmd --reload
    # You can access Cockpit at https://<server_ip>:9090.

Make the first directory server instance
- Option 1: Interactive setup, follow prompts to help configure your instance: run "sudo dscreate interactive"
- Option 2: Non-interactive instance creation:
    # First, build a configuration file (.inf) for the instance, at minimum containing this info (replace with yours):

```
---
# /path/to/instance_name.inf

# Specify the desired instance name (leave cn=directory_servers,cn=config as they are)
dn = cn=INSERT_INSTANCE NAME,cn=directory_servers,cn=config

# Define the base DN for your directory data, the root of your directory hierarchy.
#  For example, replace 'your_domain' with actual domain name, then dc=com or net or edu, etc.
directory = dc=YOUR_DOMAIN,dc=com

# Set the administrator password
adminPassword = your_strong_password
---
```

    #Next run this pointing to your new inf file:
sudo dscreate from-file /path/to/instance_name.inf

Prepare for next section "Securing 389 Directory Server"
    # Install these to move onto configuring authentification, authorization, and secure communication for the server.
sudo dnf install openssl openssl-libs cyrus-sasl cyrus-sasl-lib cyrus-sasl-gssapi cyrus-sasl-md5 krb5-workstation

Securing 389-DS with OpenSSL, Cyrus SASL, Kerberos

After creating your first instance in 389-DS, you can install and set up these, then apply to others as added.
Install: sudo dnf install openssl openssl-libs cyrus-sasl cyrus-sasl-lib cyrus-sasl-gssapi cyrus-sasl-md5 krb5-workstation

Configuring SSL/TLS for 389 DS
Obtain/ generate SSL/TLS Certificates:
If you don't have one, you can generate a self-signed certificate for testing purposes or get one from a trusted CA (Let's Encrypt)

To generate a self-signed certificate:
openssl req -new -x509 -days 365 -nodes -out /etc/dirsrv/slapd-INSTANCE1/ca.crt -keyout /etc/dirsrv/slapd-INSTANCE1/ca.key
Replace INSTANCE1 with your instance name. This creates a self-signed certificate and private key valid for 365 days.
Place your certificates in the appropriate directory for your 389 DS instance, usually under /etc/dirsrv/slapd-INSTANCE1/
[Note there is another option, but you have less granular control about the certificate: "sudo dsctl example tls generate-self-signed-cert --subject "/CN=example.com"]

Configure 389 Directory Server for SSL/TLS:
dsconf -D "cn=Directory Manager" ldaps://localhost ssl set --enable true
dsconf -D "cn=Directory Manager" ldaps://localhost ssl cert --import --file /etc/dirsrv/slapd-INSTANCE1/ca.crt
dsconf -D "cn=Directory Manager" ldaps://localhost ssl key --import --file /etc/dirsrv/slapd-INSTANCE1/ca.key --password PASSWD

Update the Directory Server Configuration:
Edit the dse.ldif configuration file to enable SSL with 'sudo nano /etc/dirsrv/slapd-INSTANCE1/dse.ldif'
Add or update the following entries, set the correct certificate paths:

```
dn: cn=encryption,cn=config
nsslapd-security: on
nsslapd-securePort: 636
nsslapd-ssl-check-hostname: on
nsslapd-certdir: /etc/dirsrv/slapd-INSTANCE1
nsslapd-certname: Server-Cert
nsslapd-certfile: ca.crt
nsslapd-keyfile: ca.key
```

Save and close the file, then restart the server with 'sudo systemctl restart dirsrv@INSTANCE1'

Configuring SASL for 389 Directory Server
Enable SASL in the 389 Directory Server - edit the dse.ldif file with 'sudo nano /etc/dirsrv/slapd-INSTANCE1/dse.ldif'
Add or modify the following entries:

```
dn: cn=config
nsslapd-sasl-maps: on
nsslapd-sasl-max-buffer-size: 65536
nsslapd-sasl-secprops: noanonymous,noplain,novalidate
```

Configure SASL Mechanisms:
Create/edit the /etc/sasl2/slapd.conf file to specify SASL options. Note the keytab location for Kerberos setup (the next step)

```
pwcheck_method: saslauthd
mech_list: GSSAPI DIGEST-MD5 CRAM-MD5
keytab: /etc/dirsrv/slapd-INSTANCE1/ldap.servername.keytab
```

Setup Kerberos for SASL/GSSAPI:
Edit /etc/krb5.conf and replace the following with your kerberos server's information:

```
[libdefaults]
   default_realm = YOUR.REALM
   dns_lookup_realm = false
   dns_lookup_kdc = true
[realms]
   YOUR.REALM = {
      kdc = kdc.example.com
      admin_server = kdc.example.com
   }
[domain_realm]
   .example.com = YOUR.REALM
   example.com = YOUR.REALM
```

Create a service principal for 389 DS by running ktpasswd (generaties key and stores the keytab where directed)

```
Be sure pathname to keytab is matched in /etc/sasl2/slapd.conf (noted above)
ktpasswd -q -h /etc/dirsrv/slapd-INSTANCE1/ldap.servername.keytab ldap/servername@example.com
```

For this next step, make sure there is a username on the Kerberos server called 389-LDAPsetup (for this example) or something to be a admin placeholder username for the ongoing server key usage and initial setup of LDAP admin users.
Use kinit to obtain a ticket (basically kerberos-username-for-server@REALM)

kinit 389-LDAPsetup@example.com

Testing the Configuration

Verify SSL: ldapsearch -x -H ldaps://localhost -b "dc=example,dc=com"
Test SASL: ldapsearch -Y GSSAPI -H ldap://localhost -b "dc=example,dc=com" "(objectclass=*)"

More items to improve security of the 389 Directory Server

Restrict LDAP access from only the secure port (LDAPS)

```
#Put in slapd.conf
dn: cn=config
nsslapd-listenhost: localhost
nsslapd-port: 636  # Use port 636 for LDAPS
```

Save and restart the directory server instance:

```
sudo systemctl restart dirsrv@<instance_name>
```

Configure Access Control Lists (ACLs)

```
# Define ACLs to control access to the directory data, maintain data integrity and security.
# Example ACL to allow read access to all users but restrict write access to admins:
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr != "userPassword")(version 3.0; acl "Allow read"; allow (read, search, compare) userdn="ldap:///self";)
aci: (targetattr="*")(version 3.0; acl "Admin write"; allow (all) groupdn="ldap:///cn=admins,dc=example,dc=com";)

# Save this to a file acl.ldif, then, apply the ACL using:
ldapmodify -x -D "cn=Directory Manager" -W -f acl.ldif
# The content of the ACL is saved in attributes of the target dn, in this case  dn: dc=example,dc=com
```

Enable password policies to enforce complexity, expiration, and lockout rules.

Create a file password_policy.ldif with the desired policies:

```
dn: cn=config
changetype: modify
replace: nsslapd-pwpolicy
nsslapd-pwpolicy: on

dn: cn=default,ou=pwpolicies,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
pwdAttribute: userPassword
pwdMaxAge: 7776000  # 90 days
pwdMinLength: 8
pwdCheckSyntax: 1  # Enforce complexity rules
pwdInHistory: 5  # Remember past passwords
pwdLockout: TRUE
pwdLockoutDuration: 900  # 15 minutes lockout
```

Apply the policy:

```
ldapmodify -x -D "cn=Directory Manager" -W -f password_policy.ldif
```

Enable and configure logging for access and errors (edit dse.ldif):

```
dn: cn=config
nsslapd-accesslog: /var/log/dirsrv/slapd-example/access
nsslapd-errorlog: /var/log/dirsrv/slapd-example/errors
# Ensure logs are rotated and reviewed regularly:
sudo logrotate /etc/logrotate.d/dirsrv
```

Enable SNMP monitoring using dsconf:

```
sudo dsconf example config replace nsslapd-schemachecking=on
sudo dsconf example config replace snmp-port=199
```

Implement disaster recovery using dsconf to perform a backup

```
sudo dsconf example backend export --base-dn "dc=example,dc=com" /path/to/backup
```

Core Configuration and Settings for 389-Directory Server

Main Configuration (slapd.conf):

```
        # General settings
instance-name my-ldap-server  # Replace with your desired instance name
suffix "dc=example,dc=com"  # Replace with your domain name
        # Directory database configuration
database: mdb_directory  # Database type (Modify based on your setup)
index objectClass, cn  # Attributes to be indexed
        # Network settings
listen-address 0.0.0.0  # Listen on all interfaces
listen-port 389  # Standard LDAP port
        # Include additional configuration files
include "user_management.conf"
include "security.conf"
include "backup.conf"
```

User Management (user_management.conf):

```
        # User account settings
baseDn "ou=People,dc=example,dc=com"  # Base DN for user accounts
password-min-length 8
password-require-mixed-case TRUE
password-history-length 5
account-lockout-threshold 5
account-lockout-duration 30m
groupDn "ou=Groups,dc=example,dc=com"  # Base DN for groups
# password-expire: #days  # Enable password expiry after a set number of days
# user-quota: #bytes  # Set quota on the size of user entries
# dynamic-groups:  # Configure rules for dynamic group membership
```

Security Settings (security.conf):

```
        # Password complexity rules
require-mixed-case TRUE
password-minimum-length 12
require-numbers TRUE
require-special-characters TRUE
        # TLS/SSL encryption
tls-enabled TRUE
tls-certificate-file /path/to/server.crt
tls-key-file /path/to/server.key
        # Access control (modify based on your needs)
access to by dn "cn=Directory Manager,ou=People,dc=example,dc=com" read
        # Additional security options (uncomment and configure as needed)
# access-control-list:  # Define granular access control for specific DNs
# security-auditing:  # Enable security auditing for directory operations
```

Backup Configuration (backup.conf):

```
        # Backup settings
backup-frequency daily
backup-directory /var/lib/389-ds/backups
backup-retention 7
        # Backup tool configuration (replace 'tool' with your chosen tool)
backup-tool "tar"
backup-arguments "-cvzf"
```

Logging Configuration (logging.conf): (Optional, can be included in the main conf)

```
        # General logging level
log-level info
        # Specific log levels for components (uncomment and adjust as needed)
# slapd-level debug
# slapd-modules-level warn
        # Log rotation settings
log-rotate-size 10m  # Rotate logs when they reach 10MB
log-rotate-count 5  # Keep the last 5 rotated logs
        # Additional logging options (uncomment and configure as needed)
# remote-logging: # Configure remote logging to a central server
```

Tips on improving the 389 Directory Server implementation moving forward

Automated Instance Creation with a Standardized Configuration File
Streamline setup - Make clones from a config instance.inf file with the following content:

```
[general]
config_version = 2

[slapd]
instance_name = auto-made-instance
root_dn = cn=Directory Manager
root_password = <STRONG_PASSWORD>
server_port = 389
suffix = dc=example,dc=com
self_sign_cert = True  # Automatically generate a self-signed certificate

[backend-userroot]
database_name = userroot
suffix = dc=example,dc=com
```

```
        # Create the directory server instance on-demand and ensure the instance is running and verify the status:
sudo dscreate from-file instance.inf
sudo dsctl auto-made-instance status
```

Configure Backend Databases
Splitting data into separate backend databases improves load management and scalability. Add a new backend for organizational units. Add this to dse.ldif or use dsconf to configure dynamically.

```
[backend-newunit]
database_name = newunit
suffix = ou=newunit,dc=example,dc=com
```

Configure Indexing for Enhansing Performance
Use the following dsconf commands to index commonly searched attributes without the need to manually edit dse.ldif

```
        # Index uid for equality searches:
sudo dsconf example backend index create --attr-name uid --types eq
        # Index mail for equality and presence searches:
sudo dsconf example backend index create --attr-name mail --types eq pres
        # Index sn (Surname) for equality and presence searches:
sudo dsconf example backend index create --attr-name sn --types eq pres
```

Optimize Database Cache Settings - Enhance performance by tuning the database cache

```
sudo dsconf example backend config set --db-cache-size 512MB
sudo dsconf example backend config set --entry-cache-size 512MB
```

Replication for High Availability - Multi-master replication ensures data redundancy and HA

```
        # Primary Server Configuration (edit dse.ldif or use dsconf):
dn: cn=replica,cn="dc=example,dc=com",cn=mapping tree,cn=config
objectClass: top
objectClass: nsDS5Replica
nsDS5ReplicaRoot: dc=example,dc=com
nsDS5ReplicaId: 1
nsDS5ReplicaType: 3
nsDS5Flags: 1
nsDS5ReplicaBindDN: cn=Replication Manager,cn=config
```

Secondary Server Configuration:
Configure similarly, but with a unique nsDS5ReplicaId (e.g., 2).

```
        # Establish replication relationships:
sudo dsconf example repl-agmt create --suffix="dc=example,dc=com" --host=<secondary-server-ip> --port=389 --bind-
dn="cn=Replication Manager,cn=config" --bind-passwd=<bind_password> --conn-protocol=LDAP --init
```

Repeat on the secondary server pointing back to the primary server.