Визитка

СЕРГЕЙ ГОЛОВАШОВ,
ведущий инженер DevOps, руководитель центра компетенций, Bell Integrator

Визитка

СЕРГЕЙ САДОВ,
ведущий инженер DevOps, Bell Integrator

OpenIdap – от кубернетиса до серьезного промышленного решения

В статье рассказано про развертку такого решения, как OpenLdap. Сервер каталогов — идеальное средство для решения задачи каталогизации как сотрудников предприятия, так и его активов. Осталось определиться с задачами и масштабом реализации.

OpenLDAP — это открытая реализация LDAP каталога, разработанная одноимённым проектом, распространяется под собственной свободной лицензией OpenLDAP Public License.

Что представляет собой LDAP изнутри? Для чего он нужен и как к нему подступиться, чтобы развернуть его в Кубере? LDAP – это база данных имеющая древовидную структуру (называемую organization units) и содержащая любой тип структурированных однотипных данных (например, карточку сотрудника).

Сам по себе LDAP получил развитие также в специфических ответвлениях, которые имеют более узкую специализацию, например, CMDB (инвентари информации об установленном ПО в организации).

История LDAP

Начало проекту было положено в 1998 году Куртом Зейленгой (*Kurt Zeilenga*). Изначальный код OpenLDAP был скопирован с реализации LDAP Мичиганского университета, где в конечном счёте и была продолжена разработка и эволюция протокола LDAP. В апреле 2006 года главными разработчиками проекта OpenLDAP были: Ховард Чу (*Howard Chu*) (главный архитектор), Пьеранджело Мазарати (*Pierangelo Masarati*) и Курт Зейленга. Значительный вклад в проект внесли Люк Ховард (*Luke Howard*), Хальвард Фурусет (*Hallvard Furuseth*), Кана Гибсон-Маунт (*Quanah Gibson-Mount*) и Гэвин Хенри (*Gavin Henry*).

В 2015 году, с целью использования в инфраструктуре ПАО МегаФон, российскими разработчиками был создан форк ReOpenLDAP, в котором устранено несколько принципиальных ошибок в реализации механизма репликации (синхронизации содержимого) по RFC 4533, а также добавлены средства для преодоления недостатков самого протокола (некорректного удаления записей в репликах). В результате чего становится возможным построение кластера из трёх и более узлов в режиме мульти-мастер с топологией «каждый с каждым» (full mesh topology).

Приступаем к установке

Мы развернем OpenLDAP с помощью следующего скрипта:

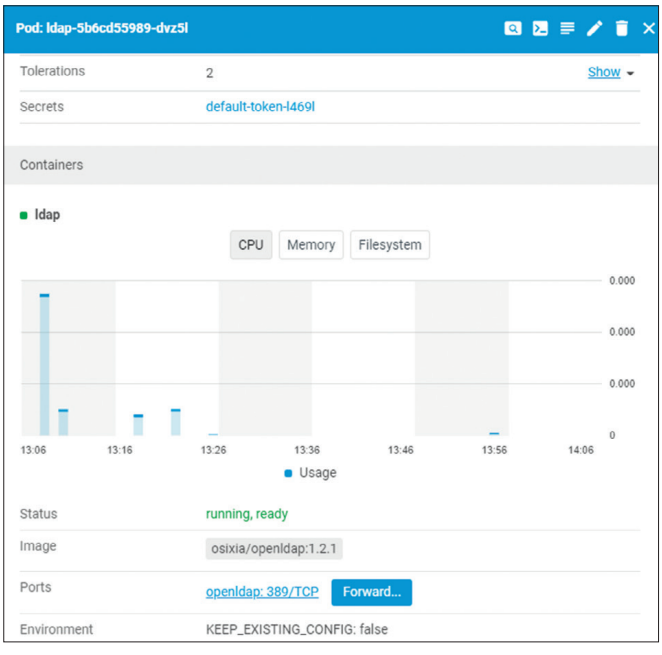
```
apiVersion: apps/v1
kind: Deployment
metadata:
   name: ldap
   labels:
      app: ldap
spec:
   replicas: 1
   selector:
      matchLabels:
         app: ldap
   template:
      metadata:
         creationTimestamp: null
         labels:
            app: ldap
      spec:
         containers:
            - name: ldap
              image: osixia/openldap:1.2.1
              volumeMounts:
                 - name: ldap-data
                   mountPath: /var/lib/ldap
                 - name: ldap-config
                   mountPath: /etc/ldap/slapd.d
                 - name: ldap-certs
                   mountPath: /container/service/slapd/
assets/certs
              ports:
                 - containerPort: 389
                   name: openldap
              env:
                 - name: LDAP_LOG_LEVEL
                   value: "256"
                 - name: LDAP_ORGANISATION
                   value: "Example Inc."
                 - name: LDAP_DOMAIN
                   value: "example.org"
                 - name: LDAP_ADMIN_PASSWORD
                   value: "admin"
                 - name: LDAP_CONFIG_PASSWORD
                   value: "config"
                 - name: LDAP_READONLY_USER
                   value: "false"
                 - name: LDAP_READONLY_USER_USERNAME
```

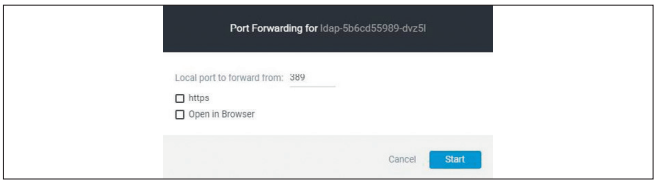
LDAP — это база данных, имеющая древовидную структуру (называемую organization units) и содержащая любой тип структурированных однотипных данных

```
                  value: "readonly"
               - name: LDAP_READONLY_USER_PASSWORD
                  value: "readonly"
               - name: LDAP_RFC2307BIS_SCHEMA
                  value: "false"
               - name: LDAP_BACKEND
                  value: "mdb"
               - name: LDAP_TLS
                  value: "true"
               - name: LDAP_TLS_CRT_FILENAME
                  value: "ldap.crt"
               - name: LDAP_TLS_KEY_FILENAME
                  value: "ldap.key"
               - name: LDAP_TLS_CA_CRT_FILENAME
                  value: "ca.crt"
               - name: LDAP_TLS_ENFORCE
                  value: "false"
               - name: LDAP_TLS_CIPHER_SUITE
                  value: "SECURE256:+SECURE128:-VERS-TLS-ALL:
+VERS-TLS1.2:-RSA:-DHE-DSS:-CAMELLIA-128-CBC:-CAMELLIA-256-CBC"
               - name: LDAP_TLS_VERIFY_CLIENT
                  value: "demand"
               - name: LDAP_REPLICATION
                  value: "false"
               - name: LDAP_REPLICATION_CONFIG_SYNCPROV
                  value: "binddn=\"cn=admin,cn=config\
" bindmethod=simple credentials=$LDAP_CONFIG_PASSWORD
searchbase=\"cn=config\" type=refreshAndPersist retry=\
"60 +\" timeout=1 starttls=critical"
               - name: LDAP_REPLICATION_DB_SYNCPROV
                  value: "binddn=\"cn=admin,$LDAP_BASE_
DN\" bindmethod=simple credentials=$LDAP_ADMIN_PASSWORD
searchbase=\"$LDAP_BASE_DN\" type=refreshAndPersist
interval=00:00:00:10 retry=\"60 +\" timeout=1 starttls=critical"
               - name: LDAP_REPLICATION_HOSTS
                  value: "#PYTHON2BASH:['ldap://ldap-one-
service', 'ldap://ldap-two-service']"
               - name: KEEP_EXISTING_CONFIG
                  value: "false"
               - name: LDAP_REMOVE_CONFIG_AFTER_SETUP
                  value: "true"
               - name: LDAP_SSL_HELPER_PREFIX
                  value: "ldap"
         volumes:
            - name: ldap-data
              hostPath:
                 path: "/data/ldap/db"
            - name: ldap-config
              hostPath:
                 path: "/data/ldap/config"
            - name: ldap-certs
              hostPath:
                 path: "/data/ldap/certs"
---
apiVersion: v1
kind: Service
metadata:
   labels:
      app: ldap
   name: ldap-service
spec:
   ports:
      - port: 389
   selector:
      app: ldap
```

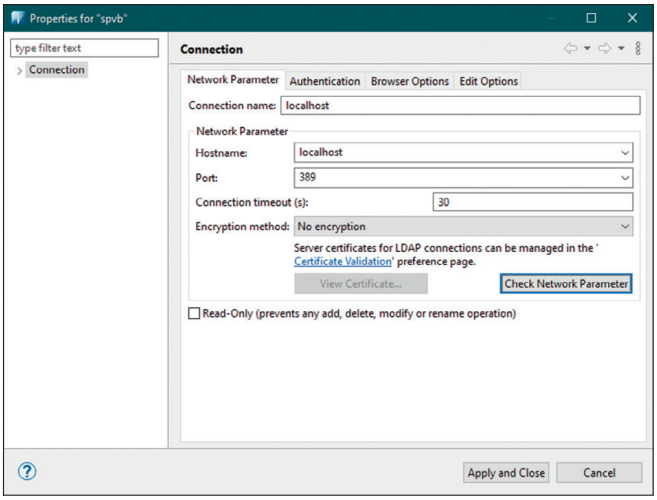
Как вы видите, мы подняли сам контейнер и сервис к нему. А теперь ставим Apache Directory Studio с сайта разработчика https://directory.apache.org/studio и подключимся к самому LDAP.

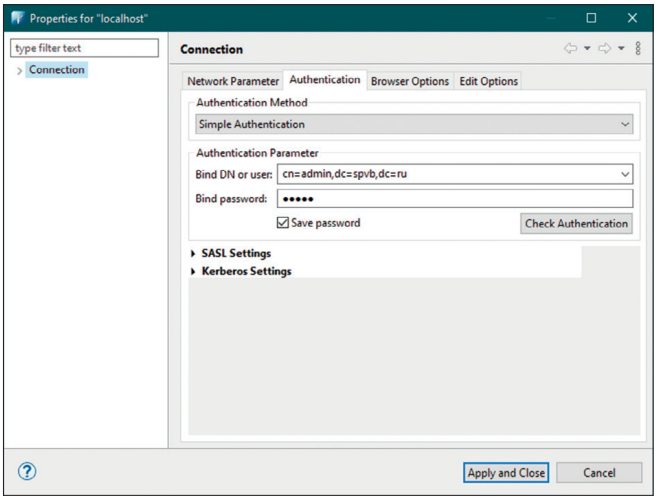
Для настройки подключения пробросим порт из работающего pod с LDAP-сервером на нашу локальную машину. Сделаем это в Lens:



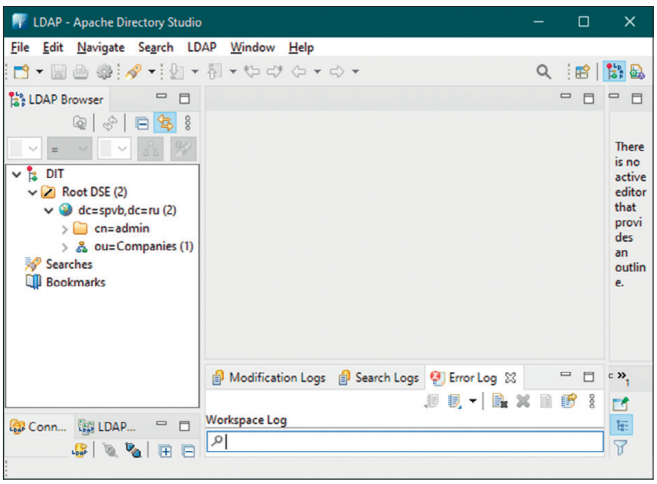


Переходим к созданию соединения в Apache Directory Studio:





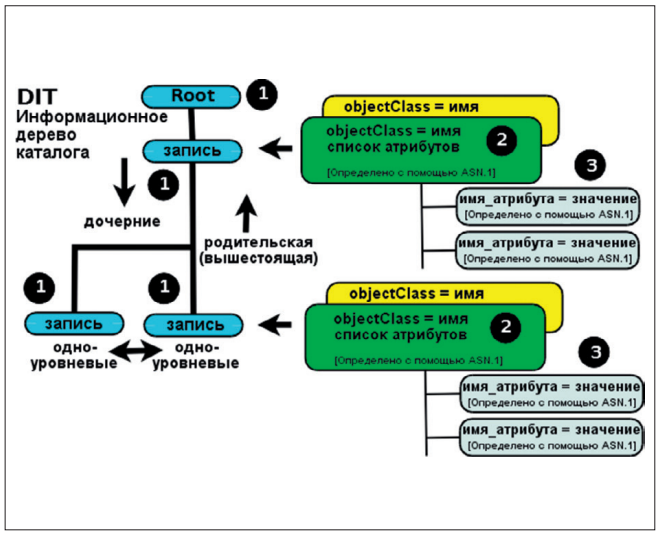
В итоге мы подключились к нашему LDAP-серверу и готовы создавать нашу первую структуру с отделами и пользователями.



Администрирование OpenLDAP, создание схемы и записей

Данные в LDAP – это древовидная иерархия объектов, называемых записями - DIT Directory Information Tree (информационное дерево каталога), и начинающихся с корня (root).

Каждая запись является экземпляром одного или нескольких объектных классов (objectClass). Объектные классы содержат атрибуты (attribute). Атрибуты имеют имена и содержат данные.



Существует большое количество предопределённых объектных классов с атрибутами для почти всех возможных применений каталогов LDAP. Однако бывает так, что среди них нет того, который необходим для описания вашей конкретной структуры.

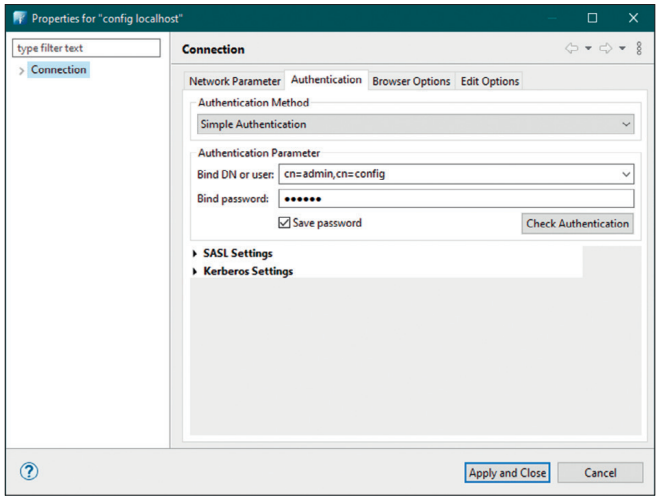
Добавление записей может происходить разными путями, один из которых — использовать файлы формата LDAP Data Interchange Files (LDIF) Файлы LDIF — это текстовые файлы, описывающие DIT, и данные, добавляемые в каждый атрибут.

Ниже приведён LDIF-файла, описывающий корневое DN (dc=spvb,dc=ru) и добавляющий нашу структуру:

```
dn: dc=spvb,dc=ru
objectClass: dcObject
objectClass: organization
objectClass: top
dc: spvb
o: SPVB

dn: ou=Companies,dc=spvb,dc=ru
objectClass: organizationalUnit
objectClass: top
ou: Companies

dn: ou=SPVB,ou=Companies,dc=spvb,dc=ru
objectClass: organizationalUnit
objectClass: top
ou: SPVB

dn: ou=Users,ou=SPVB,ou=Companies,dc=spvb,dc=ru
objectClass: organizationalUnit
objectClass: top
ou: Users

dn: ou=Computers,ou=SPVB,ou=Companies,dc=spvb,dc=ru
objectClass: organizationalUnit
objectClass: top
ou: Computers
```

Добавление записей может происходить разными путями, один из которых — использовать файлы формата LDAP Data Interchange Files (LDIF) Файлы LDIF — это текстовые файлы, описывающие DIT, и данные, добавляемые в каждый атрибут

Прежде чем добавлять пользователей в созданную службу каталога, нам необходимо создать свой objectClass с недостающими конкретно нам атрибутами. Сделаем это, выполнив LDIF-файл, подключившись к LDAP с правами на изменение конфигурации:



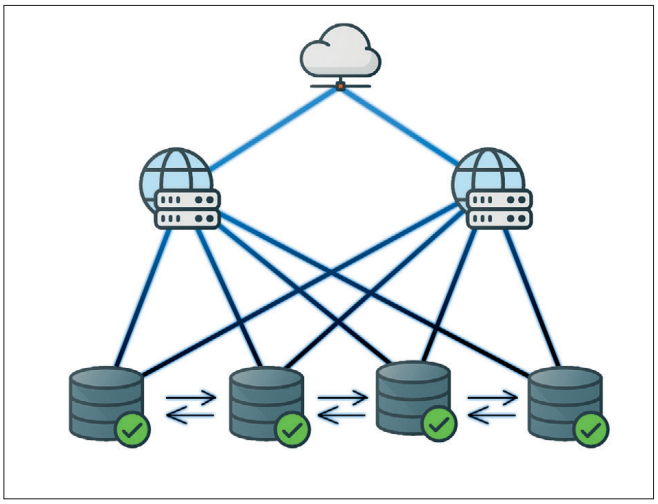

```
dn: cn=spvbPerson,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: spvbPerson
olcAttributeTypes: ( 1.1.1.1.1
   NAME 'CannotChangePassword'
   EQUALITY booleanMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.7)
olcAttributeTypes: ( 1.1.1.1.2
   NAME 'Created'
   EQUALITY caseExactMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
olcObjectClasses: ( 1.1.1.1.0
   NAME 'spvbPerson'
   DESC 'spvbPerson'
   SUP top
   AUXILIARY
   MAY ( CannotChangePassword $ Created ) )
```

Наконец, добавим первого пользователя:

```
dn: uid=Иванов Иван Иванович,ou=Users,ou=SPVB,ou=Companies,
dc=spvb,dc=ru
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: spvbPerson
cn: Иванов Иван Иванович
sn: Иванов
uid: Иванов Иван Иванович
userPassword:
CannotChangePassword: FALSE
mail: ivanov@mail.ru
Created: 04.12.2022 15:26:59
```

Переходим к серьезным вещам: разворачиваем отказоустойчивый кластер

Ну вот мы плавно и подошли к развертке нашего OpenLDAP в кластере для промышленной эксплуатации. Схема будет следующая:



Два сервера LDAP в режиме Cache proxy. И четыре сервера, непосредственно хранящие у себя базу данных и синхронизирующие её между собой.

Итак, установка OpenLDAP (для всех серверов эта команда одинакова):

```
yum -y install openldap compat-openldap openldap-clients
openldap-servers openldap-servers-sql openldap-devel
nss-pam-ldapd
```

А теперь приступаем к серверам кэша.

Сгенерим SHA-пароль, который будет использоваться для авторизации на всех узлах нашего кластера: slappasswd -s <пароль>

Запишите его отдельно! Все переменные %PASSWORD% далее по тексту нужно будет заменить на данный пароль, а зашифрованный пароль вставлять вместо %SHAPASSWORD%! Переменная %NODE% заменяется на адреса серверов в виде ldap:\\192.168.1.150, несколько адресов пишутся через пробел, например: ldap:\\192.168.1.150 ldap:\\192.168.1.151 ldap:\\192.168.1.153 ldap:\\192.168.1.154

Готовим файл настроек прокси кэш сервера slapd.conf:

```
### Schema includes ######################################
include             /etc/openldap/schema/core.schema
include             /etc/openldap/schema/cosine.schema
include             /etc/openldap/schema/inetorgperson.schema
include             /etc/openldap/schema/misc.schema
include             /etc/openldap/schema/nis.schema
include             /etc/openldap/schema/ffperson.schema

allow bind_v2

## Module paths ##########################################
modulepath                  /usr/lib64/openldap/
```

```
moduleload                  back_ldap
moduleload                  rwm
moduleload                  pcache
moduleload                  memberof

# Main settings ##############################################
pidfile                      /var/run/openldap/slapd.pid
argsfile                    /var/run/openldap/slapd.args

### Database definition (Proxy to AD) ######################
database                   ldap
readonly                   no
protocol-version           3
rebind-as-user
uri                         "%NODE%"
suffix                     "dc=organization,dc=ru"
rootdn                     "cn=admin,dc=organization,dc=ru"
rootpw                     %SHAPASSWORD%

#overlay             rwm
#rwm-map             attribute   uid       sAMAccountName
#rwm-map             attribute   mail     proxyAddresses
#rwm-map             attribute   homeDirectory UNIXHOMEDIRECTORY
#rwm-map             objectclass posixGroup     group

overlay              pcache
pcache               hdb 100000 3 1000 100
pcachePersist        TRUE
pcacheAttrset        0 1.1
pcacheTemplate       (&(|(objectClass=))) 0 3600
pcacheTemplate       (objectClass=*) 0 3600
# User Name Field (Advanced Tab)
#pcacheAttrset       1 displayname
#pcacheTemplate      (objectClass=*) 1 3600
# Group Field
pcacheAttrset        2 memberOf
pcacheTemplate       (objectClass=*) 2 3600

overlay              memberof

access to *
     by self write
          by anonymous none
               by * read

directory                    /var/lib/ldap


### Logging ##################################################
loglevel                     0


chase-referrals          no
idassert-bind          bindmethod=simple
                       mode=self
                       binddn="cn=admin,dc=ffin,dc=ru"
                       credentials="%PASSWORD%"
idassert-authzFrom        "dn.regex:.*"

access                          to *
                                   by * write
```

Если у вас уже есть подготовленные схемы БД, то копируем их в /etc/openldap/schema/, не забывая при этом прописать путь с названием в начале файла конфигурации.

Добавляем в автозапуск через rc.local. Кто-то может сделать это через systemd, но мы для упрощения делаем это так:

```
echo "/usr/sbin/slapd" >> /etc/rc.d/rc.local
/usr/sbin/slapd
```

Переходим к настройке серверов, которые отвечают за хранение непосредственно самой базы данных, и готовим их файл конфигурации slapd.conf.

В значении %SERVERID% прописываем уникальный номер каждого узла, начиная с 1. %SHAPASSWORD%, как и выше это наш зашифрованный пароль.

```
include             /etc/openldap/schema/core.schema
include             /etc/openldap/schema/cosine.schema
include             /etc/openldap/schema/inetorgperson.schema
include             /etc/openldap/schema/misc.schema
include             /etc/openldap/schema/nis.schema
include             /etc/openldap/schema/ffperson.schema

modulepath          /usr/lib64/openldap/
moduleload          syncprov
moduleload          back_ldap
moduleload    memberof

serverID    %SERVERID%

database mdb
suffix "dc=organization,dc=ru"
checkpoint 32 30
maxsize 10485760
rootdn "cn=admin,dc=organization,dc=ru"
rootpw "%SHAPASSWORD%"
directory "/var/lib/ldap"
index objectClass eq

access to *
      by self write
      by anonymous none
      by * read


overlay memberof

overlay syncprov
syncprov-checkpoint 100 10
syncprov-sessionlog 100
```

На этом конфигурация сервера не заканчивается. Теперь его необходимо связать со всеми остальными. Это делается следующим образом:

```
syncrepl rid=002 provider=ldap://192.168.1.152 searchbase=
"dc=organization,dc=ru" binddn="cn=admin,dc=organization,
dc=ru" interval=00:00:02:00 bindmethod=simple credentials=
%PASSWORD% schemachecking=on type=refreshAndPersist retry="60 +"
syncrepl rid=003 provider=ldap:///192.168.1.153 searchbase=
"dc=organization,dc=ru" binddn="cn=admin,dc=organization,
dc=ru" interval=00:00:02:00 bindmethod=simple credentials=
%PASSWORD% schemachecking=on type=refreshAndPersist retry="60 +"
syncrepl rid=003 provider=ldap:///192.168.1.154 searchbase=
"dc=organization,dc=ru" binddn="cn=admin,dc=organization,
dc=ru" interval=00:00:02:00 bindmethod=simple credentials=
%PASSWORD% schemachecking=on type=refreshAndPersist retry="60 +"
```

Тем самым, мы связываем сервер 192.168.1.150 с тремя оставшимися и говорим, что синхронизация бд dc=organization,dc=ru должна осуществляться с этими узлами. Это необходимо сделать на каждом узле базы данных.

По сути, конечно, это не кластер, это четыре самостоятельных сервера, между которыми включен механизм синхронизации. Но это то, что нам предлагает сам разработчик.

Надеемся, вам эта статья поможет развернуть OpenLDAP и успешно с ним управляться в вашей работе на кубернетисе при решении больших задач построения отказоустойчивой инфраструктуры на предприятии. EOF

Ключевые слова: Openldap, кластер, кубернетис, создание схем, администрирование базы данных, инструменты администрирования