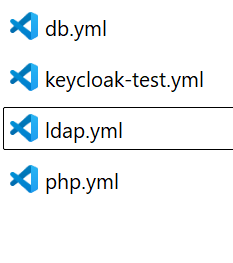
Запускаем миникуб

Создаём 4 файла .yml



db.yml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: postgres-pvc

spec:

  accessModes:

    - ReadWriteOnce

  resources:

    requests:

      storage: 8Gi

---

apiVersion: apps/v1

kind: Deployment

metadata:

  name: postgres-deployment

spec:

  replicas: 1

  selector:

    matchLabels:

      app: postgres

  template:

    metadata:

      labels:

        app: postgres

    spec:

      containers:

      - name: postgres

        image: postgres:latest

        env:

        - name: POSTGRES\_DB

          value: keycloakdb

        - name: POSTGRES\_USER

          value: keycloakuser

        - name: POSTGRES\_PASSWORD

          value: keycloakpass

        volumeMounts:

        - mountPath: /var/lib/postgresql/data

          name: postgredb

      volumes:

      - name: postgredb

        persistentVolumeClaim:

          claimName: postgres-pvc

---

apiVersion: v1

kind: Service

metadata:

  name: postgres-service

spec:

  type: ClusterIP

  selector:

    app: postgres

  ports:

  - port: 5432

    targetPort: 5432

keycloak-test.yml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: keycloak-deployment

spec:

  replicas: 1

  selector:

    matchLabels:

      app: keycloak

  template:

    metadata:

      labels:

        app: keycloak

    spec:

      containers:

      - name: keycloak

        image: bitnami/keycloak:latest

        env:

        - name: DB\_VENDOR

          value: "POSTGRES"

        - name: DB\_ADDR

          value: "postgres-service" # Имя сервиса PostgreSQL

        - name: DB\_DATABASE

          value: "keycloakdb"

        - name: DB\_SCHEMA

          value: "public"

        - name: DB\_USER

          value: "keycloakuser"

        - name: DB\_PASSWORD

          value: "keycloakpass"

        - name: KEYCLOAK\_ADMIN

          value: "admin"

        - name: KEYCLOAK\_ADMIN\_PASSWORD

          value: "123"

        - name: KEYCLOAK\_HTTP\_PORT

          value: "8180"

        ports:

        - containerPort: 8180

---

apiVersion: v1

kind: Service

metadata:

  name: keycloak-service

spec:

  type: NodePort

  selector:

    app: keycloak

  ports:

  - protocol: TCP

    port: 8180

    targetPort: 8180

    nodePort: 30080

ldap.yml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: openldap

spec:

  replicas: 1

  selector:

    matchLabels:

      app: openldap

  template:

    metadata:

      labels:

        app: openldap

    spec:

      containers:

      - name: openldap

        image: osixia/openldap:latest

        ports:

        - containerPort: 389

        - containerPort: 636

        volumeMounts:

        - mountPath: /container/service/slapd/assets/certs

          name: certificates-volume

        - mountPath: /var/lib/ldap

          name: database-volume

        - mountPath: /etc/ldap/slapd.d

          name: config-volume

        env:

        - name: LDAP\_ORGANISATION

          value: ramhlocal

        - name: LDAP\_DOMAIN

          value: ramhlocal.com

        - name: LDAP\_ADMIN\_USERNAME

          value: admin

        - name: LDAP\_ADMIN\_PASSWORD

          value: admin\_pass

        - name: LDAP\_CONFIG\_PASSWORD

          value: config\_pass

        - name: LDAP\_BASE\_DN

          value: dc=ramhlocal,dc=com

        - name: LDAP\_TLS\_CRT\_FILENAME

          value: server.crt

        - name: LDAP\_TLS\_KEY\_FILENAME

          value: server.key

        - name: LDAP\_TLS\_CA\_CRT\_FILENAME

          value: ramhlocal.com.ca.crt

        - name: LDAP\_READONLY\_USER

          value: "true"

        - name: LDAP\_READONLY\_USER\_USERNAME

          value: user-ro

        - name: LDAP\_READONLY\_USER\_PASSWORD

          value: ro\_pass

      volumes:

      - name: certificates-volume

        hostPath:

          path: /path/to/local/data/certificates

      - name: database-volume

        hostPath:

          path: /path/to/local/data/slapd/database

      - name: config-volume

        hostPath:

          path: /path/to/local/data/slapd/config

---

apiVersion: v1

kind: Service

metadata:

  name: openldap-service

spec:

  type: NodePort

  selector:

    app: openldap

  ports:

  - name: ldap-port

    protocol: TCP

    port: 389

    targetPort: 389

    nodePort: 30089

  - name: ldaps-port

    protocol: TCP

    port: 636

    targetPort: 636

    nodePort: 30636

php.yml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: phpldapadmin

spec:

  replicas: 1

  selector:

    matchLabels:

      app: phpldapadmin

  template:

    metadata:

      labels:

        app: phpldapadmin

    spec:

      containers:

      - name: phpldapadmin

        image: osixia/phpldapadmin:latest

        ports:

        - containerPort: 80

        env:

        - name: PHPLDAPADMIN\_LDAP\_HOSTS

          value: openldap-service

        - name: PHPLDAPADMIN\_HTTPS

          value: "false"

---

apiVersion: v1

kind: Service

metadata:

  name: phpldapadmin-service

spec:

  type: NodePort

  selector:

    app: phpldapadmin

  ports:

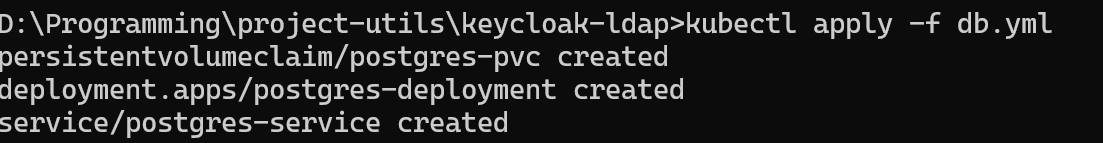
  - protocol: TCP

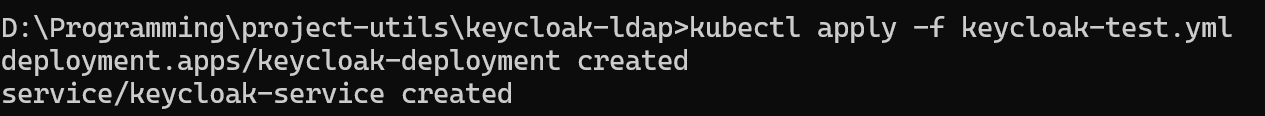
    port: 80

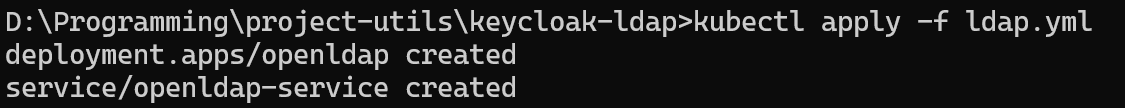
    targetPort: 80

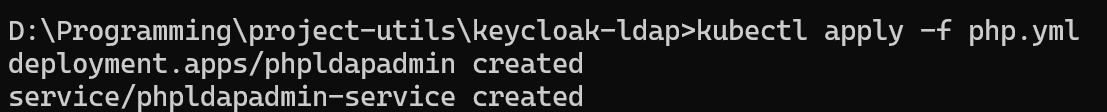
    nodePort: 31080

Применяем эти файлы для kubectl

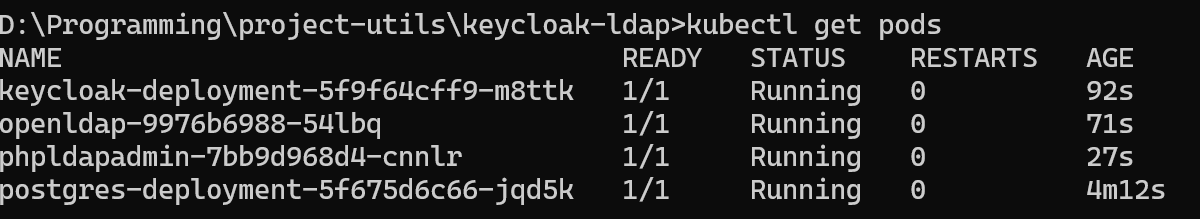




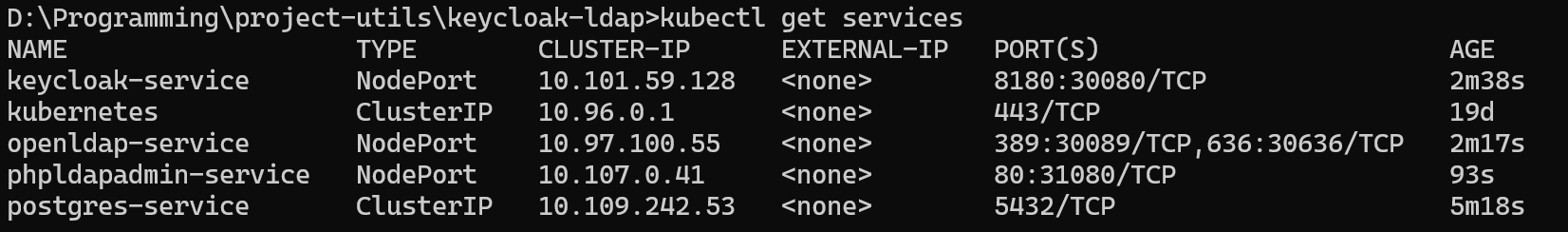




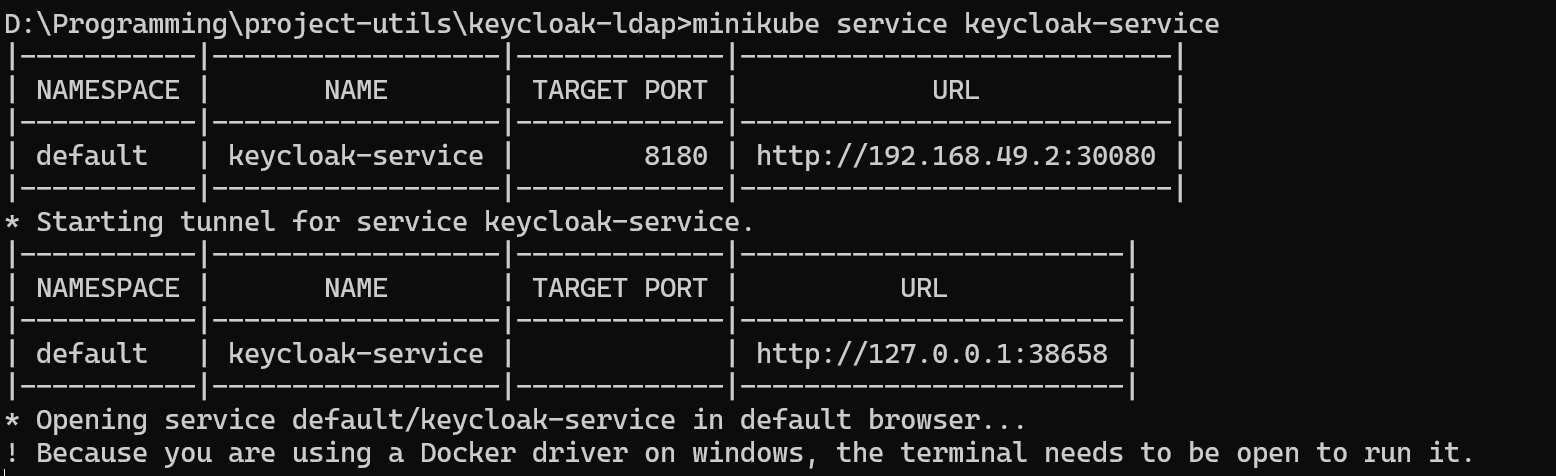
Проверяем что все поды активны



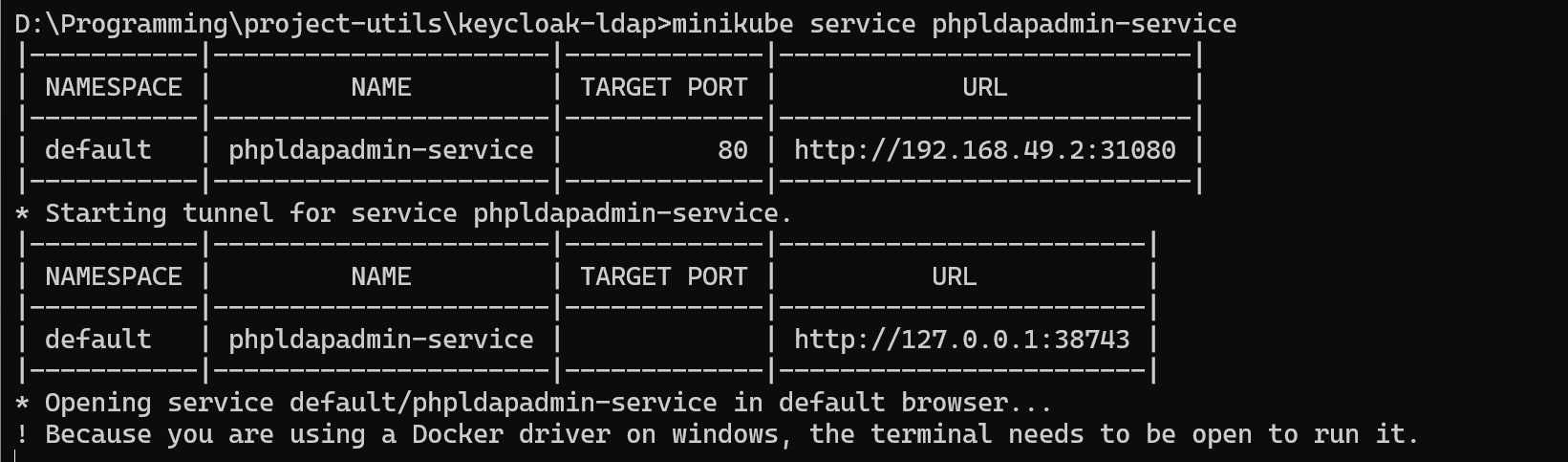
Открываем список сервисов



Запускаем keycloak



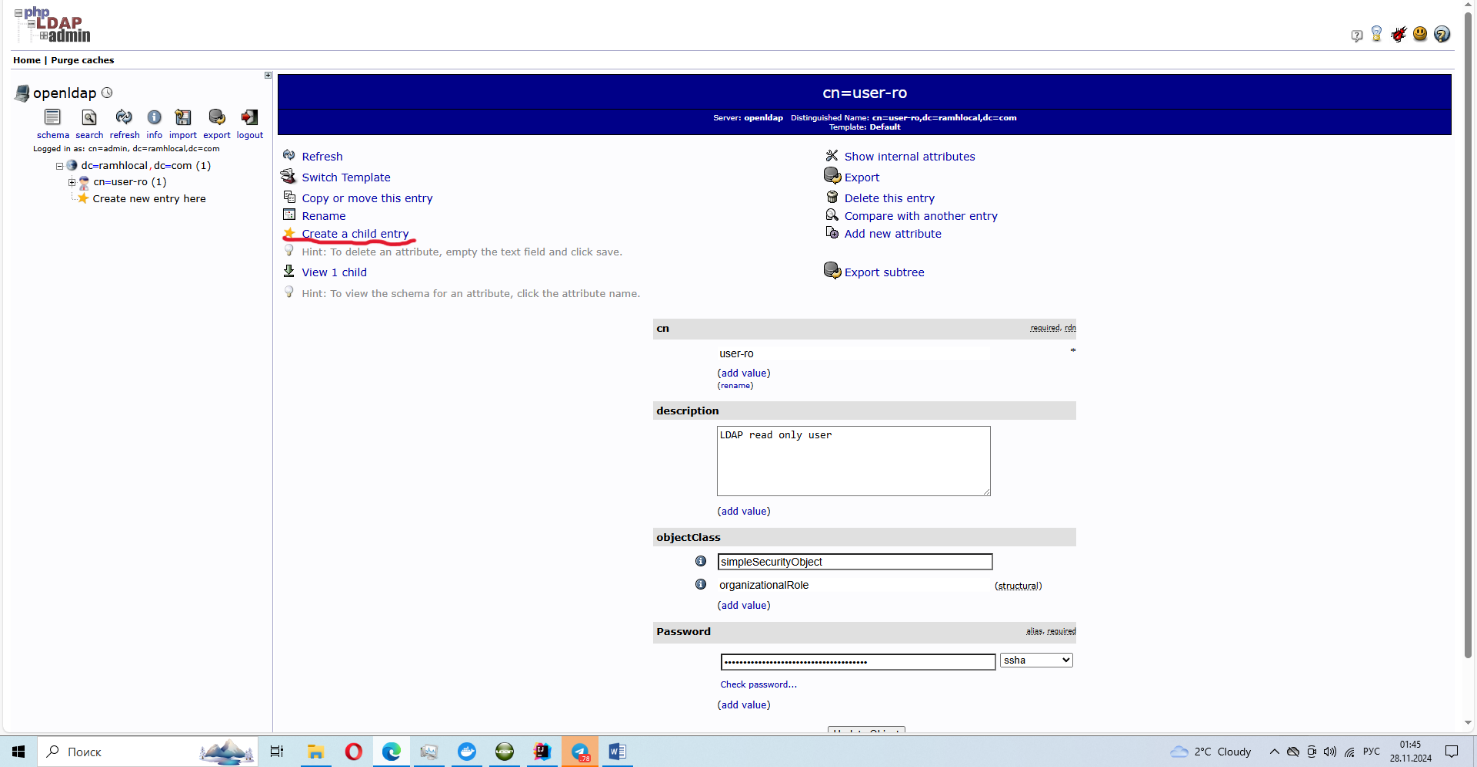
Запускаем phpldapadmin



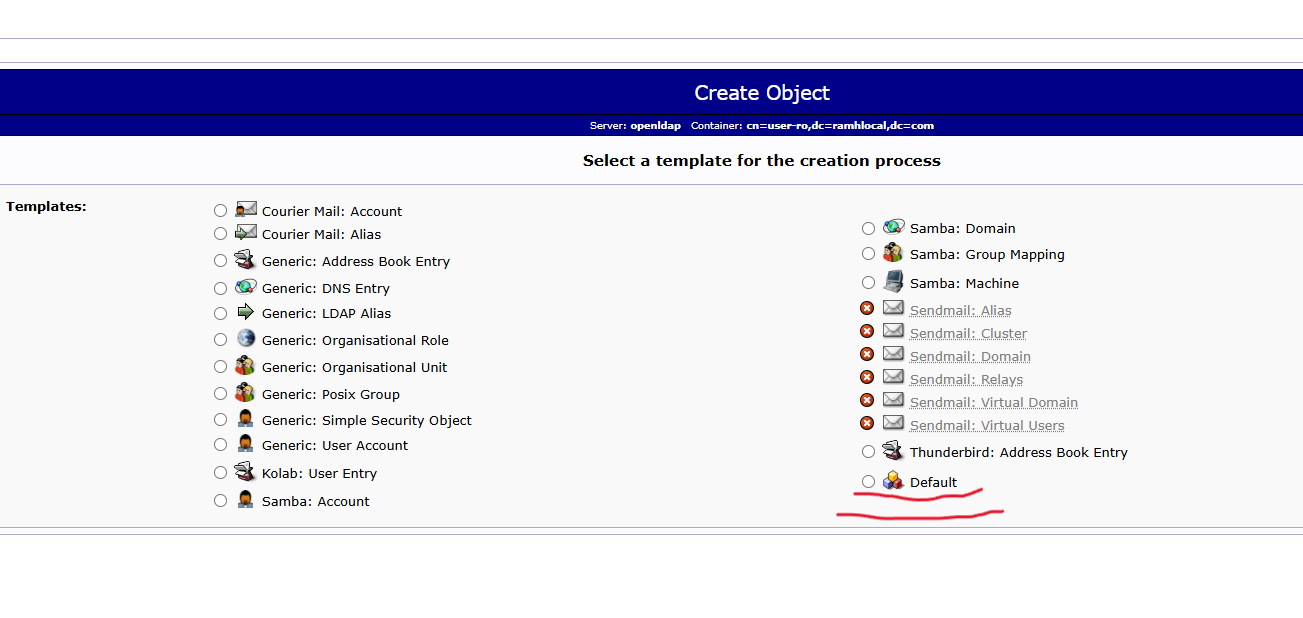
Перейдите на phpLdapAdmin и залогиньтесь у меня это:  
login: cn=admin, dc=ramhlocal,dc=com

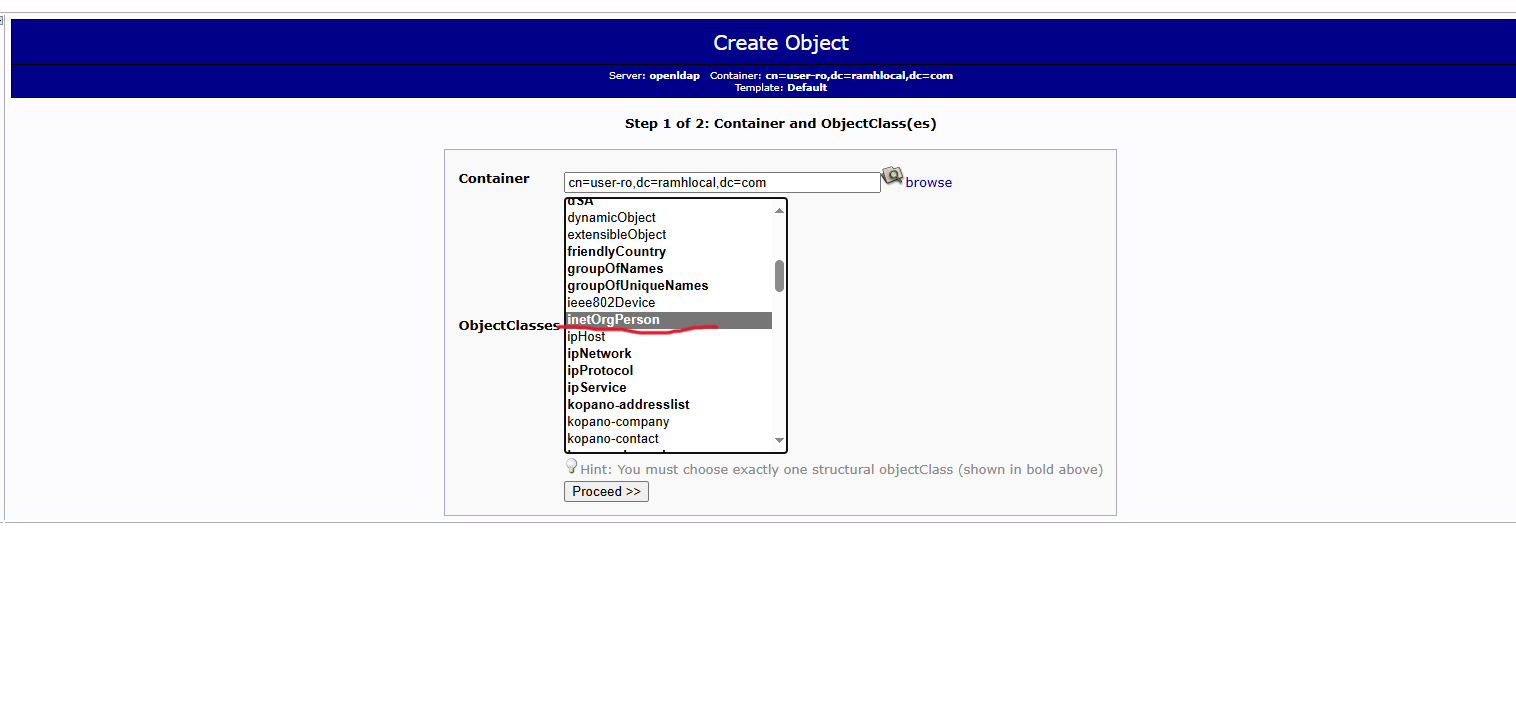
Password: admin\_pass



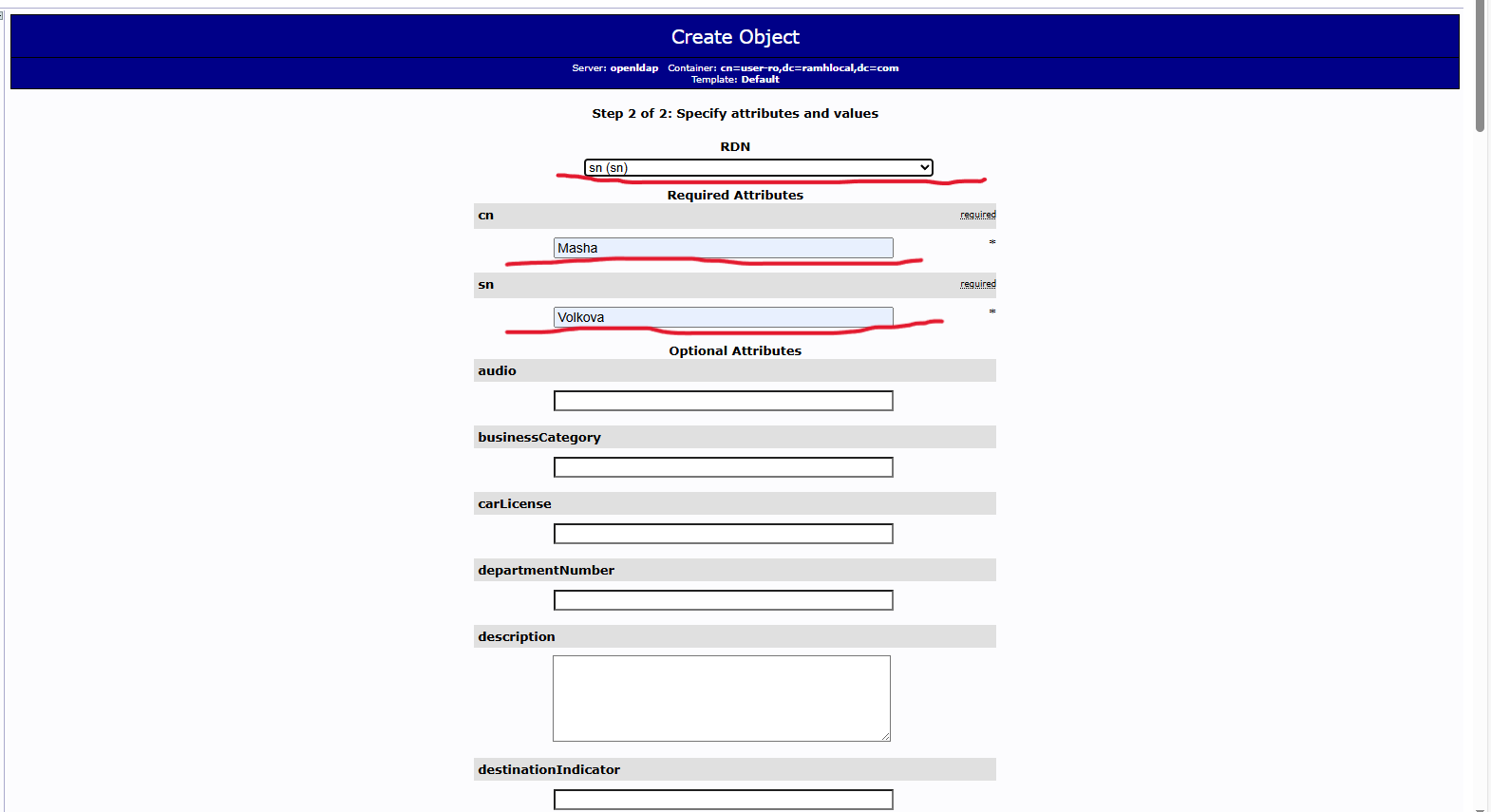
Далее создаем юзера:

Выбираем Default:



Затем обязательно выбираем: 

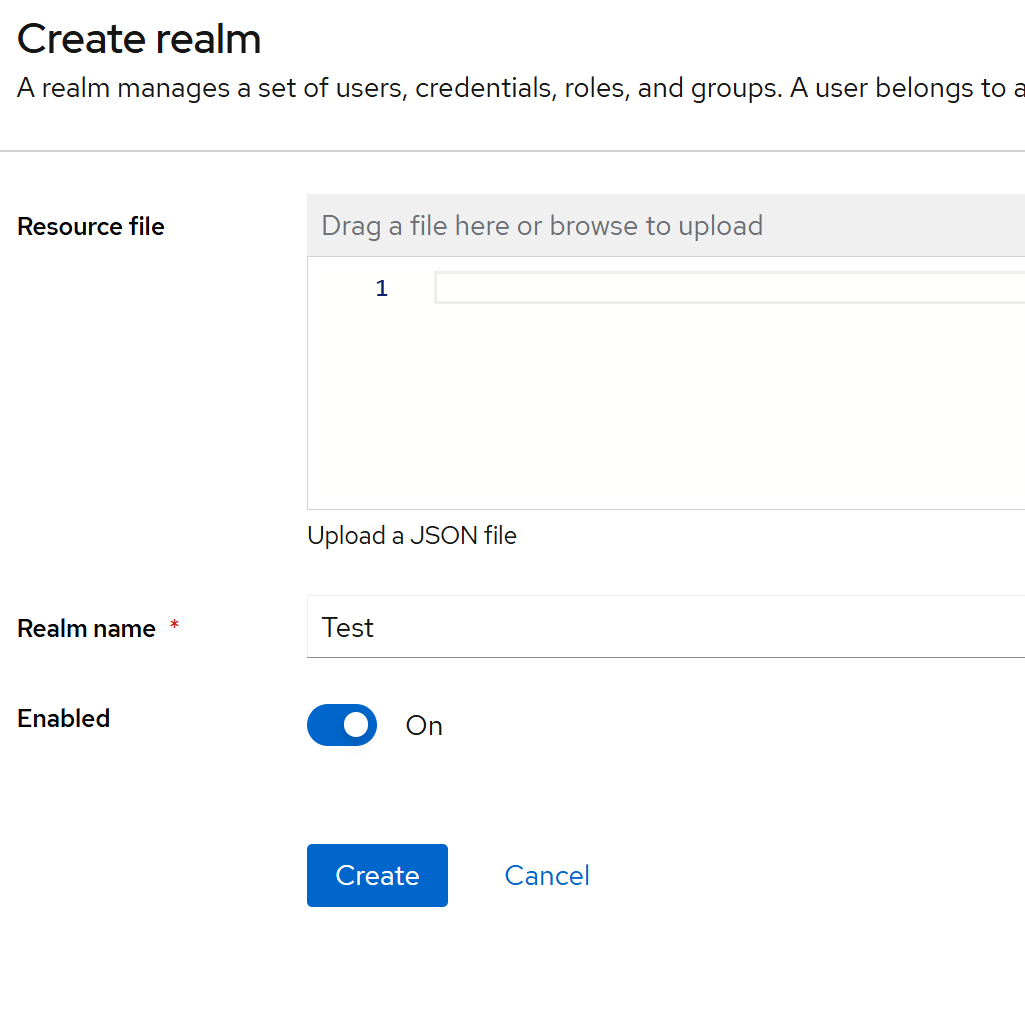
И создаем юзера:



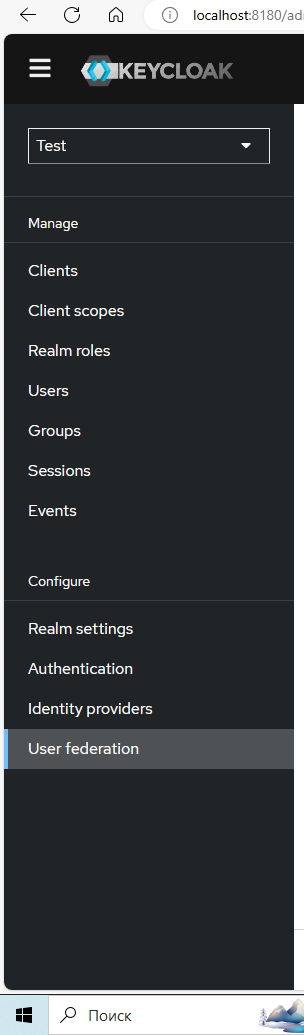
В RDN можно выбрать и cn (по желанию), это лишь отвечает за то, что будет служить юзернеймом

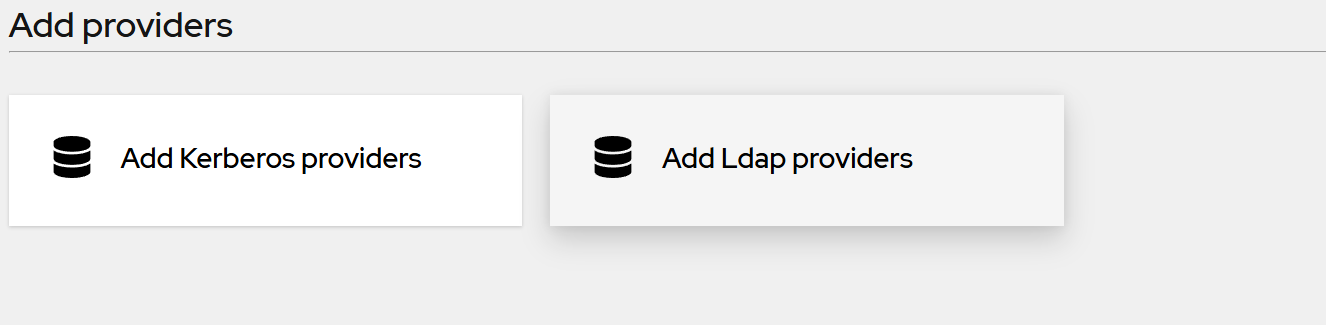
Далее заходим в keycloak под админом и заходим в нужный вам реалм (у меня это Test)

Перейдем в keycloak и Создадим Realm Test

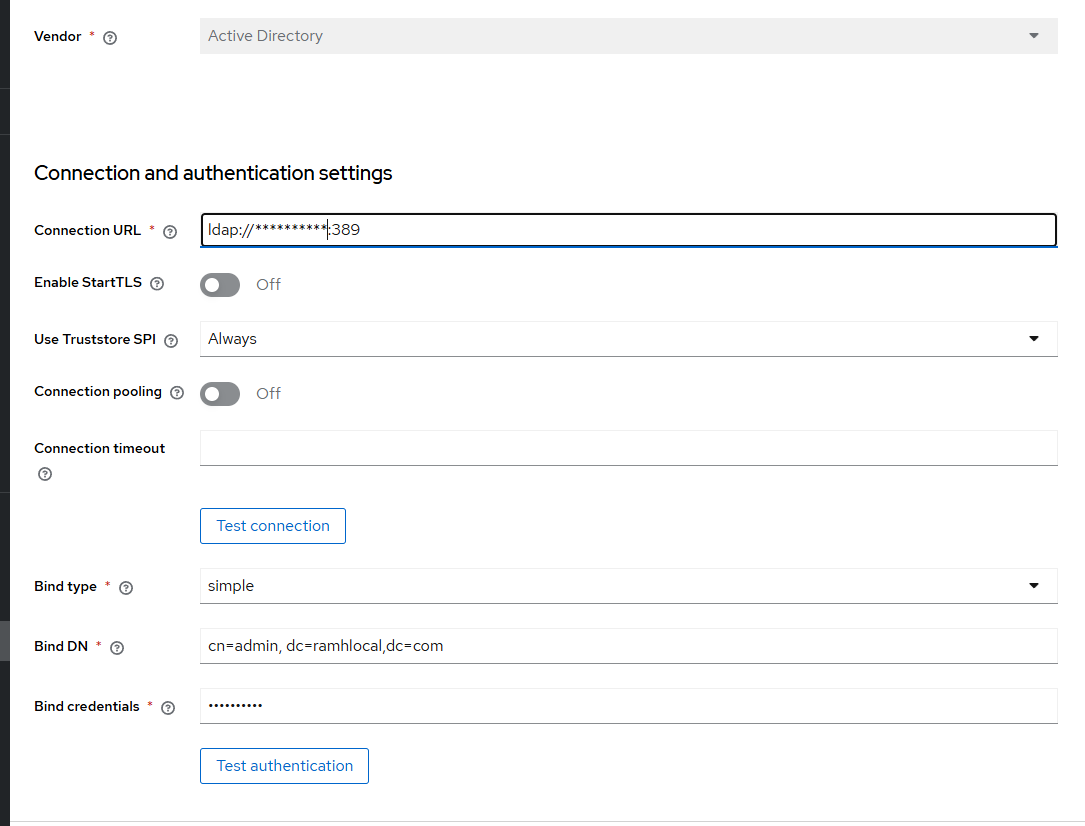


Переходим по User Federation:



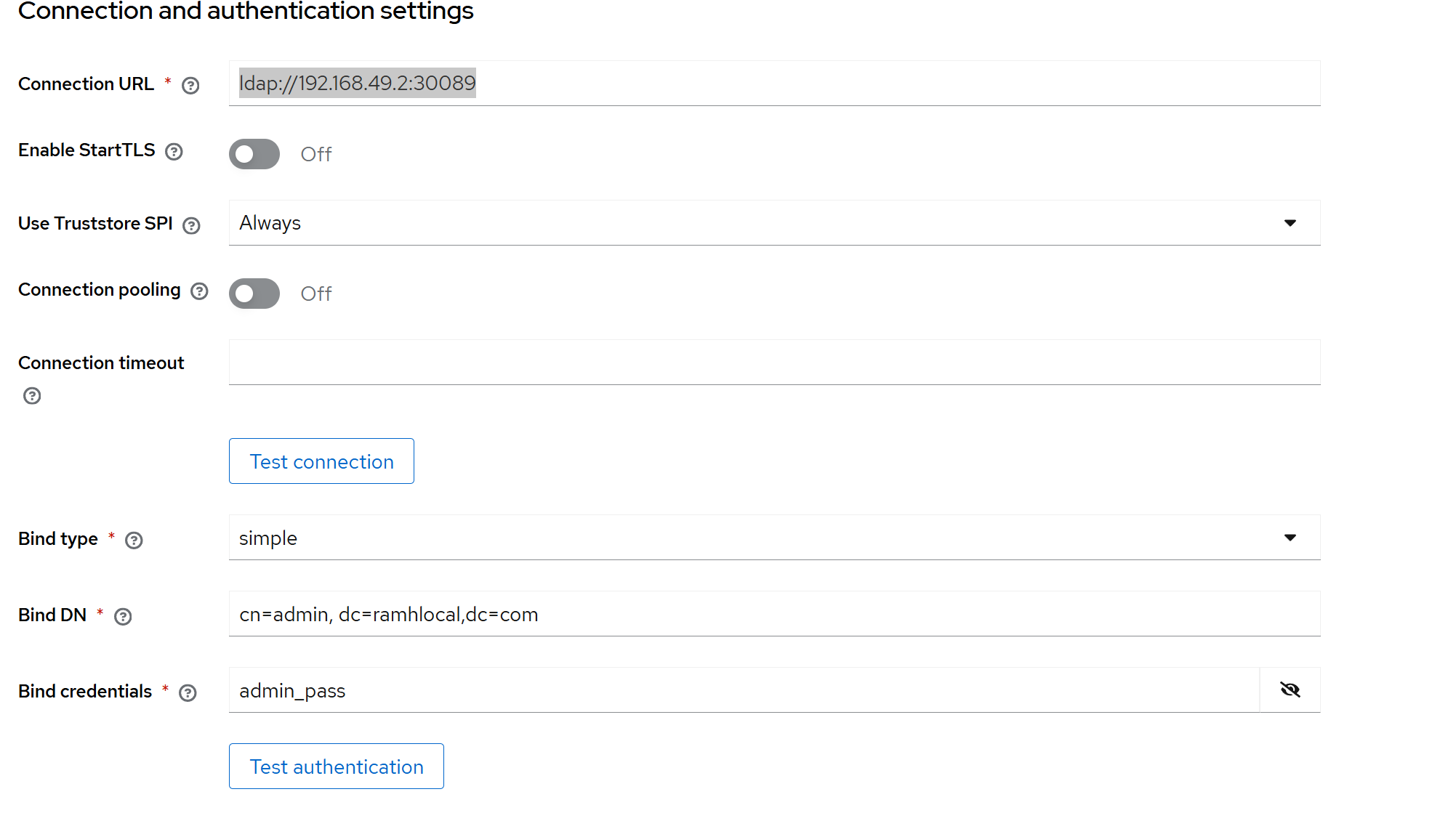


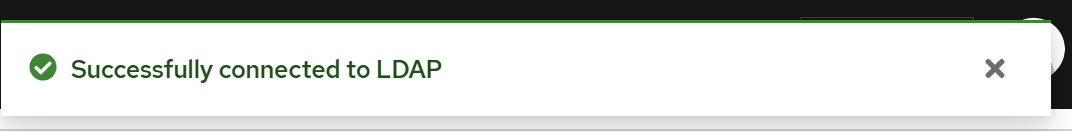
Далее настраиваем наше подключение. ВАЖНО! Там где у меня звездочки вставьте свой айпишник, не localhost а именно айпишник с портом Ldap-а! (ДЛЯ СЕРВИСА МИНИКУБА МЫ БЕРЕМ АЙПИШНИК СЕРВИСА LADAP(НЕ phpldapadmin а ldap) (НЕ ТУНЕЛИРОВАННЫЙ) ИХ МОЖЕТ БЫТЬ НЕСКОЛЬКО ПРОБУЕМ ВСЕ)) НАПРИМЕР ldap:// 192.168.49.2:30089



\\\

Вот так выглядит для сервиса в миникубе

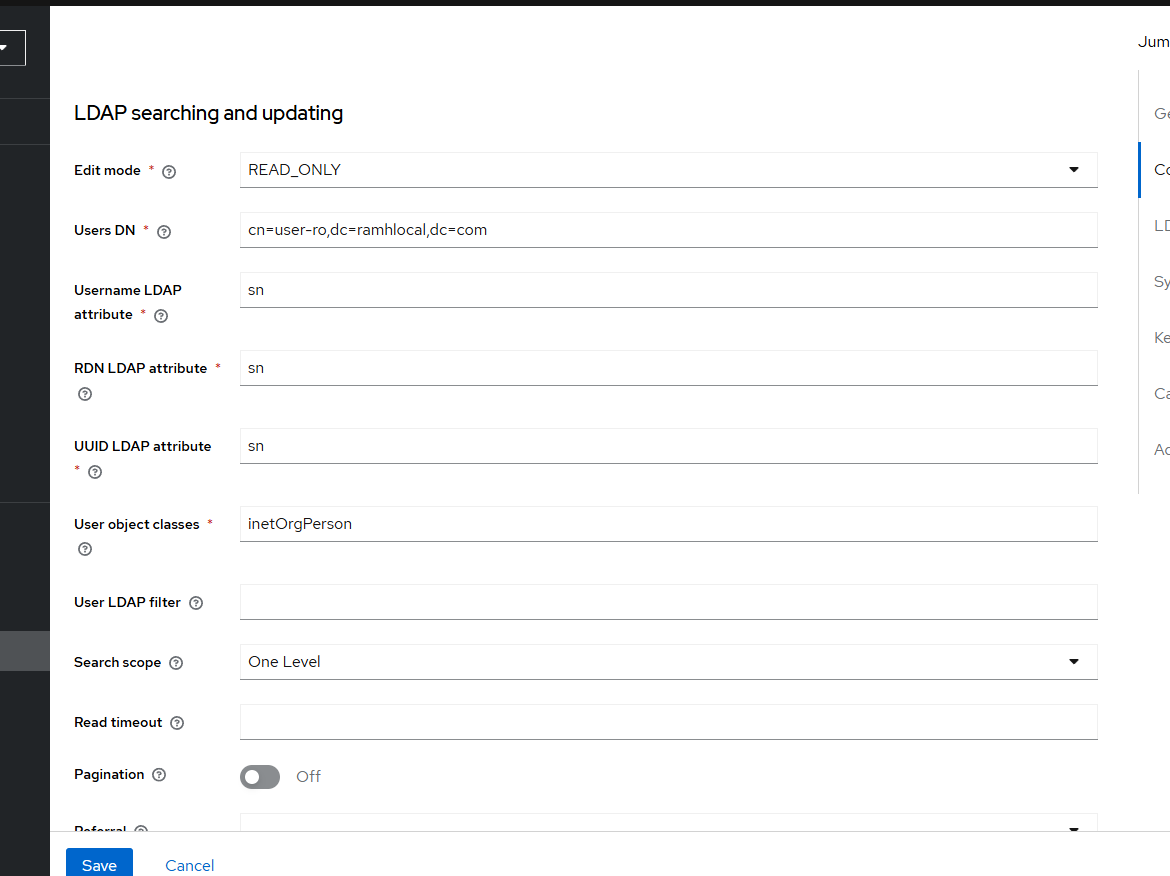




\\\

Bind dn и bind credentials это наш пароль и логин админа. Обязательно проверьте соединение и аутентификацию, проблем быть не должно.

Далее:



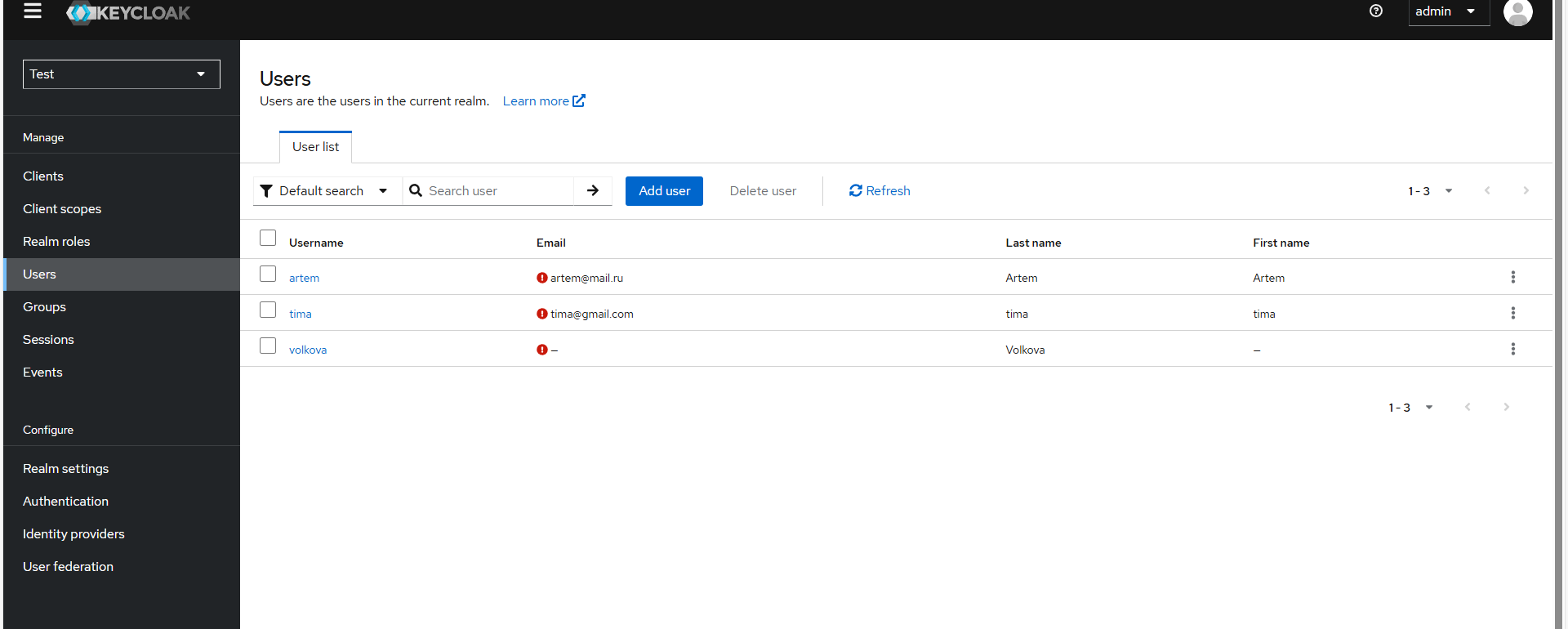
Username LDAP attribute – это атрибут который мы задаем юзернейму (у меня sn)

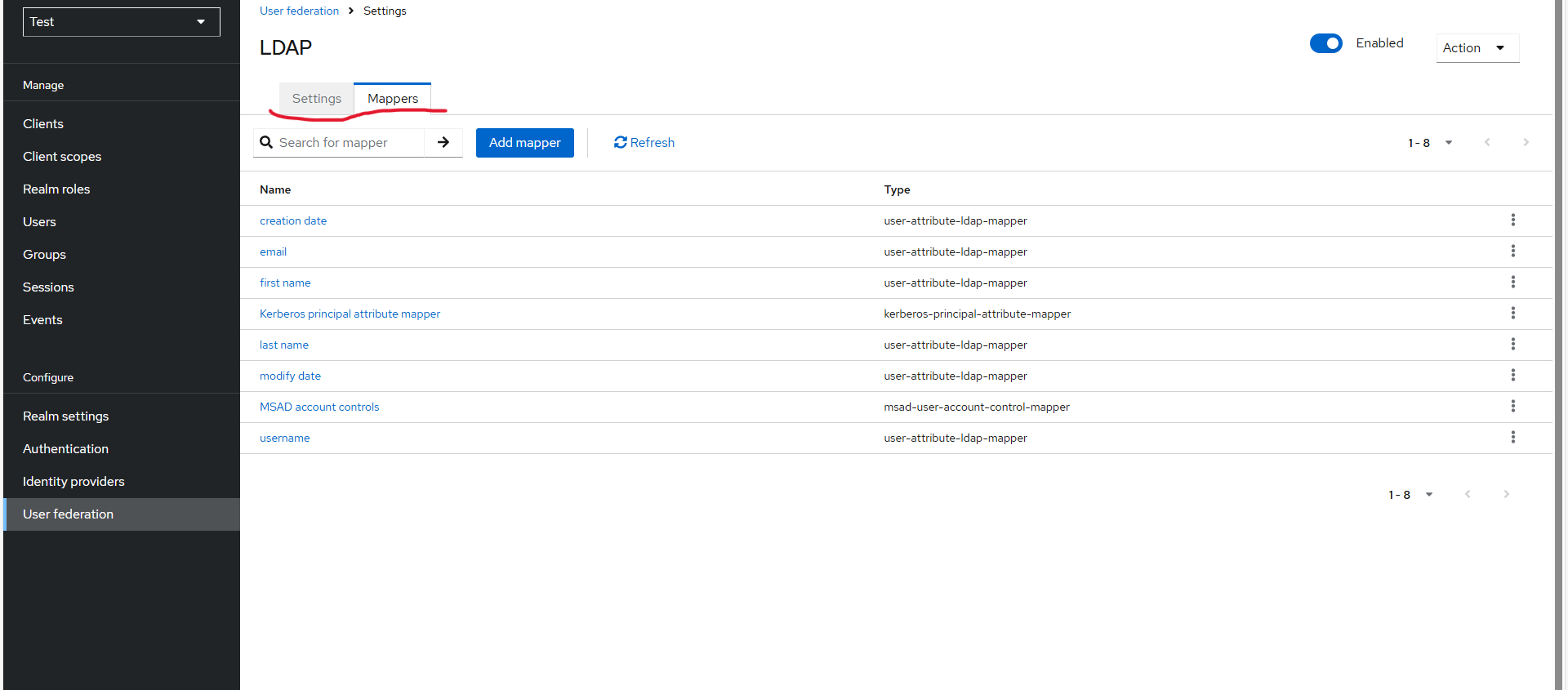
RDN LDAP ATTRIBUTE – На примере моего dn: sn=Volkova,cn=user-ro,dc=ramhlocal,dc=com я должен задать самый левый rdn т.е. sn (DN состоит из Relative Distinguish Name (RDN))

UUID LDAP ATTRIBUTE – уникальный атрибут пользователя в моем случае это опять же sn

User object classes – его мы задавали когда создавали юзера в phpldap (inetOrgPerson)

Далее переходим в Users и вуаля наш юзер импортировался с Ldap!



Так же если вам надо интегрировать еще какие-то данные, например email, то в user federation вам нужно перейти в mappers

Вот я замапил имя:

