Установка и настройка Keycloak

Установка Keycloak

> **Важно**
>
> Если у вас уже есть Keycloak, то пропустите этот шаг и перейдите к его настройке.

Для установки Keycloak можете воспользоваться следующим чартом

Перед установкой необходимо создать следующий файл `values.yaml`, подставив везде свое значение в переменную `IP_LOADBALANCER`. Если у вас есть собственное доменное имя, то замените `IP_LOADBALANCER.nip.io` на свое доменное имя.

Файл `values.yaml`

```yaml
keycloakUser: "admin"
keycloakPassword: "admin"


databaseUser: "keycloak-user"
databasePassword: "dbpassword"

replicas: 1


extraEnv: |
 - name: JAVA_OPTS
   value: >-
     -XX:+UseContainerSupport
     -XX:MaxRAMPercentage=50.0
     -Djava.net.preferIPv4Stack=true
     -Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS
     -Djava.awt.headless=true
     -Dkeycloak.profile.feature.upload_scripts=enabled
 - name: KEYCLOAK_LOGLEVEL
   value: INFO
 - name: PROXY_ADDRESS_FORWARDING
   value: "true"


extraEnvFrom: |
 - secretRef:
     name: '{{ include "keycloak.fullname" . }}-cred'
 - secretRef:
     name: '{{ include "keycloak.fullname" . }}-db'

secrets:
 cred:
   stringData:
     KEYCLOAK_USER: "{{ .Values.keycloakUser }}"
     KEYCLOAK_PASSWORD: "{{ .Values.keycloakPassword }}"
 db:
   stringData:
     DB_USER: '{{ .Values.databaseUser }}'
     DB_PASSWORD: '{{ .Values.databasePassword }}'


# resources:
#   requests:
#     cpu: 500m
#     memory: 1024Mi
#   limits:
#     cpu: 2000m
#     memory: 2048Mi


ingress:
 enabled: true
 ingressClassName: "nginx"
 servicePort: http
 annotations:
   ingress.kubernetes.io/ssl-redirect: "true"
```

```
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/proxy-body-size: "128k"
    nginx.ingress.kubernetes.io/server-snippet: |
      more_set_headers "Access-Control-Allow-Origin: $http_origin";
      location ~* /auth/realms/[^/]+/metrics {
          return 403;
      }
  rules:
    - host: "IP_LOADBALANCER.nip.io"
      paths:
        - path: /auth
          pathType: Prefix
  tls:
    - hosts:
        - "IP_LOADBALANCER.nip.io"
      secretName: dev-wildcard
  console:
    enabled: true
    ingressClassName: "nginx"
    annotations:
      ingress.kubernetes.io/ssl-redirect: "true"
      nginx.ingress.kubernetes.io/ssl-redirect: "true"
      nginx.ingress.kubernetes.io/proxy-body-size: "128k"
    rules:
      - host: "IP_LOADBALANCER.nip.io"
        paths:
          - path: /auth/admin/
            pathType: Prefix
    tls:
      - hosts:
          - "IP_LOADBALANCER.nip.io"
        secretName: dev-wildcard
```

После создания файла, выполните установку Keycloak:

```
helm repo add codecentric https://codecentric.github.io/helm-charts
helm repo update
helm upgrade --install keycloak codecentric/keycloak \
--values values.yaml \
--namespace default \
--kube-context $CONTEXT_NAME
```

Настройка доступа к Keycloak

0. Пройдите по адресу `https://IP_LOADBALANCER.nip.io/auth/admin/master/console/` и авторизуйтесь в системе.
1. Создайте клиента под именем `klmg` и добавить клиенту в поле Valid redirect URIs текущий host в формате `https://{HOST}/*` (для локальной установки - `https://IP_LOADBALANCER.nip.io/*`).
2. В настройках клиента поле `Access Type` должно быть в значении `public`. В более новых версиях Keycloak, аналогом поля `Access Type` являются поля `Client authentication` и `Authorization`, они должны быть в состоянии `off`.

3. Настроить маппинг:
    a. **В дефолтной версии:** В настройках клиента необходимо перейти в `Mappers`, нажать кнопку `Create`.
    b. **В новой версии:**
    В настройках клиента необходимо перейти в `Client scopes`. Из дефолтного списка `Assigned client scope` выбрать `<название клиента>-dedicated` и добавить маппер `by configuration` нажав `Add mapper`.
    Также нажать на `Add client scope` и выбрать `openid` -> `Add` -> `Default`.
    В настройках указать:
        - Name: roles
        - Mapper type: User Client Role
        - Token Claim Name: roles
        - Остальные настройки оставить по умолчанию, нажать `Save`.
4. В настройках клиента необходимо перейти в `Roles` и создать роль `predicate_admin` для предоставления выбранным пользователям прав админа (возможность видеть и изменять все сущности созданные на стенде любым пользователем) и `predicate_metric` (доступ до базовых метрик).
5. Создать необходимых пользователей и выдать им при необходимости роль `predicate_admin`. Если у пользователя нет роли `predicate_admin`, то он ему нужно будет выдать роль `predicate_metric`, чтобы он мог видеть базовые метрики. Для этого перейдите в раздел `Users`, выберите пользователя, перейдите в раздел `Role Mappings`.
    a. **В дефолтной версии:** В поле `Client Roles` выберите клиента `klmg`, в поле `Available Roles выберите роль` выберите `predicate_admin` и нажмите `Add selected`.
    b. **В новой версии:** Нажмите `Assign role` -> `Fileter by client`. В поиске введите `predicate_admin`, выберите роль и нажмите `Assign`.