Keycloak Installation and Configuration

Keycloak Installation

> **Important**
>
> If you already have Keycloak, skip this step and proceed to its configuration.

To install Keycloak, you can use the following chart.

Before installation, you need to create the following `values.yaml` file, replacing the variable `IP_LOADBALANCER` with your value everywhere. If you have your own domain name, replace `IP_LOADBALANCER.nip.io` with your domain name.

File `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
```

After creating the file, execute the Keycloak installation:

```
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
```

Configuring Access to Keycloak

0. Go to `https://IP_LOADBALANCER.nip.io/auth/admin/master/console/` and log in to the system.
1. Create a client named `klmg` and add the current host in the Valid redirect URIs field in the format `https://{HOST}/*` (for local installation - `https://IP_LOADBALANCER.nip.io/*`).
2. In the client settings, the `Access Type` field should be set to `public`. In newer versions of Keycloak, the equivalent of the `Access Type` field are the `Client authentication` and `Authorization` fields, which should be set to `off`.

3. Configure the mapping:
    a. **In the default version:** In the client settings, go to `Mappers`, click the `Create` button.
    b. **In the new version:**
    In the client settings, go to `Client scopes`. From the default list of `Assigned client scope`, select `<client name>-dedicated` and add the `by configuration` mapper by clicking `Add mapper`.
    Also click on `Add client scope` and select `openid` -> `Add` -> `Default`.
    In the settings specify:
        - Name: roles
        - Mapper type: User Client Role
        - Token Claim Name: roles
        - Leave other settings as default, click `Save`.
4. In the client settings, go to `Roles` and create the role `predicate_admin` to grant selected users admin rights (the ability to see and modify all entities created on the stand by any user) and `predicate_metric` (access to basic metrics).
5. Create the necessary users and assign them the `predicate_admin` role if needed. If a user does not have the `predicate_admin` role, they will need to be assigned the `predicate_metric` role so they can see basic metrics. To do this, go to the `Users` section, select the user, and go to the `Role Mappings` section.
    a. **In the default version:** In the `Client Roles` field, select the client `klmg`, in the `Available Roles` field select the role `predicate_admin` and click `Add selected`.
    b. **In the new version:** Click `Assign role` -> `Filter by client`. In the search, enter `predicate_admin`, select the role and click `Assign`.