Logowanie do Camundy 8: Management Identity, SSO z Entra ID i droga bez Microsoftu

◫◫ Poziom: zaawansowane

◫◫ Dotyczy: Camunda 8.10 (wykres 15.4.0): Orchestration Cluster, Connectors, Management Identity, Camunda Hub; Microsoft Entra ID (dzierżawa zielona.onmicrosoft.com), Keycloak 26, Authentik; Azure CLI

▯ Czas: ~120 min

Przełączamy Camundę na `bpm01` z logowania Basic na logowanie jednym kontem firmowym przez OIDC i dokładamy dwie aplikacje: Management Identity i Camunda Hub z Web Modelerem. Główna droga to Microsoft Entra ID, bo Zielona ma konta w chmurze Microsoftu; obok opisujemy drogę bez Microsoftu przez Keycloak albo Authentik. Wracamy tu przy nowym pracowniku w zespole procesów, przy wygasającym sekrecie i przy zmianie dostawcy tożsamości.

◫◫ ZAŁOŻENIA SCENARIUSZA

konta pracowników synchronizuje do Entra ID `sync01` według strony Entra Connect i Cloud Sync: synchronizacja ad.zielona.example z Entra ID, a `bpm01` ma wyjście do internetu na `login.microsoftonline.com`. Optimize nie jest zainstalowany, więc nie zakładamy dla niego rejestracji. W wariancie bez Microsoftu przyjmujemy Keycloak w klastrze na `bpm01` z kontami z FreeIPA (`ipa01`) albo Authentik jako zewnętrzny dostawca.

◫◫ Skróty użyte na tej stronie (15)

Skrót	Co znaczy
AD	Active Directory: katalog Microsoftu z kontami użytkowników, komputerów i grup w firmie
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą

Skrót	Co znaczy
BPM	Business Process Management: zarządzanie procesami biznesowymi, czyli opisywanie, wykonywanie i mierzenie obiegu spraw w firmie; też potoczna nazwa silnika procesów
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
etcd	etc distributed: rozproszony magazyn par klucz-wartość, w którym Kubernetes trzyma cały stan klastra, a zmiany uzgadnia protokołem Raft
GLPI	Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01
GUID	Globally Unique Identifier: unikalny w skali globalnej identyfikator obiektu, tu konkretnego protektora klucza BitLockera
HTTP	Hypertext Transfer Protocol: protokół stron WWW
HTTPS	HTTP zabezpieczony szyfrowaniem TLS
LDAP	Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)
MFA	Multi-Factor Authentication: logowanie drugim składnikiem, np. kodem z telefonu
OIDC	OpenID Connect: protokół logowania przez zewnętrznego dostawcę tożsamości, na którym opiera się funkcja SSO w Vaultwardenie
REST	sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu
SPA	Single-Page Application: aplikacja WWW działająca w całości w przeglądarce na jednej stronie; w Entra ID osobny typ platformy rejestracji bez sekretu klienta
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

Fiszka

Komendy `az` wykonujemy w Azure Cloud Shell albo na stacji administratora po `az login` kontem z rolą Application Administrator (tylko tyle, ile trzeba do rejestracji aplikacji). Komendy `kubectl` i `helm` jak na stronie o instalacji. Rejestracje zakładamy w portalu, bo jest ich kilka i każda ma inne ustawienia; komendy służą do odczytu i rutyny.

TYLKO ODCZYT

Odczyt identyfikatora dzierżawy: az account show

```
az account show --query tenantId -o tsv
```

Identyfikator dzierżawy (GUID) wchodzi do wszystkich adresów OIDC w `values.yaml`.

□□ **GUID** = Globally Unique Identifier: unikalny w skali globalnej identyfikator obiektu, tu konkretnego protektora klucza BitLockera · **OIDC** = OpenID Connect: protokół logowania przez zewnętrznego dostawcę tożsamości, na którym opiera się funkcja SSO w Vaultwardenie · **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Odczyt konfiguracji OIDC dzierżawy: curl .well-known/openid-configuration

```
curl -s https://login.microsoftonline.com/IdentyfikatorDzierzawy/v2.0/.well-known/openid-configuration | jq -r '.issuer, .authorization_endpoint, .token_endpoint, .jwks_uri'
```

Zwraca dokładnie te cztery adresy, które wpisujemy w `global.identity.auth`. Przepisujemy je stąd, a nie z pamięci.

Przykład:

```
https://login.microsoftonline.com/7d3e1c90-5b2a-4f61-9a0e-2c8b4d6f1e73/v2.0
https://login.microsoftonline.com/7d3e1c90-5b2a-4f61-9a0e-2c8b4d6f1e73/oauth2/v2.0/authorize
https://login.microsoftonline.com/7d3e1c90-5b2a-4f61-9a0e-2c8b4d6f1e73/oauth2/v2.0/token
https://login.microsoftonline.com/7d3e1c90-5b2a-4f61-9a0e-2c8b4d6f1e73/discovery/v2.0/keys
```

□□ **OIDC** = OpenID Connect: protokół logowania przez zewnętrznego dostawcę tożsamości, na którym opiera się funkcja SSO w Vaultwardenie · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

□□ TYLKO ODCZYT

Lista rejestracji aplikacji Camundy: az ad app list

```
az ad app list --display-name camunda --query "[].{nazwa:displayName, appId:appId, obiekt:id}" -o table
```

Pokazuje identyfikatory aplikacji (appId, czyli identyfikator klienta i zarazem odbiorca tokenu) i identyfikatory obiektów potrzebne do zmian przez Graph.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

◫◫ ZMIENIA SYSTEM

Ustawienie tokenów w wersji 2: az rest PATCH applications

```
az rest --method PATCH --uri
https://graph.microsoft.com/v1.0/applications/IdObiektuAplikacji --headers Content-
Type=application/json --body '{"api":{"requestedAccessTokenVersion":2}}'
```

To samo co `"requestedAccessTokenVersion": 2` w manifeście aplikacji w portalu. Bez tego Entra wydaje tokeny w wersji 1.0, a Camunda odrzuca je przy sprawdzaniu wystawcy.

◫◫ **az** = Azure CLI: narzędzie chmury Microsoftu

◫◫ ZMIENIA SYSTEM

Dodanie nowego sekretu klienta: az ad app credential reset --append

```
az ad app credential reset --id IdAplikacji --append --display-name bpm01 --years 1
```

Dopisuje nowy sekret obok starego (bez `--append` stare zostałyby usunięte) i wypisuje go raz. Wartość od razu przenosimy do Vaultwardena i do sekretu Kubernetesa.

◫◫ **az** = Azure CLI: narzędzie chmury Microsoftu

◫◫ TYLKO ODCZYT

Daty wygaśnięcia sekretów: az ad app credential list

```
az ad app credential list --id IdAplikacji --query "[].{opis:displayName,
koniec:endDateTime}" -o table
```

Przykład:

```
Opis    Koniec
------  --------------------
bpm01   2027-11-04T09:12:40Z
```

◫◫ **az** = Azure CLI: narzędzie chmury Microsoftu

◫◫ ZMIENIA SYSTEM

Sekret Kubernetesa z sekretami klientów: kubectl create secret generic

```
kubectl -n camunda create secret generic entra-credentials --from-literal=identity-client-secret='SekretIdentity' --from-literal=orchestration-cluster-client-secret='SekretOrchestration' --from-literal=hub-api-client-secret='SekretHubApi'
```

Nazwy kluczy są takie jak w przykładzie z dokumentacji Camundy dla Entra, więc pliki wartości z dokumentacji i z tej strony do siebie pasują.

kubectl = Kubernetes control: główne narzędzie Kubernetesa

Z życia wzięte

Z ŻYCIA WZIĘTE

Piątek, 15:40. Marek pisze, że od rana nie może wejść do Tasklist: Entra pyta o hasło, potem Camunda pokazuje „403”. Sprawdzamy w Management Identity i w Admin: Marek nie ma żadnej roli, bo reguły mapowania dają dostęp grupie `GG-BPM-Magazyn`, a Marek przeszedł z innego działu i nikt nie dodał go do grupy w AD. Po dodaniu go do grupy i synchronizacji przez `sync01` wylogowuje się, loguje ponownie i widzi swoje zadania kompletacji, a my dopisujemy krok „grupy BPM” do listy przyjęcia pracownika w GLPI.

Jak to działa

Rozwiń: jak to działa pod spodem

Trzy miejsca, w których mieszka „kto jest kim”

Pojęcie: OIDC (OpenID Connect). Sposób logowania, w którym aplikacja nie zna hasła użytkownika: odsyła przeglądarkę do dostawcy tożsamości (Entra ID, Keycloak), a ten po zalogowaniu oddaje podpisany token z informacjami o osobie. Aplikacja sprawdza podpis

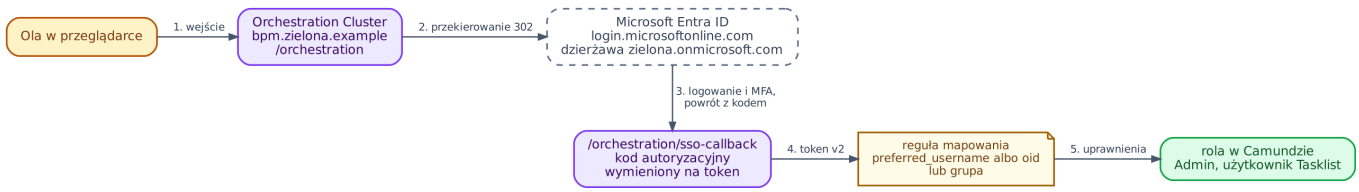
i na tej podstawie wpuszcza.

Pojęcie: Management Identity. Osobna aplikacja Camundy, która łączy się z dostawcą OIDC i zarządza rolami dla Camunda Hub i Optimize. Orchestration Cluster ma własne zarządzanie użytkownikami i rolami w aplikacji Admin (profil `admin` w wykresie), więc reguły dostępu ustawiamy w dwóch miejscach.

Pojęcie: reguła mapowania (mapping rule). Zasada „kto ma w tokenie takie roszczenie (claim) o takiej wartości, dostaje taką rolę”. Na przykład `preferred_username = ola.wisniewska@zielona.example` daje rolę administratora, a grupa z Entra daje rolę użytkownika Tasklist.

Miejsce	Co ustala	Gdzie w Camundzie
Entra ID (albo Keycloak)	kto może się zalogować, MFA, grupy, ważność sekretów	portal Entra, `az`
Management Identity	role w Camunda Hub (Web Modeler) i Optimize	`https://bpm.zielona.example/identity`
Admin w Orchestration Cluster	role w Operate, Tasklist i API	`https://bpm.zielona.example/orchestration/admin` (ścieżkę sprawdzamy w interfejsie)

Droga logowania



Rysunek 1. Logowanie przez Entra ID: Ola wchodzi na /orchestration, Camunda przekierowuje do login.microsoftonline.com w dzierżawie zielona.onmicrosoft.com, po logowaniu i MFA przeglądarka wraca na /orchestration/sso-callback z kodem wymienianym na token v2, reguła mapowania czyta preferred_username albo oid albo grupę i nadaje rolę (kliknięcie otwiera pełny rozmiar)

Po powrocie z Entra aplikacja wymienia jednorazowy kod na tokeny, sprawdza podpis kluczami z adresu `jwks_uri` i wystawcę z `issuer`. Dlatego wersja tokenu musi się zgadzać: token 1.0 ma innego wystawcę niż adres `.../v2.0`, który wpisaliśmy.

Pięć rejestracji według dokumentacji, cztery u nas

Dokumentacja Camundy 8.10 dla Entra zakłada osobną rejestrację dla każdej aplikacji, bo identyfikator klienta jest zarazem odbiorcą tokenu (audience). Connectors korzystają z rejestracji Orchestration Cluster.

Rejestracja	Platforma	Adres przekierowania	Sekret
`camunda-bpm01-identity`	Web	`https://bpm.zielona.example/identity/auth/login-callback`	tak
`camunda-bpm01-orchestration`	Web	`https://bpm.zielona.example/orchestration/sso-callback`	tak (też dla Connectors)
`camunda-bpm01-hub-api`	Web	brak	tak (dostęp do API Web Modelera)
`camunda-bpm01-hub-ui`	aplikacja jednostronicowa (SPA)	`https://bpm.zielona.example/modeler/login-callback`	nie
Optimize	Web	`.../api/authentication/callback`	pomijamy, Optimize nie jest zainstalowany

Każda rejestracja dostaje: tokeny w wersji 2, opcjonalne roszczenie `preferred_username` w tokenie dostępu i w tokenie ID (Management Identity czyta je z tokenu dostępu, bez niego użytkownik dostaje 403) oraz zgodę administratora na zakresy `openid`, `profile`, `email`, `offline_access` i `IdAplikacji/.default`. Zakres `offline_access` daje odświeżanie tokenu; bez niego użytkownik loguje się ponownie po wygaśnięciu tokenu.

Pierwszy administrator: preferred_username czy oid

Management Identity zakłada pierwszą regułę z `initialClaimName` i `initialClaimValue` tylko przy pierwszym starcie. Później zmiana tych wartości w Helm nic nie daje, trzeba poprawiać bazę `identity` ręcznie. Dokumentacja pokazuje przykład z `preferred_username` (adres e-mail), ale do produkcji zaleca `oid`, czyli identyfikator obiektu w Entra, bo nie zmienia się przy zmianie nazwiska albo adresu.

Wybieramy `oid` konta administratora. Identyfikator odczytujemy poleceniem `az ad user show --id ola.wisniewska@zielona.example --query id -o tsv`.

Droga bez Microsoftu

Nie każda firma ma Entra ID, a Zielona ma też FreeIPA dla serwerów linuksowych. Camunda 8.10 obsługuje dwa tryby: zewnętrzny Keycloak (`global.identity.auth.type: KEYCLOAK`, domyślny) i zewnętrzny dostawca OIDC (`MICROSOFT` albo `GENERIC`). Od 8.10 wykres nie instaluje Keycloaka sam, a dokumentacja Camundy poleca operator Keycloaka z obrazem Keycloaka przygotowanym przez Camundę.

Droga	Kto zarządza kontami	Co stawiamy	Typ w `values.yaml`	Uwagi
Microsoft Entra ID	Entra (synchronizacja z AD przez `sync01`)	nic poza rejestracjami	`MICROSOFT`	MFA i dostęp warunkowy z Entra; zależność od internetu
Keycloak w klastrze	Keycloak, konta federowane z FreeIPA (`ipa01`) albo z AD przez LDAP	operator Keycloaka, baza `keycloak` w PostgreSQL	`KEYCLOAK`	obsługiwana seria 26.x; działa bez internetu; kolejna aplikacja do aktualizowania
Authentik	Authentik, konta z LDAP albo własne	Authentik obok, na przykład na `docker01`	`GENERIC`	ten sam tryb co dowolny dostawca OIDC; wszystkie adresy podajemy ręcznie
Google Workspace	Google	nic poza klientem OAuth	`GENERIC` (do sprawdzenia)	zgodności z Camundą nie sprawdzaliśmy

Fragmenty `values.yaml` dla obu dróg bez Microsoftu są w kroku 7.

Przebieg

Zanim zaczniemy

1. Camunda działa według strony Camunda 8 Self-Managed przez Helm: instalacja próbna na bpm01, a jej `values.yaml` leży w `/k8s/helm/camunda/`.

2. Bazy `identity` i `hub` z rolami i sekretami `pg-bpm-identity` i `pg-bpm-hub` istnieją w klastrze `pg-bpm` (PostgreSQL w Kubernetesie: operator CloudNativePG, kopie, odtworzenie i aktualizacja).
3. W AD są grupy `GG-BPM-Admini`, `GG-BPM-Modelarze` i `GG-BPM-Magazyn`, synchronizowane do Entra (nazwy do decyzji autora).
4. Kopie przed zmianą: `Backup` dla `pg-bpm`, kopia Zeebe, migawka etcd.

1. Rejestracje w Entra ID

Dla każdego wiersza z tabeli „Pięć rejestracji według dokumentacji, cztery u nas” zakładamy rejestrację w portalu:

```
Microsoft Entra admin center → Entra ID → App registrations → New registration → Name: camunda-bpm01-orchestration → Supported account types: Accounts in this organizational directory only → Redirect URI: Web, https://bpm.zielona.example/orchestration/sso-callback → Register
```

```
App registration → Token configuration → Add optional claim → Access: preferred_username, ID: preferred_username → Add
```

```
App registration → API permissions → Add a permission → Microsoft Graph → Delegated: openid, profile, email, offline_access → Grant admin consent for Zielona
```

Potem tokeny w wersji 2 (karta „Ustawienie tokenów w wersji 2” albo manifest w portalu) i sekret dla rejestracji typu Web (karta „Dodanie nowego sekretu klienta”). Rejestracja `camunda-bpm01-hub-ui` dostaje platformę „Single-page application” z adresem `.../modeler/login-callback` i nie dostaje sekretu.

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran App registrations → camunda-bpm01-orchestration → Authentication, widoczna platforma Web z adresem `https://bpm.zielona.example/orchestration/sso-callback`.

2. Sekrety w klastrze

Sekret `entra-credentials` zakładamy kartą z Fiszki, z trzema sekretami klientów. Ich wartości zapisujemy w Vaultwardenie w folderze „BPM”, z datą wygaśnięcia w opisie.

3. Plik wartości dla OIDC

Zmiany trzymamy w osobnym pliku `/k8s/helm/camunda/values-entra.yaml`, który podajemy po `values.yaml`, więc jego klucze wygrywają:

```
# Logowanie przez Microsoft Entra ID, Management Identity i Camunda Hub na bpm01.
global:
  security:
    authentication:
      method: oidc
  identity:
    auth:
      enabled: true
      type: MICROSOFT
      issuer: https://login.microsoftonline.com/IdentyfikatorDzierzawy/v2.0
      issuerBackendUrl: https://login.microsoftonline.com/IdentyfikatorDzierzawy/v2.0
      tokenUrl: https://login.microsoftonline.com/IdentyfikatorDzierzawy/oauth2/v2.0/token
      authUrl: https://login.microsoftonline.com/IdentyfikatorDzierzawy/oauth2/v2.0/authorize
      jwksUrl: https://login.microsoftonline.com/IdentyfikatorDzierzawy/discovery/v2.0/keys
      identity:
        clientId: IdAplikacjiIdentity
        audience: IdAplikacjiIdentity
        secret:
          existingSecret: entra-credentials
          existingSecretKey: identity-client-secret
        initialClaimName: oid
        initialClaimValue: IdObiektuAdministratora
      camundaHub:
        clientId: IdAplikacjiHubUi
        clientApiAudience: IdAplikacjiHubUi
        publicApiAudience: IdAplikacjiHubApi
        redirectUrl: https://bpm.zielona.example/modeler

identity:
  enabled: true
  contextPath: /identity
  fullURL: https://bpm.zielona.example/identity
  externalDatabase:
    enabled: true
```

```
    host: pg-bpm-rw.camunda.svc
    port: 5432
    database: identity
    username: identity
    secret:
      existingSecret: pg-bpm-identity
      existingSecretKey: password

orchestration:
  security:
    authentication:
      oidc:
        clientId: IdAplikacjiOrchestration
        audience: IdAplikacjiOrchestration
        usernameClaim: preferred_username
        clientIdClaim: azp
        preferUsernameClaim: true
        redirectUrl: https://bpm.zielona.example/orchestration
        scope:
          - openid
          - profile
          - offline_access
          - IdAplikacjiOrchestration/.default
        secret:
          existingSecret: entra-credentials
          existingSecretKey: orchestration-cluster-client-secret
    initialization:
      defaultRoles:
        admin:
          users:
            - ola.wisniewska@zielona.example
  env:
    - name: CAMUNDA_SECURITY_AUTHENTICATION_OIDC_USER_INFO_ENABLED
      value: "false"
    - name: SERVER_MAX_HTTP_REQUEST_HEADER_SIZE
      value: "65536"

connectors:
  security:
    authentication:
```

```
      oidc:
        clientId: IdAplikacjiOrchestration
        audience: IdAplikacjiOrchestration
        tokenScope: IdAplikacjiOrchestration/.default
        secret:
          existingSecret: entra-credentials
          existingSecretKey: orchestration-cluster-client-secret

camundaHub:
  enabled: true
  contextPath: /modeler
  restapi:
    externalDatabase:
      host: pg-bpm-rw.camunda.svc
      port: 5432
      database: hub
      username: hub
      secret:
        existingSecret: pg-bpm-hub
        existingSecretKey: password
    mail:
      fromAddress: camunda@zielona.example
```

Skąd te klucze i wartości:

Fragment	Źródło i uwaga
`global.identity.auth.*` z typem `MICROSOFT`	komentarz „Mode B: External OIDC Provider” w `values.yaml` wykresu 15.4.0 i strona dokumentacji o Entra
`identity.externalDatabase.*`	wymagane w trybie zewnętrznego OIDC, bo bez Keycloaka Management Identity potrzebuje własnej bazy
`identity.contextPath`	według opisu w `values.yaml` niepusta ścieżka wymaga HTTPS; mamy HTTPS na Traefiku
`usernameClaim`, `clientIdClaim`, `preferUsernameClaim`	wartości z przykładu dokumentacji dla Entra; w `values.yaml` te klucze są oznaczone jako przestarzałe, więc przy kolejnej wersji sprawdzamy ich następców
zmienne `env`	przykład dokumentacji: wyłączone zapytanie o dane użytkownika i większy limit nagłówka, bo tokeny Entra bywają duże

Fragment	Źródło i uwaga
`camundaHub.restapi.mail.fromAddress`	z przykładu dokumentacji; serwer SMTP (MailCow, `poczta.zielona.example`) ustawiamy kluczami `spring.mail.*` według dokumentacji Hub, których tu nie zgadujemy

4. Wdrożenie

```
cd /k8s/helm/camunda
helm template camunda camunda/camunda-platform --version 15.4.0 -n camunda -f values.yaml -f
values-entra.yaml > podglad-entra.yaml
helm upgrade camunda camunda/camunda-platform --version 15.4.0 -n camunda -f values.yaml -f
values-entra.yaml
kubectl -n camunda get pods -w
kubectl -n camunda get ingress
```

Po zmianie dochodzą pody Management Identity i Camunda Hub (REST API, aplikacja WWW, WebSockets), a Ingress dostaje ścieżki `/identity` i `/modeler`. Od 8.10 klaster startuje nawet wtedy, gdy dostawca OIDC jest nieosiągalny, i tylko zapisuje ostrzeżenia, więc „wszystkie pody Ready” nie znaczy jeszcze „logowanie działa”.

5. Pierwsze logowanie i reguły mapowania

W oknie prywatnym otwieramy `https://bpm.zielona.example/orchestration`, logujemy się kontem Oli i sprawdzamy, że Operate pokazuje jej nazwę. Potem `https://bpm.zielona.example/identity` (Management Identity) i `https://bpm.zielona.example/modeler` (Hub).

Reguły mapowania według dokumentacji: dla Web Modelera reguła na rolę „Web Modeler” (w 8.10 role Hub, a stare role zostają dla zgodności), dla dostępu do API Web Modelera osobna rola i reguła na token klienta. W Admin Orchestration Cluster zakładamy reguły dla Tasklist i Operate. Grupy z Entra trafiają do tokenu dopiero po włączeniu roszczenia `groups` w „Token configuration”, a ich wartością są identyfikatory grup (GUID), nie nazwy; dokumentacja Camundy dla Entra grup nie opisuje, więc tę część sprawdzamy w laboratorium.

Grupa w AD	Reguła	Rola
`GG-BPM-Admini`	`groups` zawiera identyfikator grupy	admin w Orchestration Cluster, administrator Hub
`GG-BPM-Modelarze` (Ola, pani Halina)	jak wyżej	modelowanie w Hub
`GG-BPM-Magazyn` (Marek i magazyn)	jak wyżej	użytkownik Tasklist

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran Management Identity → Mapping rules z regułą dla grupy `GG-BPM-Modelarze`, widoczne pola „Claim name: groups”, „Claim value: (GUID grupy)” i przypisana rola.

6. Pilnowanie sekretów

Sekret klienta w Entra wygasa w dniu ustawionym przy tworzeniu (u nas rok). Po tym dniu logowanie przestaje działać nagle, zwykle z błędem „invalid_client” w dzienniku poda. Raz w miesiącu sprawdzamy daty kartą „Daty wygaśnięcia sekretów”, a 30 dni przed końcem zakładamy w GLPI zadanie wymiany:

1. Nowy sekret z `--append` (stary nadal działa).
2. Nowa wartość w Vaultwardenie i w sekrecie `entra-credentials` (`kubectl -n camunda edit secret` albo `delete` i `create`).
3. `kubectl -n camunda rollout restart` dla wdrożeń, które czytają sekret, i test logowania.
4. Usunięcie starego sekretu w portalu po potwierdzeniu.

7. Droga bez Microsoftu: Keycloak albo Authentik

Wariant z Keycloakiem: Keycloak 26 z operatorem i bazą `keycloak` w `pg-bpm`, konta z FreeIPA przez federację LDAP. Instalację operatora i realmu `camunda-platform` robimy według strony Camundy „Deploy required dependencies with Kubernetes operators”, a wykres Camundy tylko się do niego łączy (komentarz „Mode A: External Keycloak” w `values.yaml`):

```
global:
  security:
    authentication:
      method: oidc
  identity:
    keycloak:
      internal: true
      url:
        protocol: "http"
        host: "NazwaUslugiKeycloak"
        port: "PortUslugiKeycloak"
      contextPath: "/auth"
```

```
      auth:
        adminUser: "admin"
        secret:
          existingSecret: "keycloak-admin"
          existingSecretKey: "admin-password"
    auth:
      enabled: true
      publicIssuerUrl: "https://bpm.zielona.example/auth/realms/camunda-platform"
identity:
  enabled: true
```

Hasło administratora Keycloaka leży w `auth.secret`, tak jak w strukturze `values.yaml` wykresu 15.4.0 (skrócony przykład w komentarzu wykresu podaje je o poziom wyżej). `internal: true` każe wspólnemu Ingressowi przekazywać ścieżkę `/auth` do Keycloaka, więc logowanie zostaje pod tym samym adresem. Nazwę usługi i port bierzemy z obiektu `Keycloak` po instalacji operatora (`kubectl get svc`). Dokumentacja Camundy ostrzega, że obrazy Keycloaka starsze niż `quay-optimized-26.6.4` mogą psuć żądania HTTP/2 bez szyfrowania; wtedy pomaga nowszy obraz albo `QUARKUS_HTTP_HTTP2=false`.

Wariant z Authentik: Authentik jako dowolny dostawca OIDC, tryb `GENERIC` z tymi samymi kluczami co dla Entra, tylko z adresami Authentik. Adres wystawcy ma w Authentik postać z nazwą aplikacji (według dokumentacji Authentik, do potwierdzenia przy instalacji), a cztery adresy odczytujemy tak samo z `.well-known/openid-configuration`:

```
global:
  security:
    authentication:
      method: oidc
  identity:
    auth:
      enabled: true
      type: GENERIC
      publicIssuerUrl: https://auth.zielona.example/application/o/NazwaAplikacji/
      issuerBackendUrl: https://auth.zielona.example/application/o/NazwaAplikacji/
      tokenUrl: AdresTokenuZOpenidConfiguration
      authUrl: AdresAutoryzacjiZOpenidConfiguration
      jwksUrl: AdresJwksZOpenidConfiguration
```

Authentik ma certyfikat z urzędu Zielonej, więc Camunda musi mu ufać: certyfikat główny podajemy w `global.tls.caBundle.secret` (wykres składa go w opcje Javy), inaczej w dzienniku pojawia się `PKIX path building failed`.

Weryfikacja

```
kubectl -n camunda get pods
kubectl -n camunda get ingress -o wide
curl -s -o /dev/null -w "%{http_code} %{redirect_url}\n"
https://bpm.zielona.example/orchestration/
```

Oczekujemy podów Identity i Hub w stanie gotowym, Ingressu ze ścieżkami `/orchestration`, `/connectors`, `/identity` i `/modeler` oraz przekierowania 302 na `login.microsoftonline.com` (albo na `/auth` w wariancie z Keycloakiem). Konto Oli loguje się do Operate, Tasklist, Management Identity i Hub, a konto spoza grup BPM dostaje 403.

Wycofanie zmian

`helm upgrade` z samym `values.yaml` (bez `values-entra.yaml`) wraca do logowania Basic i wyłącza Identity i Hub. Bazy `identity` i `hub` zostają w `pg-bpm` z danymi, więc ponowne włączenie nie zaczyna od zera; reguła pierwszego administratora nie zostanie wtedy założona drugi raz. Rejestracje w Entra usuwamy w portalu dopiero po decyzji, że wariant z Entra nie wróci.

Gdy coś nie zadziała

Objaw	Przyczyna	Co robimy
Po logowaniu w Entra błąd wystawcy (issuer) w dzienniku	tokeny w wersji 1.0	`requestedAccessTokenVersion: 2` w każdej rejestracji
Entra: „AADSTS50011: redirect URI mismatch”	adres przekierowania różni się od zarejestrowanego (ścieżka, ukośnik, `http`)	adres z tabeli rejestracji, dokładnie co do znaku
Management Identity zwraca 403 zalogowanemu użytkownikowi	brak `preferred_username` w tokenie dostępu	opcjonalne roszczenie w „Token configuration”
Pierwszy administrator nie ma uprawnień	zła wartość `initialClaimValue`, a Identity już raz wystartował	poprawka w bazie `identity` według dokumentacji, bo Helm jej już nie zmieni
Logowanie działało rok i przestało, `invalid_client` w dzienniku	wygasły sekret klienta	procedura z kroku 6
`PKIX path building failed` przy Keycloaku albo Authentik	certyfikat z urzędu Zielonej nieznany Javie	`global.tls.caBundle.secret` z certyfikatem głównym

Objaw	Przyczyna	Co robimy
Błąd 400 o zbyt dużym nagłówku po logowaniu	duży token z wieloma grupami	`SERVER_MAX_HTTP_REQUEST_HEADER_SIZE` jak w kroku 3
Pętla przekierowań między `/identity` a Entra	`fullURL` albo `contextPath` niezgodne z regułą Ingress	ta sama ścieżka w `identity.contextPath`, `fullURL` i rejestracji

□□ Ćwiczenia

1. Dlaczego każda aplikacja Camundy ma osobną rejestrację w Entra, a nie jedną wspólną?

Odpowiedź

Bo identyfikator klienta jest zarazem odbiorcą tokenu. Przy jednej rejestracji token wydany dla Hub byłby ważny także dla Orchestration Cluster i Identity, a jedna wspólna tajemnica otwierałaby wszystko. Osobne rejestracje dają osobne sekrety, osobne uprawnienia i osobne daty wygaśnięcia.

1. Dlaczego jako pierwszego administratora wskazujemy `oid`, a nie adres e-mail?

Odpowiedź

`initialClaimValue` działa tylko przy pierwszym starcie i później nie da się go zmienić przez Helm. Adres e-mail może się zmienić (nazwisko, domena), a `oid` to stały identyfikator obiektu w Entra. Dokumentacja Camundy zaleca `oid` właśnie do produkcji.

1. Pan Wojtek pyta, co się stanie z dostępem do Camundy po odejściu pracownika. Co odpowiadamy przy Entra, a co przy Keycloaku z FreeIPA?

Odpowiedź

Przy Entra wyłączamy konto w AD, `sync01` przenosi to do Entra, a pracownik nie dostanie nowego tokenu; dotychczasowa sesja wygaśnie z tokenem. Przy Keycloaku z federacją z FreeIPA wyłączamy konto w FreeIPA i Keycloak przestaje je wpuszczać. W obu przypadkach reguły mapowania oparte na grupach nie wymagają zmian w Camundzie.

1. Logowanie przez Keycloak działa, ale co jakiś czas żądania kończą się dziwnymi błędami protokołu. Co sprawdzamy według dokumentacji Camundy?

Odpowiedź

Wersję obrazu Keycloaka: obrazy starsze niż `quay-optimized-26.6.4` mogą psuć żądania HTTP/2 bez szyfrowania. Podnosimy obraz albo ustawiamy `QUARKUS_HTTP_HTTP2=false`.

Powiązane: Camunda 8 Self-Managed przez Helm: instalacja próbna na bpm01 · Entra Connect i Cloud Sync: synchronizacja ad.zielona.example z Entra ID · FreeIPA: jedno logowanie dla wszystkich linuksów · Konta z Active Directory: Directory Connector i SSO · Częste problemy z silnikami procesów: Traefik 502, Zeebe, PVC i OIDC

Źródła (odczyt 1 listopada 2026):

1. docs.camunda.io: Set up the Helm chart with an external Microsoft Entra tenant (wersja 8.10: pięć rejestracji, platformy Web i SPA, adresy przekierowania, zakresy, `requestedAccessTokenVersion: 2`, `preferred_username`, sekret `entra-credentials`, `initialClaimName`, zalecenie `oid`, klucze `orchestration`, `connectors`, `camundaHub`)
2. github.com/camunda/camunda-platform-helm: `charts/camunda-platform-8.10/values.yaml` w wersji 15.4.0 (komentarze „Mode A: External Keycloak” i „Mode B: External OIDC Provider”, `identity.externalDatabase.*`, `identity.contextPath` wymaga HTTPS, `global.identity.auth.type`: `KEYCLOAK`, `MICROSOFT`, `GENERIC`, `global.tls.caBundle`, przestarzałe klucze `usernameClaim` i pokrewne)
3. docs.camunda.io: 8.10 Release announcements (start mimo nieosiągalnego dostawcy OIDC, nowe role Hub, usunięty podwykres Keycloaka) i What's new in 8.10 (mapowanie ról 8.9 na 8.10)
4. docs.camunda.io: Deploy required dependencies with Kubernetes operators (operator Keycloaka, obraz od Camundy, `/auth`, ostrzeżenie o `quay-optimized-26.6.4`) i Supported environments (Keycloak 26.x)
5. Adres wystawcy Authentik i dostęp Google Workspace: nie sprawdzaliśmy w tej sesji, oznaczone w tekście jako do potwierdzenia.

Revision #2
Created 2026-11-01 18:02:12 UTC by Tomasz Konieczny
Updated 2026-11-01 18:03:37 UTC by Tomasz Konieczny