Jenkins: logowanie kontem z Active Directory

Typ: procedura

Dotyczy: Jenkins LTS, Active Directory Plugin, Active Directory na dc01/dc02

Czas: ~35 min

Efekt: logowanie do Jenkinsa kontem domenowym z Active Directory, z automatycznym nadaniem uprawnień administratora osobom z grupy `GG_Jenkins_Admin`, oraz awaryjne wejście na wypadek, gdyby kontrolery domeny akurat nie odpowiadały.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
CA	Certificate Authority: urząd certyfikacji, który podpisuje certyfikaty
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
LDAP	Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)
LDAPS	LDAP szyfrowany TLS, zwykle port 636
LTS	Long Term Support: wersja z długim wsparciem poprawkami

Zanim zaczniemy

> **Z ŻYCIA WZIĘTE**
>
> Tomek dostaje dostęp do Jenkinsa drugiego tygodnia pracy. Zamiast czekać, aż administrator założy mu konto lokalne w Jenkinsie i wyśle hasło mailem, loguje się od razu tym samym kontem domenowym, którym loguje się do komputera i do Gitei.

Jenkins umie sprawdzać hasła w katalogu LDAP na dwa sposoby: ogólną wtyczką **LDAP Plugin**, tą samą co w Gitea czy innych aplikacjach, albo dedykowaną wtyczką **Active Directory Plugin**, napisaną specjalnie pod Microsoft Active Directory. Obie łączą się tym samym protokołem LDAP, opisanym na stronie Logowanie kontem z Active Directory książki Gitea, różnią się tym, ile pracy

wykonują za nas automatycznie.

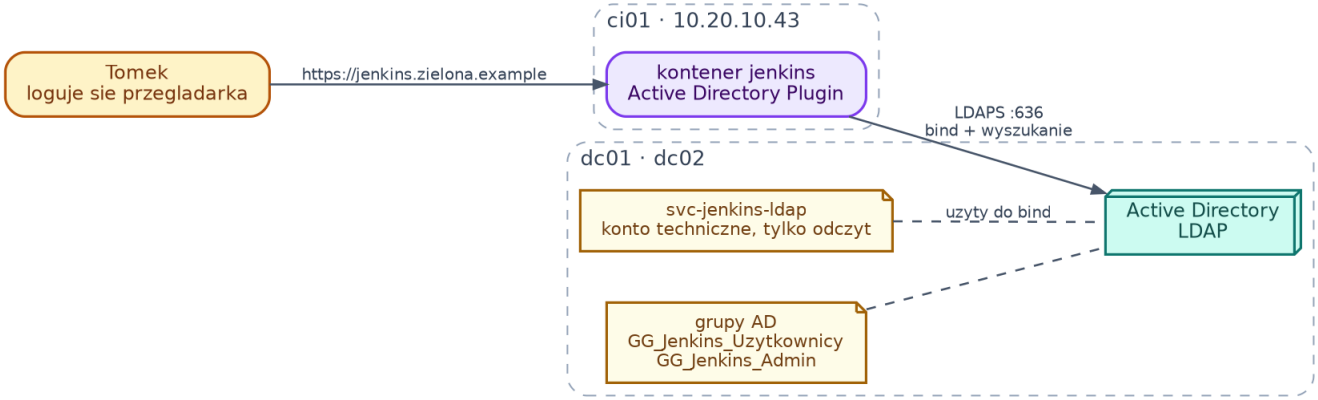
Cecha	LDAP Plugin	Active Directory Plugin
Adres kontrolera domeny	wpisujemy ręcznie, z zapasowym wpisem na wypadek awarii	znajduje sam przez rekordy DNS SRV domeny, bez wpisywania `dc01` czy `dc02`
Grupy zagnieżdżone	wymaga reguły `LDAP_MATCHING_RULE_IN_CHAIN` w filtrze wyszukiwania	rozpoznaje zagnieżdżone grupy bez dodatkowej reguły
Konto awaryjne, gdy katalog nie odpowiada	brak wbudowanego, trzeba ręcznie edytować `config.xml`	pole „Use Jenkins Internal Database", patrz krok 5
Dla jakiego katalogu	dowolny serwer LDAP: Active Directory, FreeIPA, OpenLDAP	wyłącznie Microsoft Active Directory

W Zielonej, z jednym lasem Active Directory i dwoma kontrolerami domeny, wybieramy Active Directory Plugin: mniej pól do wypełnienia i wbudowane konto awaryjne, którego LDAP Plugin nie ma.

Wymaganie	Uwaga
Konto techniczne w AD do bind LDAP	zakładamy w kroku 1, prawo tylko do odczytu katalogu
Dwie grupy w AD: użytkownicy i administratorzy Jenkinsa	zakładamy w kroku 1
LDAPS na porcie 636 osiągalny z `ci01` do `dc01` / `dc02`	sprawdzamy zaporą, zanim zaczniemy konfigurację w Jenkinsie
Zaufanie kontrolera do certyfikatu CA-Zielona	patrz krok 4, inaczej połączenie LDAPS kończy się błędem

> Po przenosinach urzędu według strony Przenosiny urzędu z dc01 bez przerwy: dwa urzędy równolegle i wygaszenie CA-Zielona dokładamy tu w ten sam sposób zaufanie do `Zielona-Root-CA`, a `CA-Zielona.crt` zostaje obok do wygaszenia starego urzędu.

Uważaj: konto techniczne wskazane przy bind LDAP powinno mieć wyłącznie prawo odczytu katalogu, na tej samej zasadzie najmniejszych uprawnień, na jakiej opieramy konta monitorujące bazy danych. Jenkins nie potrzebuje żadnego prawa zapisu w Active Directory do samego uwierzytelniania użytkowników.



Kontener jenkins na ci01 łączy się z Active Directory na dc01 po LDAPS, przez konto techniczne svc-jenkins-ldap i dwie grupy uprawnień (kliknięcie otwiera pełny rozmiar)

Przebieg

1. Konto techniczne i grupy w Active Directory

Zakładamy konto techniczne do bind LDAP, z hasłem nieulegającym wygaśnięciu i bez prawa jego zmiany przez samo konto:

```
New-ADUser -Name "svc-jenkins-ldap" `
  -SamAccountName "svc-jenkins-ldap" `
  -UserPrincipalName "svc-jenkins-ldap@ad.zielona.example" `
  -Path "OU=KontaTechniczne,DC=ad,DC=zielona,DC=example" `
  -AccountPassword (Read-Host -AsSecureString "Haslo konta") `
  -Enabled $true `
  -PasswordNeverExpires $true `
  -CannotChangePassword $true
```

Zakładamy dwie grupy globalne, zgodnie z konwencją nazewniczą `GG_<Aplikacja>_<Rola>` używaną już przy innych aplikacjach w Zielonej: jedną dla wszystkich osób z dostępem do Jenkinsa, drugą dla administratorów.

```
New-ADGroup -Name "GG_Jenkins_Uzytkownicy" -GroupScope Global -GroupCategory Security -Path
"OU=Grupy,DC=ad,DC=zielona,DC=example"
```

```
New-ADGroup -Name "GG_Jenkins_Admin" -GroupScope Global -GroupCategory Security -Path
"OU=Grupy,DC=ad,DC=zielona,DC=example"
```

Dodajemy Olę do grupy administratorów:

```
Add-ADGroupMember -Identity "GG_Jenkins_Admin" -Members "ola.wisniewska"
```

2. Instalacja wtyczki

W Manage Jenkins → Plugins → Available plugins szukamy „Active Directory" i instalujemy, zaznaczając „Restart Jenkins when installation is complete and no jobs are running".

```
Manage Jenkins → Plugins → Available plugins → "Active Directory" → Install
```

3. Konfiguracja Security Realm

W Manage Jenkins → Security, w sekcji Security Realm wybieramy „Active Directory" i wypełniamy formularz.

```
Manage Jenkins → Security → Security Realm → Active Directory
```

Pole	Wartość w Zielonej	Uwaga
Domain Name	`ad.zielona.example`	wystarczy sama nazwa domeny, wtyczka sama znajdzie kontrolery przez DNS
Domain Controllers	`dc01.ad.zielona.example:636` `dc02.ad.zielona.example:636`	opcjonalne, wpisujemy tylko jeśli automatyczne wykrywanie zawiedzie
Site	puste	ma znaczenie tylko przy kilku lokalizacjach z osobnymi kontrolerami, w Zielonej jedna
Bind DN	`CN=svc-jenkins-ldap,OU=KontaTechniczne,DC=ad,DC=zielona,DC=example`	konto techniczne z kroku 1
Bind Password	hasło konta technicznego	to samo hasło co przy `New-ADUser`
Start TLS / LDAPS	LDAPS, port 636	ruch szyfrowany, patrz krok 4

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran Manage Jenkins → Security z sekcją Security Realm ustawioną na „Active Directory", widoczne pole Domain Name z wartością „ad.zielona.example" i

zaznaczone LDAPS.

4. Zaufanie do firmowego CA

Kontener Jenkinsa działa na maszynie wirtualnej Javy, która trzyma własny magazyn zaufanych certyfikatów, osobny od systemowego magazynu hosta. Certyfikat, którym `dc01` i `dc02` podpisują LDAPS, musi trafić właśnie do tego magazynu Javy, inaczej połączenie kończy się błędem certyfikatu, mimo poprawnej reszty konfiguracji.

```
sudo keytool -importcert -noprompt \
  -alias ca-zielona \
  -file /SciezkaDoPliku/CA-Zielona.crt \
  -keystore /docker/apps/jenkins/CA-Zielona-cacerts \
  -storepass changeit
```

(gdzie `/docker/apps/jenkins/CA-Zielona-cacerts` to kopia domyślnego magazynu `cacerts` z obrazu Jenkinsa, z dopisanym naszym certyfikatem CA, wskazywana potem obrazowi kontenera zmienną `JAVA_OPTS=-Djavax.net.ssl.trustStore=...`)

> **Pojęcie: keytool.** Narzędzie z pakietu Javy do zarządzania magazynami kluczy i certyfikatów w formacie `cacerts`: dodaje, usuwa i wypisuje certyfikaty, którym dana maszyna wirtualna Javy ufa niezależnie od tego, co na temat zaufania myśli system operacyjny, na którym ta maszyna działa.

5. Konto awaryjne na wypadek niedostępności AD

W tym samym formularzu Security Realm zaznaczamy opcję „Use Jenkins Internal Database" i wskazujemy dokładnie jedno istniejące konto lokalne administratora jako login konta awaryjnego. Gdy oba kontrolery domeny są niedostępne (awaria sieci, planowana przerwa techniczna na `dc01` i `dc02` naraz), logowanie tym jednym kontem lokalnym wciąż działa, zamiast całkowicie zablokować dostęp do Jenkinsa.

> To konto lokalne trzyma się z tym samym rygorem co każde inne konto z pełnymi uprawnieniami: silne, unikalne hasło, znane wyłącznie zespołowi IT, nigdy nie używane do codziennej pracy.

6. Strategia autoryzacji

Kto po zalogowaniu co widzi i co może kliknąć, ustawiamy macierzą uprawnień albo strategią opartą na rolach, dokładnie tak, jak opisuje strona Poświadczenia i bezpieczeństwo. W Zielonej stosujemy Role-based Authorization Strategy: rolę `admin` z pełnią uprawnień przypisujemy grupie `GG_Jenkins_Admin`, a rolę `deweloper` (odczyt, budowanie, bez zmiany konfiguracji systemu) przypisujemy grupie `GG_Jenkins_Uzytkownicy`.

```
Manage Jenkins → Manage and Assign Roles → Assign Roles
```

Weryfikacja

Testujemy logowanie kontem domenowym na ekranie logowania Jenkinsa, loginem `tomek.nowak` i hasłem domenowym, zamiast kontem lokalnym.

```
docker compose logs -f jenkins | grep -i "active directory\|ldap"
```

Przykład:

```
jenkins  | INFO ActiveDirectorySecurityRealm - Domain ad.zielona.example resolved to
dc01.ad.zielona.example, dc02.ad.zielona.example
jenkins  | INFO ActiveDirectorySecurityRealm - tomek.nowak authenticated successfully
```

Sprawdzamy w Manage Jenkins → Users, że konto `tomek.nowak` pojawiło się na liście po pierwszym logowaniu, oraz w Manage Jenkins → Manage and Assign Roles, że dostał rolę zgodną z przynależnością do grupy.

Wycofanie zmian

Powrót do logowania kontem wbudowanej bazy Jenkinsa, gdy trzeba tymczasowo wyłączyć Active Directory: w Manage Jenkins → Security zmieniamy Security Realm z powrotem na „Jenkins' own user database" i zapisujemy. Konta utworzone wcześniej przez logowanie domenowe zostają widoczne na liście użytkowników, ale nikt nimi już się nie zaloguje, dopóki realm nie wróci na Active Directory.

Gdy coś nie zadziała

Objaw	Przyczyna	Co robimy
`Failed to connect to any domain controller`	LDAPS na porcie 636 zablokowany zaporą między `ci01` a `dc01`/`dc02`	sprawdzamy regułę na FortiGate i osiągalność portu z poziomu `ci01`
`x509: certificate signed by unknown authority` w logu kontenera	certyfikat CA-Zielona brakuje w magazynie `cacerts` używanym przez JVM kontenera	powtarzamy krok 4 i restartujemy kontener
Konto istnieje w AD, ale logowanie odrzucone	konto poza grupą `GG_Jenkins_Uzytkownicy`, a Jenkins skonfigurowany, żeby wpuszczać tylko tę grupę	dodajemy konto do grupy, także pośrednio przez grupę zagnieżdżoną
Konto loguje się, ale nie widzi uprawnień administratora mimo członkostwa w `GG_Jenkins_Admin`	rola `admin` w Manage and Assign Roles nie ma jeszcze przypisanej tej grupy	uzupełniamy przypisanie w Assign Roles, bez restartu Jenkinsa
Oba kontrolery domeny padły, nikt nie może się zalogować	konto awaryjne z kroku 5 nie zostało włączone przed awarią	jedyne wyjście to edycja `config.xml` w `JENKINS_HOME`: zatrzymujemy Jenkinsa, usuwamy sekcje `<securityRealm>` i `<authorizationStrategy>`, zmieniamy `<useSecurity>` na `false`, uruchamiamy ponownie, logujemy się bez hasła i włączamy konto awaryjne, zanim przywrócimy zabezpieczenia

Powiązane: Poświadczenia i bezpieczeństwo · Logowanie kontem z Active Directory (książka Gitea) · Active Directory: konta, grupy, jednostki organizacyjne

Źródła: plugins.jenkins.io/active-directory · jenkins.io/doc/book/security/access-control/disable · `keytool` man page

---

Revision #23
Created 2026-09-28 16:57:46 UTC by Tomasz Konieczny
Updated 2026-10-26 14:15:20 UTC by Tomasz Konieczny