

18.11.2025 – DOAG 2025 Konferenz + Ausstellung

Mit SQLcl und GitLab-Pipelines den CI/CD-Prozess optimieren

Automatisierung von CI/CD für APEX-Anwendungen mittels SQLcl Projects und GitLab-Pipelines

Maurice Wilhelm

code of change

 **Hyand** by
GOD | MT

Über mich

Maurice Wilhelm

Oracle APEX Consultant @Hyand

8+ Jahre Erfahrung mit Oracle-Technologien

APEX-Enthusiast seit Beginn

Interessensschwerpunkte auf DevOps, CI/CD und Automation



High delivery reliability and international network of locations

Hamburg
Wolfsburg
2 x Braunschweig
2 x Berlin
Dortmund
Ratingen
Cologne
Frankfurt
Munich
Wrocław
Kaunas
Vilnius

Cluj-Napoca

Pune

> 900

Employees

(of which 160 in shoring
locations)

> 150

Clients

> 10

Industries

16

Locations

Agenda

- Motivation
- GitLab Runner und Repositories
- CI/CD Variablen
- Datenbank-Proxy User
- Steuerung der GitLab API mittels APEX-App
- Chancen und Herausforderungen
- Migration von bestehenden SQLcl LB-Projekten
- Upgrade eines SQLcl-Projektes

SQLcl Projects

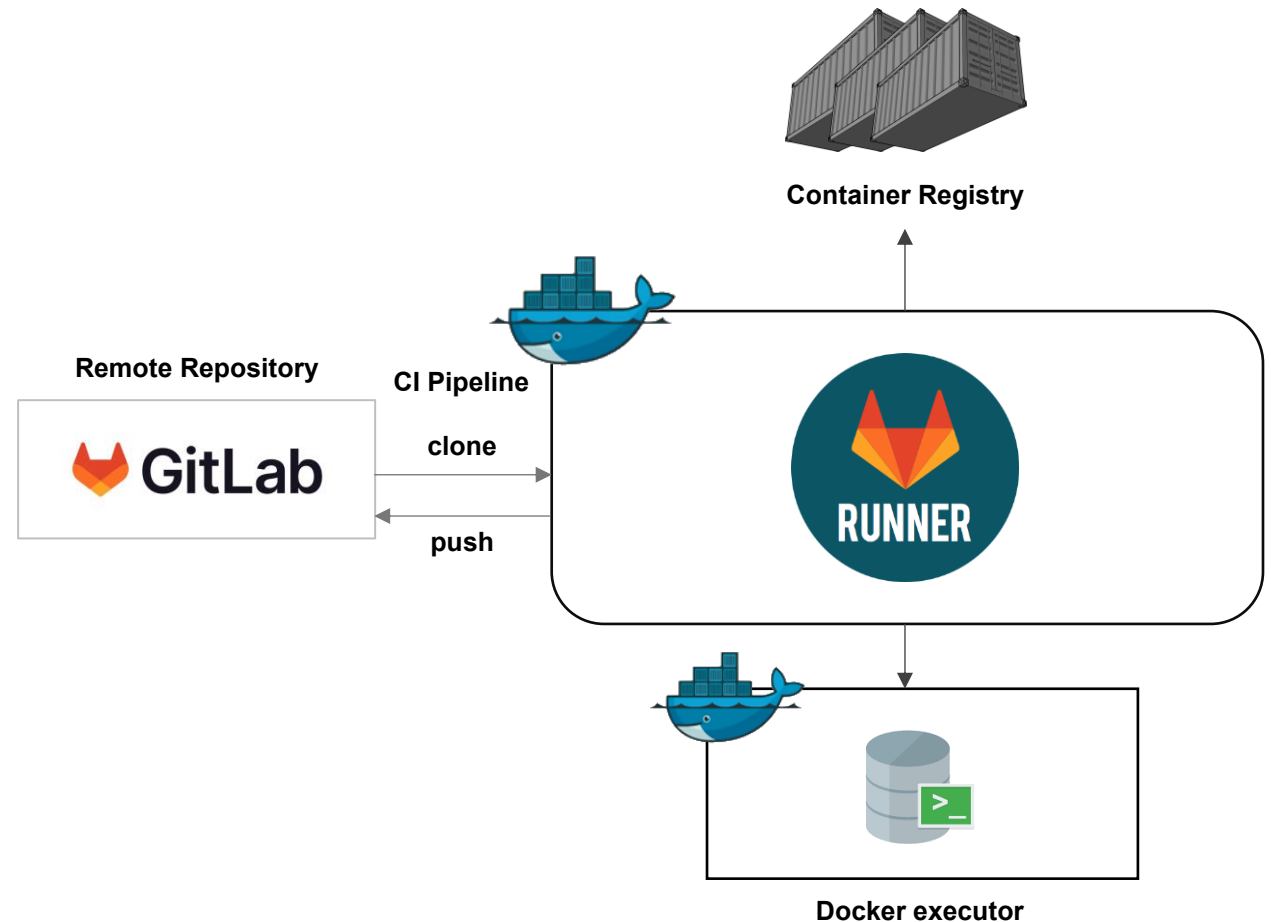
Versionierung vereinfachen - Fehler minimieren:

Mit SQLcl und GitLab-Pipelines den CI/CD-Prozess optimieren

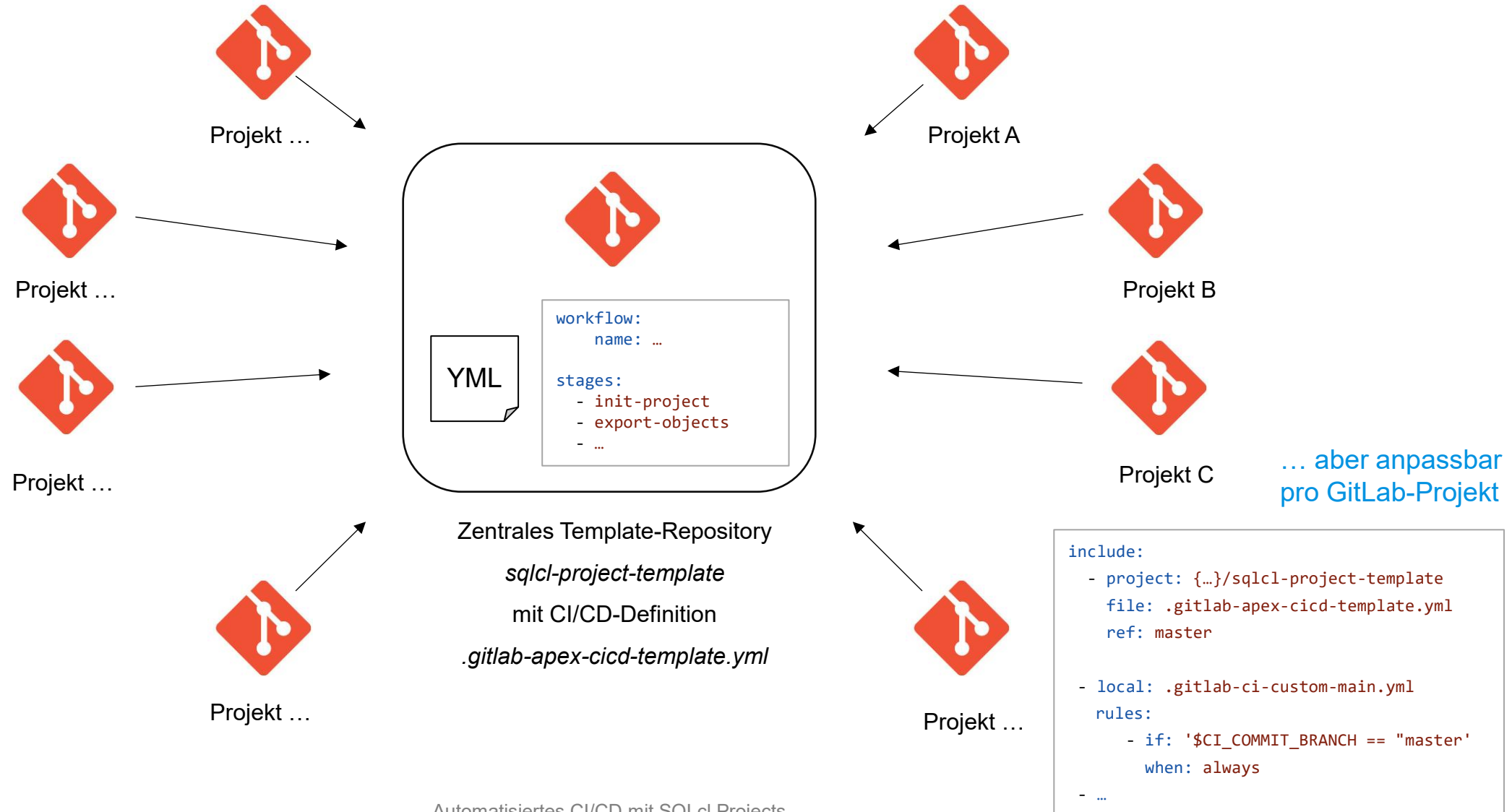
- SQLcl v24.3: Neues „project“-Kommando
- Export von Datenbankobjekten, Generierung von inkrementellen Changesets (Liquibase v4.25)
- Changesets: Änderungen in „src“ werden mit „defaultBranch“ verglichen, ähnlich „git diff“
- **Beschleunigung des Entwicklungsprozesses** durch Automatisierung
- **Minimierung von Fehlern** bei Versionierung und Deployment

CI/CD Bash-Skript wird mittels GitLab Runner ausgeführt

- 2 Repositories
 - sqlcl-project-cicd
 - sqlcl-project-template
- GitLab pipeline
 - GitLab Runner (Docker executor)
 - Custom Docker Image auf Basis von SQLcl Image v25.3.0
 - Bash script cicd.sh



Ein Template für *alle* - GitLab-Jobs zentral definieren, Struktur gewinnen



Mit „include:“ greifen wir auf zentrale Pipeline-Definitionen zu

- CI/CD-Template in „Template-Repository“ ausgelagert
- Pro GitLab Projekt: nur wenige CI/CD-Variablen und eine „gitlab-ci.yml“ notwendig
- Datenbank-Verbindung:
 - Deployment-User und EZCONNECT-Details auf Gruppen-Ebene definierbar
 - Alternativ auch benutzerdefiniert pro GitLab-Projekt
- Pull/Push ins Repository per SSH
 - Private Key: globale CI/CD Variable (Gruppe)
 - Public Key: als „Deploy Key“ in mehreren GitLab-Projekten wiederverwendbar

```
include:  
- project: {...}/sqlcl-project-template  
  file: .gitlab-apex-cicd-template.yml  
  ref: master
```

Pipeline-Variablen werden auf Gruppen- und Repository-Ebene festgelegt

Group/Project	Key	Type	Flag(s)	Beispiel/Syntax
Group	VAR_GLOBAL_DB_DEV_EZCONNECT_ADDRESS	Variable	Visible	db.host.tld:port/service_name
	VAR_GLOBAL_DB_DEV_DEPLOYMENT_USER	Variable	Visible	my_proxy_user
	VAR_GLOBAL_DB_DEV_DEPLOYMENT_USER_PW	Variable	Masked	my_proxy_pw
	VAR_GLOBAL_SSH_PRIVATE_KEY_GITLAB (*)	File	Visible	-----BEGIN RSA PRIVATE KEY----- xyz -----END RSA PRIVATE KEY-----
Project	VAR_REPO_DB_DEV_PROJECT_USER	Variable	Visible	my_dev_schema_name

...

* Public Key von `VAR_GLOBAL_SSH_PRIVATE_KEY_GITLAB` muss als Deploy Key mit Schreibrechten im GitLab-Projekt gespeichert/freigeschaltet werden.

Für die Datenbank-Zugriffe kann ein Proxy-User genutzt werden

- Proxy-User für Deployments
- Seit Oracle 10g Release 2 verfügbar
- Minimale Rechte notwendig
- Nur ein Passwort muss gespeichert werden

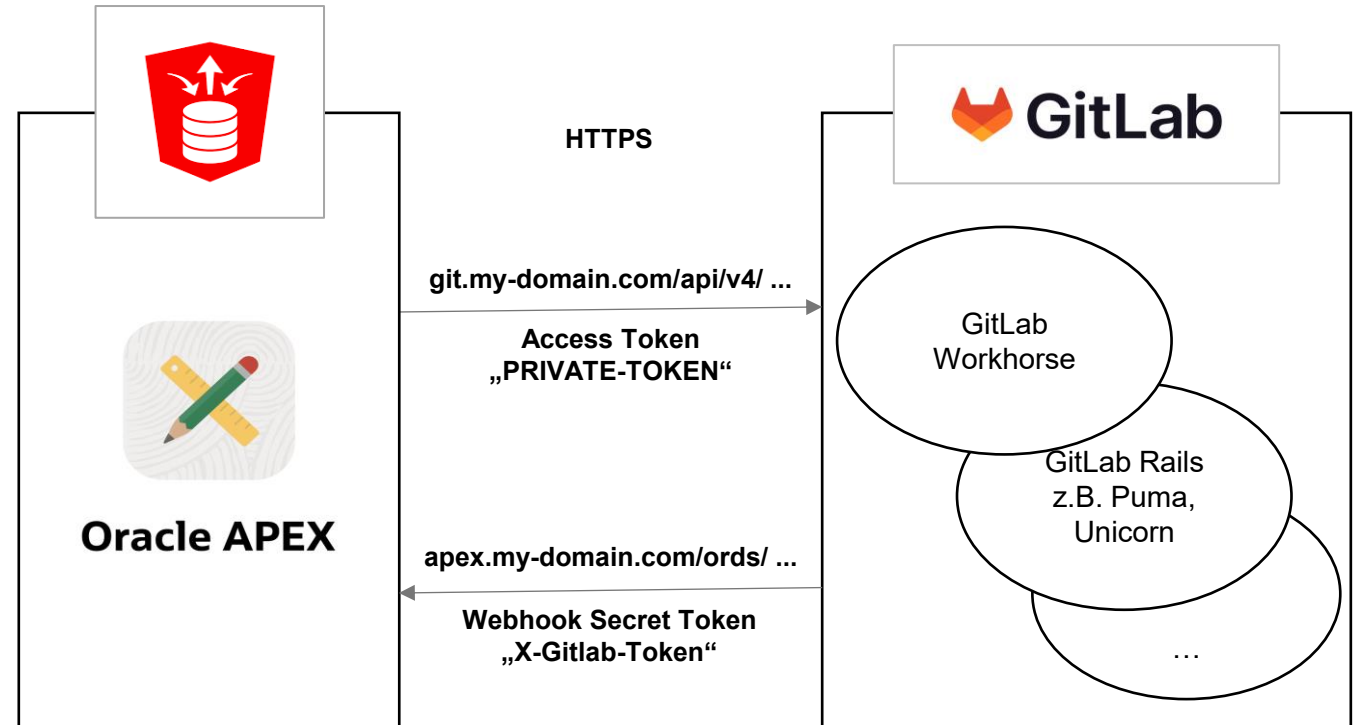
```
grant create session to {my_proxy_user};
```

```
alter user {my_dev_user} grant connect through  
{my_proxy_user};
```

```
connect my_proxy_user[my_dev_user]@...
```

Mittels APEX-Anwendung können Pipelines gesteuert und Job-Logs nachverfolgt werden

- APEX-Anwendung für das Anlegen von Projekten
- Pipelines starten, Logs einsehen
- Komfortablere UI/UX
- Kommunikation: Access Token und Webhook Secret Token



SQLcl Projects ermöglicht automatisierten CI/CD Workflow

Vorteile

- Objekte können dynamisch exportiert werden
(DBMS_METADATA.GET_DDL)
- Automatische Generierung von Liquibase Changesets
- Verringert händischen Aufwand für CI/CD
- Universell einsetzbar

Nachteile

- Filterung: nur „where“ clauses, keine Transformationen
- Wenn lokal eingesetzt: Entwickler müssen mit „project“-Kommando (+LB) vertraut sein
- Liquibase Deployments: „Move forward“
- Schemareferenzen
- Export der APEX-Metadaten nur in Form von „vollständiger App“

Existierende SQLcl Liquibase-Projekte können migriert werden

- Oracle: Hilfe bereitstellen
- Weg 1: Wechsel zwischen Releases
- Weg 2: Alle Changesets migrieren:
 1. Branch(es) erstellen und project init, export, stage ausführen
 2. Release per Kommando erstellen
 3. dist/install.sql anpassen: „liquibase update“ in „liquibase changelog-sync“ ändern (oder manuell ausführen)
 4. Alte Liquibase-Objekte in Zieldatenbank entfernen (optional)
 5. Deployment per Kommando ausführen

Ein SQLcl-Upgrade ist unkompliziert möglich

1. Neuen Branch erstellen
2. sqlcl->version in .dbtools/project.config.json ändern
3. „project export“ ausführen – **ohne** staging
4. Branch in „development“ zurückmergen

Es existieren Alternativen aus der Oracle-Community zu SQLcl Projects

 README  MIT license 

ADT - APEX Deployment Tool

ADT is an open-source tool written in Python which allows you to connect to your Oracle database, export objects and data and APEX applications, files and individual components into a folder structure. It helps you to automate the patching and deploying/migrating your changes to other environments in multiple variants.

It does not store anything in your database.

I have been building these CI/CD tools since 2008 and ADT is the newest version, heavily based on the previous [OPY](#) tool, which unfortunately outgrown to a hefty spaghetti code and became more and more difficult to extend. So, I have decided to start from scratch for like 15th time...

[GitHub - jkvetina/ADT: APEX Deployment Tool](https://github.com/jkvetina/ADT: APEX Deployment Tool) @Jan Květina

Demo

Bildquellen

<https://practicalbytes.de/managed-gitlab/> (abgerufen am 15.11.2025)

<https://oneclick-cloud.com/de/blog/trends/docker/> (abgerufen am 15.11.2025)

<https://gitlab.com/gitlab-community/gitlab-org/gitlab-runner> (abgerufen am 15.11.2025)

<https://www.oracle.com/database/technologies/appdev/sqlcl/sqlcl-faq.html> (abgerufen am 15.11.2025)

<https://pixabay.com/de/vectors/container-versand-fracht-logistik-147973> (abgerufen am 15.11.2025)

<https://git-scm.com/community/logos> (abgerufen am 15.11.2025)

<https://www.oappsnet.com/oracle-forms-migration-to-apex> (abgerufen am 15.11.2025)

<https://www.oracle.com/de/database/technologies/appdev/rest.html> (abgerufen am 15.11.2025)

<https://github.com/jkvetina/ADT> (abgerufen am 15.11.2025)

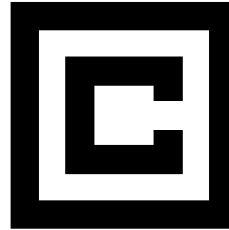
Vielen Dank für Ihre Aufmerksamkeit!

Say Hy_

+49 (0) 2102 30 961-0

<https://www.linkedin.com/in/maurice-wilhelm-511570272>

<https://blog.maurice-wilhelm.de>



© 2024 – The developed thoughts and ideas are the intellectual property of Hyand and are subject of copyright law. Reproduction, transfer to third parties or use – even of parts – is only permitted with the express of Hyand.