Lernjournal DevOps Frühlingssemester 2024

ZHAW SML  
DevOps FS2024  
  
Name: Nolè Vito Alessandro  
Matrikelnummer 19-684-067  
Datum: 26.05.2024

**Inhaltsverzeichnis**

[1 Einführung 3](#_Toc167647895)

[1.1 Person, Motivation, IT-Knowhow 3](#_Toc167647896)

[1.2 Umgebung Hardware/Software 3](#_Toc167647897)

[2 DevOps-Lernjournale 3](#_Toc167647898)

[2.1 Installation Software 3](#_Toc167647899)

[2.2 Versionskontrolle (Git) 7](#_Toc167647900)

[2.3 Build Tools 10](#_Toc167647901)

[2.3.1 Gradle 10](#_Toc167647902)

[2.3.2 NPM 12](#_Toc167647903)

[2.4 DevOpsDemo 14](#_Toc167647904)

[2.4.1 DevOpsDemo (Frontend) 14](#_Toc167647905)

[2.4.2 Spring Boot (Backend) 16](#_Toc167647906)

[2.5 Unit Tests 19](#_Toc167647907)

[2.6 Code Quality 23](#_Toc167647908)

[2.7 Integration Tests 27](#_Toc167647909)

[2.8 Containers 29](#_Toc167647910)

[2.9 Continuous Integration ½ 31](#_Toc167647911)

[2.10 Continuous Integration 2/2 33](#_Toc167647912)

[2.11 Docker Deployment 37](#_Toc167647913)

[2.12 Cloud Deployment 40](#_Toc167647914)

[2.13 DevopsDemo als Azure Web-App 45](#_Toc167647915)

[3 Schlussteil 49](#_Toc167647916)

[3.1 Reflexion, Zusammenfassung 49](#_Toc167647917)

[3.2 Feedback zur Vorlesung (optional) 50](#_Toc167647918)

Falls das Lernjournal im PDF eine schlechte Screenshot-Qualität anzeigt kann das Wordfile und die PowerPoint vom Screencast in diesem Repo heruntergeladen werden:

<https://github.com/nolevit1/lernjournal>

# Einführung

## Person, Motivation, IT-Knowhow

Mein Name ist Vito Alessandro Nolè und ich freue mich, die Reise durch die Welt von DevOps zu dokumentieren. Ich habe vor 5 Jahren eine NodeJs App in den App- und Playstore published (Discover Neuhausen) und habe auch mit Github gearbeitet. Meine Motivation für diesen Kurs liegt darin, die Zusammenarbeit zwischen Entwicklungs- und Betriebsteams zu sehen und dadurch qualitativ hochwertige Software schneller und zuverlässiger bereitzustellen.

## Umgebung Hardware/Software

Hardware: Mac mit M1 Prozessor. Seit dem Sommer 2023 auf Apple umgestiegen und bin auf die Installation und Funktionsweisen gespannt (hoffentlich keine Probleme).

Software: Homebrew Paketmanager für macOS, der die Installation von Software vereinfacht.

# DevOps-Lernjournale

## Installation Software

|  |  |
| --- | --- |
| GitHub-Konto | <https://github.com/nolevit1> |
| Git-Client |  |
| Java | Ein Bild, das Text, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Gradle | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Visual Studio Code | Ein Bild, das Text, Screenshot, Schrift, Software enthält.  Automatisch generierte Beschreibung |
| VS Extension Pack for Java | Ein Bild, das Text, Software, Multimedia-Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| VS Extension Gradle for Java | Ein Bild, das Text, Multimedia-Software, Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| VS Spring Boot Extension Pack | Ein Bild, das Text, Multimedia-Software, Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| VS Extension Coverage Gutters | Ein Bild, das Text, Multimedia-Software, Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| Node |  |
| NPM |  |
| Selenium IDE | Ein Bild, das Text, Software, Schrift, Screenshot enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Display enthält.  Automatisch generierte Beschreibung |
| Docker Desktop | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung  Ein Bild, das Screenshot, Text enthält.  Automatisch generierte Beschreibung |
| Docker Hub | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Jenkins | Ein Bild, das Screenshot, Text enthält.  Automatisch generierte Beschreibung |
| SonarQube | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Azure Portal | Ein Bild, das Screenshot, Text, Webseite, Website enthält.  Automatisch generierte Beschreibung |
| Azure Abonnement | Ein Bild, das Screenshot, Text, Webseite, Website enthält.  Automatisch generierte Beschreibung  Ein Bild, das Screenshot, Text, Software enthält.  Automatisch generierte Beschreibung |
| Azure CLI | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Postman | Ein Bild, das Screenshot, Text enthält.  Automatisch generierte Beschreibung |

## Versionskontrolle (Git)

|  |  |
| --- | --- |
| Person 1, Name/Kürzel | Nolè Vito Alessandro (nolevit1) |
| Person 1, Repo URL | <https://github.com/nolevit1/week_3> |
| Person 2, Name/Kürzel | Aghishenth Thayalakumar (thayaagh) |
| Person 2, Repo URL | <https://github.com/thayaagh/Montag> |
| Beschreibung Zusammenarbeit | Gegenseitiges Commiten von Codezeilen und Dateien, inkl. Branch mit Aghishenth. Zugriff gegenseitig als Collaborator. Klonen von seinem Projekt auf meinem Rechner und Änderungen vornehmen und commiten & pushen. Szenario als wäre er oder ich sein Lehrmeister der den Code überprüft. Dies ist der Prozess vom Programmieren bei einem Team. |
| Collaborator hinzufügen | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Klonen | Ein Bild, das Multimedia-Software, Text, Software, Grafiksoftware enthält.  Automatisch generierte Beschreibung |
| Änderungen vornehmen | Ein Bild, das Software, Screenshot, Multimedia-Software, Text enthält.  Automatisch generierte Beschreibung |
| Weitere Änderunge vornehmen und commited und pushen | Ein Bild, das Text, Multimedia-Software, Software, Grafiksoftware enthält.  Automatisch generierte Beschreibung |
| Ansicht im Github | Ein Bild, das Text, Software, Zahl, Screenshot enthält.  Automatisch generierte Beschreibung |
| Mit Files erweitern | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Weitere Änderungen | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Files ersichtlich im neuen Ordner | Ein Bild, das Text, Screenshot, Zahl, Software enthält.  Automatisch generierte Beschreibung |
| Textänderungen direkt ersichtlich | Ein Bild, das Text, Screenshot, Zahl enthält.  Automatisch generierte Beschreibung |
| Aghishenth seine Änderungen in meinem Projekt | Ein Bild, das Screenshot, Text enthält.  Automatisch generierte Beschreibung |
| Weitere Änderungen von Aghishenth in meinem Projekt | Ein Bild, das Text, Zahl, Software, Schrift enthält.  Automatisch generierte Beschreibung |
| Erstellung vom neuen Branch | Ein Bild, das Screenshot, Text, Grafiksoftware, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neue Datei | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Commiten vom Branch | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Pushen vom Branch | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Ansicht vom Branch von Aghishenth in meinem Projekt | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Akzeptieren vom Branch | Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Erfolgreiches Mergen | Ein Bild, das Text, Software, Webseite, Computersymbol enthält.  Automatisch generierte Beschreibung |

## Build Tools

### Gradle

|  |  |
| --- | --- |
| Repo URL  (wurde erst im nachhinein gepusht) | <https://github.com/nolevit1/sw4> |
| Namen der verwendeten Beispiele:  <https://search.maven.org> | Ein Bild, das Screenshot, Text, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Ascii render Library | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Testen und Erweitern | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |

### NPM

|  |  |
| --- | --- |
| Repo URL | <https://github.com/nolevit1/sw4npm> |
| Namen der verwendeten Beispiele | ChartJS und AnimeJS |
| NPM Install | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neuer Chart | Ein Bild, das Screenshot, Text, Diagramm enthält.  Automatisch generierte Beschreibung |
| AnimeJs | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neue Animation angepasst | Ein Bild, das Screenshot, Multimedia-Software, Software, Grafiksoftware enthält.  Automatisch generierte Beschreibung |

## DevOpsDemo

### DevOpsDemo (Frontend)

|  |  |
| --- | --- |
| Repo URL (Fork) | <https://github.com/nolevit1/DevOpsDemo> |
| Erfolgreichere lokale Installation und Ausführung:  Frontend funktioniert | Ein Bild, das Screenshot, Text, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neue Kachel für Mitarbeiteransicht im guimodel.ts  <https://github.com/nolevit1/DevOpsDemo/tree/main/frontend/src/app/gui-model> | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Neue Seite für Mitarbeiteransicht guimodel.ts  <https://github.com/nolevit1/DevOpsDemo/tree/main/frontend/src/app/gui-model> | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Frontend Ansicht mit:   * Titeländerung * Neue Kachel * Kachel mit Verlinkung auf eine neue Page | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Mitarbeiter Page mit geladenen Datensätzen |  |
| Form für Erfassung von einem neuen Mitarbeiter | Ein Bild, das Text, Screenshot, Diagramm, Software enthält.  Automatisch generierte BeschreibungEin Bild, das Screenshot, Text, Rechteck, Reihe enthält.  Automatisch generierte Beschreibung |
| Suche, Löschen und zurück Button funktionieren | Ein Bild, das Text, Software, Webseite, Computersymbol enthält.  Automatisch generierte Beschreibung |
| ToDo Kachel und Funktionen funktionieren | Ein Bild, das Text, Screenshot, Zahl, Reihe enthält.  Automatisch generierte Beschreibung |

### Spring Boot (Backend)

|  |  |
| --- | --- |
| Repo URL (Fork) | <https://github.com/nolevit1/DevOpsDemo> |
| Erfolgreichere lokale Installation und Ausführung:  Frontend funktioniert (siehe auch Init Data Todo und Init Data Employee) |  |
| Employee.java | <https://github.com/nolevit1/DevOpsDemo> /tree/main/backend/src/main/java/ch/zhaw/iwi/devops/demo  Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| EmployeeController.java | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Test | Ein Bild, das Text, Schrift, Screenshot, weiß enthält.  Automatisch generierte Beschreibung |
| Anzahl Mitarbeiter | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte Beschreibung |
| Alle Mitarbeiter |  |
| Test direkt auf Mitarbeiterdaten | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Get | Ein Bild, das Text, Software, Screenshot, Webseite enthält.  Automatisch generierte Beschreibung |
| Post | Ein Bild, das Screenshot, Text enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Zahl, Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| Put | Ein Bild, das Screenshot, Text enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Software, Zahl, Screenshot enthält.  Automatisch generierte Beschreibung |
| Delete | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Zahl, Schrift, Software enthält.  Automatisch generierte Beschreibung |

## Unit Tests

|  |  |
| --- | --- |
| Repo URL | <https://github.com/nolevit1/sw6> |
| Titel Idee | Score Keeper Kata (Basketball Punkte) |
| Beschreibung und Ziel der Software | Bereitstellung richtiger Daten auf den Anzeigetafeln der Basketballspiele. Jedes der beiden Teams (Team A und Team B) kann entweder 1, 2 oder 3 Punkte in einem Spielzug erzielen. |
| Quelle Idee | <https://kata-log.rocks/score-keeper-kata> |
| Schritt 1: Test schreiben  Als erster Schritt wird ein Test im AppTest.java geschrieben  Der Code wird  nicht kompiliert, da es die Klasse  ScoreKeeper gar noch nicht gibt. | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Schritt 2:  Nun wird die fehlende Klasse erstellt. Der Code kompiliert und der Test wird grün. | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Schritt 3:  **Nicht** die Logik weiterprogrammieren, sondern wieder einen **neuen Test** schreiben | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Schritt 4-5:  Nun wird der Code angepasst und es werden  alle bestehenden Tests gestartet: | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Schritt 6:  Klasse wird angepasst und test werden erfolgreich ausgeführt. | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Schritt 7:  Weitere Tests | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Schritt 8:  Code Anpassung | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Schritt 9-11:  Refactoring und weitere Test und Anpassungen nach dem Fehlschlagen der Test. | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Software, Schrift enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |

## Code Quality

|  |  |
| --- | --- |
| Repo URL | <https://github.com/nolevit1/DevOpsDemo> |
| Gradle – Tasks Verification | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Jacoco Test Coverage | Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Frontend Lint Test | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Fehler einbauen | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| *ES LINT* zeigt den Fehler an, damit man lint nicht immer ausführen muss | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Gradle Sonar | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Befehl | gradle sonar -Dsonar.projectKey=devopsdemo-backend -Dsonar.projectName='devopsdemo-backend' -Dsonar.host.url=http://localhost:9000 -Dsonar.token=sqp\_ecee7d3234c1c5f48c894f1f66a92ed9ccff6c07 |
| Sonarqube Dashboard | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Ansicht Backend mit allen Infos | Ein Bild, das Text, Software, Screenshot, Webseite enthält.  Automatisch generierte Beschreibung |
| Github Commit: Verweis Lernjournal X | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neue Analyse | gradle sonar -Dsonar.projectKey=devopsdemo-backend -Dsonar.projectName='devopsdemo-backend' -Dsonar.host.url=http://localhost:9000 -Dsonar.token=sqp\_ecee7d3234c1c5f48c894f1f66a92ed9ccff6c07 |
| Failed Fehler suchen | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Fehler Anpassen und neu commitend | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Frontend im SonarQube | npx sonar-scanner -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=devopsdemo-frontend -Dsonar.projectName=‚devopsdemo-frontend' -DSonar.token=sqp\_4b8d215e14cbb8cebe94193d4918695057c6ec47 |
| Success | Ein Bild, das Text, Schrift, Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| Neue Analyse | Ein Bild, das Screenshot, Text, Schrift enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Weitere Codeanpassungen und neue Analyse | Ein Bild, das Screenshot, Text, Reihe, Schrift enthält.  Automatisch generierte Beschreibung |
|  | Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung |

## Integration Tests

|  |  |
| --- | --- |
| Person 1, Name/Kürzel | Nolevit1 |
| Person 1, Repo URL | <https://github.com/nolevit1/week8-int-test> |
| Person 2, Name/Kürzel | Thayaagh |
| Person 2, Repo URL | <https://github.com/thayaagh/Week8_IntegrationTests> |
| Grundgedanke | <https://digitec.ch> Suchfeld nutzen und Produkte suchen |
| Selenium starten und neues Projekt anlegen | Ein Bild, das Text, Screenshot, Software, Display enthält.  Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.  Automatisch generierte Beschreibung |
| URL angeben auf denen die Aufnahme gestartet wird | Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.  Automatisch generierte Beschreibung |
| Test name definieren | Ein Bild, das Text, Elektronik, Screenshot, Software enthält.  Automatisch generierte Beschreibung |
| Maus und Tastatur wird aufgenommen währenddem ich auf digitec.ch im Suchfeld nach sonos suche.  Verify Text untersucht ob die Titelzeile «Sonos» lautet | Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte Beschreibung |
| Verify Text hier überprüft ob die Titelzeile «Sonnnos» ist, wenn nicht wird der Test nicht abgebrochen sondern läuft weiter | Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte Beschreibung |
| Beispiel wie es aussieht wenn Verify Text fehlschlägt | Ein Bild, das Text, Software, Webseite, Zahl enthält.  Automatisch generierte Beschreibung |
| Beispiel wie es aussieht wenn Verify Text fehlschlägt, aber trotzdem weitermacht und als nächstes im Suchfeld nach JBL sucht | Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte Beschreibung |
| Änderung von Verify Text zu Assert Text. Wenn hier der Test fehlschlägt, was in diesem Fall geschieht, weil Sonnos falsch geschrieben ist, wird der gesamte Test abgebrochen. | Ein Bild, das Text, Software, Zahl, Büroanwendungssoftware enthält.  Automatisch generierte Beschreibung |
| Best pratice Know-How Austausch mit Aghishenth | Aghishenth hat den Test mit Selenuim mit einer Meteo und der Microsoft seite getestet. Beide Optionen Verify Text und Assert Text wurden verwendet. Der negative Test etwas falsch anzugeben und dann zu testen ist von hoher Wichtigkeit, anstatt immer mit den richtigen Daten zu testen.  Ausserdem habe ich weitere Selenium Tests für die PowerPoint/Video erstellt und ausführlich im Video beschrieben. Diese sind auch im Repo ersichtlich. |

## Containers

|  |  |
| --- | --- |
| Voller Tag der verwendeten Images | <https://github.com/nolevit1/sw9> |
| Verwendetes Image 1: MySQL | INDEX DIGEST: sha256:4a4e5e2a19aab7a67870588952e8f401e17a330466ecfc55c9acf51196da5bd0  docker pull mysql:8.4.0-oracle |
| Verwendetes Image 1: Adminer | INDEX DIGEST: sha256:cbbcc8caedf029309ac7df722a470785e99a0942e29b3efc9367ee1640e1b330  docker pull adminer:4.8.1 |
| MySQL pull | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Adminer pull | Ein Bild, das Screenshot, Text enthält.  Automatisch generierte Beschreibung  Ein Bild, das Text, Schrift, Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| Setup von MySQL |  |
| Docker Desktop |  |
| Adminer setup |  |
| Docker Network nole-net erstellt wo beide Container hinzugefügt werden | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Wie das Netzwerk im Docker Desktop aussieht | Ein Bild, das Screenshot, Text, Reihe enthält.  Automatisch generierte Beschreibung |
| Öffnen und testen von Adminer mit der MySQL Datenbank | Anmeldung (PW: admin)  Ein Bild, das Text, Software, Screenshot, Webseite enthält.  Automatisch generierte Beschreibung |
| Auch das Erstellen neuer Datenbanken ist über die Adminer Öberfläche verfügbar | Ein Bild, das Text, Screenshot, Software, Display enthält.  Automatisch generierte Beschreibung |
| Alle Datenbanken sind ersichtlich | Ein Bild, das Text, Screenshot, Software, Zahl enthält.  Automatisch generierte Beschreibung |
| Erstellung von Tabellen und deren Attributen | Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Docker Compose vereinfacht die Konfiguration mehrerer Container.  Anstatt jeden Container einzeln zu starten und zu konfigurieren, können alle relevanten Informationen in einer zentralen Datei speichern und die Container dann mit einem einzigen Befehl starten | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Wie das Netzwerk im Docker Desktop aussieht | Ein Bild, das Screenshot, Text, Reihe enthält.  Automatisch generierte Beschreibung |

## Continuous Integration ½

|  |  |
| --- | --- |
| Repo URL (Fork) | <https://github.com/nolevit1/DevOpsDemo> |
| Jenkins Container starten |  |
| Element angelegt und devopsprojekt1 genannt | Ein Bild, das Screenshot, Text, Software enthält.  Automatisch generierte Beschreibung |
| Github Repo für die Tests angeben | Ein Bild, das Text, Screenshot, Reihe, Schrift enthält.  Automatisch generierte Beschreibung |
| Ansicht der Files die auch im Repo verfügbar sind | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Gradle build erfolgreich | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Testergebnisse | Ein Bild, das Text, Screenshot, Reihe enthält.  Automatisch generierte Beschreibung |
| Dashboard | Ein Bild, das Screenshot, Text, Software, Zahl enthält.  Automatisch generierte Beschreibung |
| Anpassung der Tests mit neuen Tests die Commited wurden | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Dashboard passt sich an (siehe Graphen) | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Nach dem committed zusätzlicher Tests | Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.  Automatisch generierte Beschreibung |
| Dashboard | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Testergebnisse | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |

## Continuous Integration 2/2

|  |  |
| --- | --- |
| Repo URLs | <https://github.com/nolevit1/DevOpsDemo> |
| Überprüfung ob NodeJS Plugin installiert ist | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Node JS Version angeben für die Installation | Ein Bild, das Text, Screenshot, Zahl, Software enthält.  Automatisch generierte Beschreibung |
| NPM installation | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Build erfolgreich | Ein Bild, das Text, Screenshot, Schrift, Algebra enthält.  Automatisch generierte Beschreibung |
| Überprüfung von NPM | NPM Error  Ein Bild, das Text, Schrift, Screenshot, weiß enthält.  Automatisch generierte Beschreibung |
| Fehlersuche | Ein Bild, das Text, Quittung, Screenshot, Reihe enthält.  Automatisch generierte Beschreibung |
| NPM Fehlersuche | Ein Bild, das Text, Dokument, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Funktioniert wieder | Ein Bild, das Screenshot, Text, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Pipeline erstellen mit dem Namen Devopspipeline | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Konfiguration | Ein Bild, das Text, Screenshot, Zahl, Reihe enthält.  Automatisch generierte Beschreibung |
| Konfiguration | checkout scmGit(branches: [[name: '\*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/nolevit1/DevOpsDemo']]) |
| Konfiguration |  |
| Jetzt bauen Status | Ein Bild, das Text, Software, Webseite, Website enthält.  Automatisch generierte Beschreibung |
| Workspace kontrollieren | Ein Bild, das Text, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Jenkinsfile erstellt (ohne Dateiendung) und ins Git Repo gepusht | Ein Bild, das Screenshot, Text, Multimedia-Software, Software enthält.  Automatisch generierte Beschreibung |
| Konfiguration Pipeline anpassen | Ein Bild, das Text, Screenshot, Zahl, parallel enthält.  Automatisch generierte Beschreibung |
| Jetzt bauen Status | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Jenkins Cloud Agent erstellen | Ein Bild, das Text, Reihe, Screenshot enthält.  Automatisch generierte Beschreibung |
| Socat für die Kommunikation pullen | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Socat Konfigurieren |  |
| IP ausfindig machen | Docker inspect socat2  IP: 172.17.0.3 |
| IP für Jenkins Cloud einrichten | Ein Bild, das Text, Quittung, Screenshot, Reihe enthält.  Automatisch generierte Beschreibung |
| Konfiguration | Ein Bild, das Text, Reihe, Zahl, Screenshot enthält.  Automatisch generierte Beschreibung |
| Konfiguration | Ein Bild, das Text, Reihe, Screenshot, Quittung enthält.  Automatisch generierte Beschreibung |
| Läuft | Ein Bild, das Text, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Erfolgreich | Ein Bild, das Text, Quittung, Schrift, weiß enthält.  Automatisch generierte Beschreibung |
| Docker |  |

## Docker Deployment

|  |  |
| --- | --- |
| Repo URLs | <https://github.com/nolevit1/DevOpsDocker-nolevit1> |
| Repo forken und lokal pullen | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Docker Build |  |
| Docker run | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Jenkins Container starten und neues Element erstellen | Ein Bild, das Text, Screenshot, Schrift, Webseite enthält.  Automatisch generierte Beschreibung |
| Beim Source Code Management mein Git Repo angeben | Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte Beschreibung |
| Build erfolgreich | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Workspace kontrollieren | Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.  Automatisch generierte Beschreibung |
| NodeJS Konfigurieren | Ein Bild, das Text, Screenshot, Quittung, Reihe enthält.  Automatisch generierte Beschreibung |
| Erfolgreich | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Docker einrichten | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Docker Image Konfiguration | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Build erfolgreich | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Build Konfiguration |  |
| Container Konfiguration - Remove | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Status | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Docker Konfiguration - Create | Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.  Automatisch generierte Beschreibung |
| Fehler behoben, IP geändert und erfolgreich getestet | Über nacht hat sich die IP Adresse geändert    Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Neuer erfolgreicher Status | Ein Bild, das Text, Screenshot, Schrift, weiß enthält.  Automatisch generierte Beschreibung |
| Docker Image | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Docker Container |  |
| Applikation test analog zum Git Repo Code | Ein Bild, das Text, Screenshot, Schrift, weiß enthält.  Automatisch generierte Beschreibung |
| Anpassungen gepusht und im Jenkins neu ausgeführt | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Resultat mit neuem Smile am Ende | Ein Bild, das Text, Screenshot, Schrift, weiß enthält.  Automatisch generierte Beschreibung |

## Cloud Deployment

|  |  |
| --- | --- |
| Repo URLs | <https://github.com/nolevit1/DevOpsAzure-nole> |
| Docker Hub URLs | <https://hub.docker.com/repository/docker/nxle/devopsazure/general> |
| Login Azure über VS Code | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Student Subscription als Standard setzten | Ein Bild, das Text, Screenshot, Schrift, Software enthält.  Automatisch generierte Beschreibung |
|  | Ein Bild, das Text, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Ressourcengruppen erstellen | Ein Bild, das Text, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Kontrolle | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
|  | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Web-App Funktioniert über Azure | Ein Bild, das Text, Screenshot, Schrift, weiß enthält.  Automatisch generierte Beschreibung |
| Web-App in der Azure übersicht | Ein Bild, das Screenshot, Text, Webseite enthält.  Automatisch generierte Beschreibung |
| Ressourcengruppe löschen | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Image bauen mit Docker Build | Ein Bild, das Text, Software, Multimedia-Software, Screenshot enthält.  Automatisch generierte Beschreibung |
| Docker run |  |
| Docker Desktop |  |
| Lokale Webapp funktioniert | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Docker Hub Repo |  |
| Image ins Docker Hub pushen. Komischerweise trotz guter Internetverbindung … könnte zwar immer besser sein aber in diesem Fall ausreichend ☺ | Drei Versuche docker zu pushen  Ein Bild, das Text, Screenshot enthält.  Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Design enthält.  Automatisch generierte Beschreibung |
| Docker Hub Image |  |
| Neue Ressourcengruppe | Ein Bild, das Text, Schrift, Screenshot enthält.  Automatisch generierte Beschreibung |
| Neuer Service Plan |  |
| Kontrolle im Azure Dashboard | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| Neue Web-app | Ein Bild, das Text, Screenshot, Software enthält.  Automatisch generierte Beschreibung |
| Azure Webapp Funktioniert | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| YML anpassungen für die Github Actions | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Github Einstellungen für die Sicherheit der Token und Profile | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Anpassung im Code und commit und push | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neues Image | Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Erfolgreiche Anpassungen | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |
| Ansicht im Github Actions | Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Ansicht in Azure | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Neuer Commit | Ein Bild, das Text, Software, Website, Webseite enthält.  Automatisch generierte Beschreibung |
| Erfolgreich | Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.  Automatisch generierte Beschreibung |

## DevopsDemo als Azure Web-App

|  |  |
| --- | --- |
| DevopsDemo als Azure Web-App | <https://github.com/nolevit1/DevOpsDemo> |
| Build erstellen | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Testen im Docker Desktop | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Funktioniert immernoch ☺ | Ein Bild, das Text, Screenshot, Schrift, Design enthält.  Automatisch generierte Beschreibung |
| Auf Docker Hub pushen | Ein Bild, das Text, Screenshot, Software enthält.  Automatisch generierte Beschreibung |
| Docker Hub Repo und Image erfolgreich erstellt | Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.  Automatisch generierte Beschreibung |
| Gleiches Szenario mit Azure Ressourcengruppe | Ein Bild, das Text, Screenshot, Schrift, Karte Menü enthält.  Automatisch generierte Beschreibung |
| Neue Azure Web-app | Ein Bild, das Text, Software, Multimedia-Software, Schrift enthält.  Automatisch generierte Beschreibung |
| YML File anpassungen mit passenden Docker hub Usernamen | Ein Bild, das Text, Screenshot, Software enthält.  Automatisch generierte Beschreibung |
| Neuer Token für Github Actions | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Github Secret Keys | Ein Bild, das Text, Screenshot, Zahl, Software enthält.  Automatisch generierte Beschreibung |
| Commiten und Pushen | Ein Bild, das Text, Software, Webseite, Website enthält.  Automatisch generierte Beschreibung |
| Devopsdemo funktioniert | Ein Bild, das Text, Screenshot, Schrift, Design enthält.  Automatisch generierte Beschreibung |
| Kontrolle vom neuen Image | Ein Bild, das Text, Screenshot, Schrift enthält.  Automatisch generierte Beschreibung |
| App funktioniert. Zusätzlich wurde azure programmer noch hinzugefügt über die Oberfläche (nicht im Code) | Ein Bild, das Text, Software, Zahl, Screenshot enthält.  Automatisch generierte Beschreibung |
| Überschreibe Frank mit Franco als Hardware Tester. | Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.  Automatisch generierte Beschreibung |
| Neue Änderungen commit und pushen | Ein Bild, das Text, Screenshot, Software, Schrift enthält.  Automatisch generierte Beschreibung |
| Github Actions | Ein Bild, das Text, Screenshot enthält.  Automatisch generierte Beschreibung |
| Erfolgreich | Ein Bild, das Text, Screenshot, Software enthält.  Automatisch generierte Beschreibung |
| Frontend Anpassung | Ein Bild, das Text, Screenshot, Software, Webseite enthält.  Automatisch generierte Beschreibung |
| Änderung nachvollziehen | Ein Bild, das Text, Software, Webseite, Computersymbol enthält.  Automatisch generierte Beschreibung |
| Änderung in der Webapp übernommen nach sehr langem warten | Ein Bild, das Text, Screenshot, Schrift, Design enthält.  Automatisch generierte Beschreibung |

# Schlussteil

## Reflexion, Zusammenfassung

Im Laufe des Frühlingssemesters 2024 wurden im DevOps-Kurs zahlreiche Übungen durchgeführt, die einen tiefen Einblick in die DevOps-Methodik gewährten. Dabei waren einige Übungen aufeinander abgestimmt, während andere unabhängig voneinander stattfanden.

Zu Beginn des Semesters stand die Einrichtung der benötigten Software und Entwicklungsumgebungen im Vordergrund. Es wurden GitHub-Konten erstellt, Git-Clients konfiguriert, und Tools wie Java, Gradle, Visual Studio Code, sowie verschiedene Erweiterungen installiert. Zusätzlich wurden npm, Selenium IDE, Docker Desktop, Jenkins, SonarQube und Azure-Services eingerichtet. Ziel war es, eine solide technische Grundlage für die folgenden Übungen zu schaffen. Einige Mac-Nutzer stiessen auf anfängliche Schwierigkeiten, die durch die Unterstützung von Adrian und dem Internet behoben wurden.

Versionskontrolle mit Git und GitHub war ein zentrales Thema. Obwohl ich bereits Vorkenntnisse hatte, gab es Neues zu entdecken, wie beispielsweise Github Actions. Wir lernten, wie man Repositories erstellt, verwaltet, klont sowie Branches und Pull Requests nutzt. Diese Fertigkeiten bildeten die Grundlage für zahlreiche Übungen, da wir nahezu wöchentlich mit Git arbeiteten. In den folgenden Wochen wurde das Augenmerk auf die Build-Tools Gradle und npm gelegt, mit denen wir erste Builds erstellten und eigene Beispiele integrierten. Das erworbene Wissen wurde in der darauffolgenden Woche in einem Projekt angewendet, in dem wir eine vorgegebene Applikation mit Front- und Backend erweitern durften.

Beim Unit Testing entwarfen wir Testcodes für Apps, wobei ich mich für das ScoreKeeper-Projekt entschied. Die testgetriebene Entwicklung fand ich besonders reizvoll, da sie eine neue Programmiermethode für mich darstellte. Trotzdem blieb ich skeptisch bezüglich der Realitätstauglichkeit, da ich es aus dem Berufsleben anders gewohnt war. Wir nutzten JUnit für das Schreiben und Integrieren von Unit Tests in unsere Projekte.

In der achten Woche konzentrierten wir uns auf die Überwachung und Verbesserung der Code-Qualität. Wir setzten SonarQube ein, um den Code zu analysieren und integrierten unser Front- und Backend in das Tool, um die Code-Qualität zu überwachen. Ausserdem wurden Integrationstests mit Selenium IDE durchgeführt, für die wir eigene Testfälle entwickelten und diese mit Kommilitonen teilten.

Die neunte Woche stand im Zeichen der Containerisierung. Wir lernten den Umgang mit Docker zur Bereitstellung von Anwendungen, die miteinander kommunizieren konnten. Ich konfigurierte MySQL und Adminer als separate Docker-Container und vernetzte sie.

In den Wochen zehn und elf erarbeiteten wir uns Kenntnisse über Continuous Integration (CI) mit Jenkins, lernten das Erstellen und Konfigurieren von Jenkins-Pipelines und setzten JUnit sowie Jacoco zur Messung von Tests und Code-Coverage ein. Dabei nutzten wir erneut unser DevOpsDemo-Repository.

Eines der Highlights war das Docker-Deployment in der zwölften Woche. Wir lernten, Docker-Container mittels Dockerfile zu erstellen und zu verwalten und automatisierten in Jenkins Build- und Deploy-Jobs, die bei jedem neuen Code-Push ausgeführt wurden. Das Semester endete mit Cloud Deployment, wobei Anwendungen in der Azure-Cloud bereitgestellt wurden. Dies beinhaltete die Nutzung von Docker und Azure-Services sowie die Konfiguration von Git-Repositories mit Dockerfiles für automatisierte Workflows.

Das Semester bot eine intensive Einführung in die Welt der DevOps, wobei die praktischen Übungen nicht nur theoretisches Wissen vermittelten, sondern auch unmittelbare Anwendungen ermöglichten, die die Effizienz und Zuverlässigkeit in der Softwareentwicklung deutlich verbesserten.

## Feedback zur Vorlesung (optional)

Das Modul DevOps war insgesamt sehr spannend und lehrreich, insbesondere die praktischen Übungen haben zum Verständnis der Entwicklungs- und Operations-Prozesse beigetragen. Es wäre jedoch bereichernd gewesen, wenn das Modul eine durchgängige Projektarbeit von Anfang bis Ende für jeden Teilnehmer vorgesehen hätte. Dadurch könnte man den gesamten Entwicklungszyklus eigenständig durchlaufen und praktische Erfahrungen sammeln.

Ein weiterer Punkt, der das Lernerlebnis verbessern würde, ist die Nutzung von GitHub Codespaces. Dies könnte potenzielle Hardware-Probleme vermeiden, die auftreten können, wenn Studierende unterschiedliche Systemkonfigurationen verwenden. Solche Schwierigkeiten haben zeitweise den Lernprozess verlangsamt und waren aufwändig zu beheben. Die Integration von GitHub Codespaces würde einen einheitlicheren Arbeitsbereich bieten, der die Durchführung der Projekte und Übungen erleichtert.

Insgesamt war das Modul jedoch sehr informativ und die praktischen Übungen waren sehr hilfreich, um die Theorie mit realen Anwendungsfällen zu verknüpfen. Adrian hat dies auch sehr gut vermittelt und das Know-How spürt man. Ich hoffe, dass zukünftige Iterationen des Kurses vielleicht eine kontinuierliche Projektarbeit integrieren können, um das Lernen noch weiter zu vertiefen.