GitLab-Training

Daily? Mo. – Do.: 10:00 – 10:30 Uhr? PUngewiß stellt Termin ein.

Wie viele Stunden/Tag?

* MFischer: 5-6h/Tag
* CKeutner: 4-6h/Tag
* PUngewiß: tbd.

Abwesenheiten

* MFischer: 20. – 24.05.2024
* CKeutner: 29.05., 31.05. und 03.06.
* PUngewiß: 17.05. und 22.05. SwissLife-Schulung, 10.06., 11.06. und 21.06.2024

Pptx/revealjs?

* Revealjs: <https://gitlab.ads.anderscore.com/presentations/anderscore-revealjs>

Beispielprojekt

* <https://gitlab.ads.anderscore.com/trainings/wicket-training>

**Schulungskonzept für einen Kurs "Gitlab: Git-Workflows & GitOps - CI/CD mit Docker und Git"**  
  
Dieser GitLab Kurs bietet den erweiterten Einstieg in Git-Workflows und die Hosting-Plattform GitLab. GitLab kann als freie DevOps-Plattform mehr, als nur Code für die Versionskontrolle zu hosten.  
  
Er baut Basiswissen über die wichtigsten Aspekte der Testautomatisierung auf und wenden dieses in praktischen Übungen und Beispielen an.  
  
Es erfolgt die Erarbeitung eines Git-Workflows und Erstellung unterschiedlicher CI/CD Pipelines für einfache bis komplexe Software Projekt/Szenarien.  
  
• In Ihrem Hause: Hamburg und Kiel--> Hamburg only wurde später entschieden  
• Dauer in Tagen: 3  
• Anzahl Teilnehmer pro Kurs: 12  
  
**1.) Themenübersicht/Agenda**  
  
**Tag 1: Einführung und Git-Workflow**  
• Begrüßung und Kursüberblick  
• Grundlagen von Git und GitLab  
• Git-Workflow im Team  
• Git Rebase und Merge-Strategien  
• Übungen: Git-Befehle, Branching und Merging  
  
**Tag 2: GitFlow, CI/CD, und GitOps**  
• Vertiefung Gitflow-Workflow  
• Tags, Releases und deren Verwaltung  
• Einführung in GitLab CI/CD und gitlab.yml  
• Praktische Übungen: Erstellen von CI/CD Pipelines  
• GitOps Grundlagen  
  
**Tag 3: Docker, GitLab CI, und Deployment**  
• Docker in der lokalen Entwicklung und mit GitLab CI  
• GitLab-Runner und Docker-Registry von GitLab  
• Erstellen von Release- und Tagged-Images  
• Deployment-Strategien und Umgebungsmanagement  
• IaC (Infrastructure as Code) und Abschlussdiskussion  
  
**2.) Zeitablauf des Unterrichtes**  
  
• Täglich 8 Stunden Unterricht, unterteilt in:  
• Vormittagsblock (inkl. 15 Min. Pause)  
• Mittagspause  
• Nachmittagsblock (inkl. 15 Min. Pause)  
• Zeiteinteilung pro Thema basiert auf Komplexität und Übungsbedarf.  
  
**3.) Inhalt/Beschreibung**  
  
**Tag 1: Einführung in Git und GitLab, Git-Workflow im Team CKeutner**  
  
*Vormittag: PUngewiß*  
1. Einführung und Kursüberblick  
• Zielsetzung des Kurses  
• Vorstellungsrunde und Erwartungsabfrage  
  
2. Grundlagen von Git und GitLab **CKeutner**  
• Geschichte und Konzepte von Git  
• Überblick über GitLab und seine Funktionen  
  
3. Git-Workflow im Team **CKeutner**  
• Branching-Modelle und deren Einsatz  
• Commit-Strategien und Best Practices  
  
*Nachmittag:* **CKeutner**  
4. Git Rebase und Merge-Strategien  
• Unterschiede zwischen Merge und Rebase  
• Konfliktlösung und Best Practices  
  
5. Praktische Übungen **CKeutner**  
• Einrichten einer GitLab-Projektumgebung  
• Anwenden von Git-Befehlen in Teamszenarien  
  
**Tag 2: Vertiefung Git-Workflow, CI/CD & GitOps**  
  
*Vormittag:*  
1. Gitflow-Workflow CKeutner  
• Einrichtung und Verwendung von Gitflow  
• Vorteile gegenüber anderen Workflows  
  
2. Tags, Releases und deren Verwaltung CKeutner  
• Erstellen und Verwalten von Tags und Releases  
• Einsatz von Tags für Versionskontrolle  
  
3. Einführung in GitLab CI/CD und gitlab.yml CKeutner  
• Grundkonzepte von Continuous Integration und Continuous Deployment  
• Erstellung und Verwaltung von .gitlab-ci.yml Dateien  
  
*Nachmittag:* CKeutner  
4. Praktische Übungen zu CI/CD  
• Erstellen von einfachen CI/CD Pipelines  
• Automatisierung von Tests und Deployments  
  
5. Grundlagen von GitOps MFischer  
• Definition und Vorteile von GitOps  
• Einsatzbereiche und Best Practices  
  
**Tag 3: Docker in der Entwicklung, GitLab CI, und Deployment-Strategien MFischer**  
  
*Vormittag:*  
1. Lokale Entwicklung mit Docker **MFischer**  
• Grundlagen von Docker und dessen Integration in Entwicklungsprozesse  
• Nutzung von Docker in Kombination mit GitLab CI  
  
2. GitLab-Runner und Docker-Registry **MFischer**  
• Konfiguration und Einsatz von GitLab-Runnern  
• Verwendung der GitLab Docker-Registry  
  
*Nachmittag:*  
3. Erstellen von Release- und Tagged-Images **MFischer**  
• Automatisierung der Erstellung von Docker-Images mit GitLab CI  
• Verwaltung und Nutzung von Images für verschiedene Umgebungen  
  
4. Möglichkeiten des Deployments und Verwaltung von Konfigurationen **MFischer**  
• Übersicht über Deployment-Strategien (z.B. Blue-Green-Deployment)  
• Verwaltung von Umgebungsvariablen und Secrets  
  
5. Abschlussübung und Diskussion **MFischer**  
• Implementierung einer kompletten CI/CD-Pipeline inklusive Deployment  
• Q&A, Feedbackrunde und Kursabschluss  
  
**4.) Methoden**  
  
• Theoretische Einführungen für Konzeptverständnis.  
• Praktische Übungen für Hands-on Erfahrung.  
• Gruppenarbeit zur Förderung der Teamarbeit und Diskussion.  
• Diskussionen und Q&A-Sessions für individuelle Fragen und Vertiefungen.  
  
**5.) Schulungsmaterial**  
  
• Präsentationen zur Vermittlung der Theorie.  
• Übungsdateien und Beispiele auf GitLab bereitgestellt.  
• Handouts und Cheatsheets als Nachschlagewerk für Teilnehmer.  
  
**6.) Dozenteneinsatz**  
  
• Erfahrene Dozenten aus der Praxis mit tiefem Verständnis von Git, GitLab, CI/CD und Docker.  
• Interaktive Lehrmethoden zur Förderung des Lernengagements.  
• Bereitstellung von individuellem Feedback und Unterstützung während der Übungen.

Bei der Entwicklung und Umsetzung der Präsentation kann ich (=Peter Hecker) unterstützen.