



Training Fact Sheet
Continuous Delivery und
Infrastructure as Code

08.11.2015 17:31

training-cd-iac.odt version 5

Christoph Lukas, email lukas@pingworks.de, tel +49 7531 366964

Beschreibung

Das Schreiben von Software Tests ist mittlerweile (fast) überall integraler Bestandteil von Softwareentwicklung. Trotzdem kämpfen viele Firmen noch mit Qualitätsmängeln kurz vor den Releases, viel zu langen Releasezyklen und ewig knappen Testumgebungen. Continuous Delivery und Infrastructure as Code helfen, diese Herausforderungen zu adressieren. Allerdings erscheinen die anstehenden Aufgaben so kompliziert, dass es den Beteiligten schwer fällt, konkrete erste Schritte abzuleiten.

Im vorliegenden Training lernen die Teilnehmer die Konzepte und Prinzipien von Continuous Delivery und Infrastructure as Code in vielen 'Hands-On' Sessions kennen und verstehen.

Am Ende des zweitägigen Trainings sind die Teilnehmer in der Lage, diese auf das eigene Unternehmen, die eigene Software und das dort genutzte Toolset zu übertragen und so konkrete erste Schritte für den eigenen Kontext zu planen.

Inhalte

- Konzepte und Prinzipien von Continuous Delivery.
 - Aufbau einer Continuous Delivery Pipeline in Jenkins anhand einer Beispielanwendung.
 - Erstellung einer Commit-Stage mit Build, Unittesting mit Rspec und Paketierung.
 - Integration eines Binärartefakt-Repositories.
 - Kontinuierliche Builds mit jeder Änderung an der Software.
 - Implementation einer Acceptance-Test-Stage zur Ausführung von Akzeptanztests.
 - Automatisierung des Deployments.
 - Deployment der Anwendung mit dem gleichen Tooling in der Pipeline und in Produktion.
 - Aggregation und Visualisierung von Testergebnissen mit Dashboards.
-
- Konzepte und Prinzipien von Infrastructure as Code.
 - Beschreibung von Infrastruktur mit Hilfe von Chef.
 - Lokale Infrastrukturentwicklung mit Virtualbox.
 - Test driven development mit Testkitchen und Serverspec Tests.
 - Die passende Virtualisierung (VMs vs. Docker).
 - Anwendung von Chef Cookbooks auf Umgebungen ohne Chef-Server.
 - Pipelines für Infrastruktur Code.
 - Beschreibung von komplexen Umgebungen mit dem Environment Cookbook Pattern.
 - Testumgebungen auf Knopfdruck mit einer Openstack Cloud.

Trainer

Alexander Birk und Christoph Lukas beschäftigen sich seit 2010 intensiv mit dem Thema Continuous Delivery und Infrastructure as Code.

In langfristigen Projekten haben sie in dieser Zeit Unternehmen wie die Hybris AG, ein nationales schweizerisches Medienunternehmen, die Allianz AG/ Allsecur AG oder die Deutsche Telekom AG in der konkreten Umsetzung von Continuous Delivery unterstützt. Zusätzlich waren sie in dieser Zeit für Unternehmen wie EIZO, Pro7/Sat1 oder ein nationales schweizerisches Transportunternehmen beratend tätig.

Zielgruppe

Das Training wendet sich an Mitarbeiter aus Unternehmen, die gerne ihren Software Release Prozess verbessern möchten. Konkret sind die Inhalte auf Entwickler, Mitarbeiter aus dem Betrieb und Team- und Abteilungsleiter zugeschnitten. Für ein inhouse Training kann der genaue inhaltliche Schwerpunkt abgestimmt werden.

Ziel

Die Teilnehmer erlernen handlungsorientiert die Konzepte und Grundsätze von Continuous Delivery und Infrastructure as Code und können diese im Anschluss auf die Situation im eigenen Unternehmen anwenden.

Methode

Die Erarbeitung der Inhalte erfolgt im Charakter eines zweitägigen Workshops anhand einer durchgängigen Beispielanwendung.

Jedes Thema startet mit einer kurzen Darstellung der Grundlagen, im Anschluss werden diese in Kleingruppen praktisch umgesetzt und diskutiert. Abgerundet wird jeder Themenblock mit einem Ausblick auf die nächsten Schritte.