

# DevOps im Oracle DB Umfeld

**Florian Feicht**  
**Trivadis GmbH**  
**München**

## Schlüsselworte

DevOps Operations Infrastruktur DBA Admin OS Lean Kanban

## Einleitung

*DevOps ist zurzeit in aller Munde. Das Thema wird vor allem im Development Bereich sehr intensiv diskutiert. Dabei ergeben sich auch für Operations sehr viele Chancen und Herausforderungen. Es gibt aber auch einige Fragen, die beantwortet werden müssen. Muss sich die Infrastruktur ändern? Oder muss sogar eine komplette neue Infrastruktur implementiert werden? Welche Prozesse müssen wie angepasst werden? Sind neue Tools notwendig?*

Ganz entgegen der weitläufigen Meinung handelt es sich bei DevOps nicht um eine Ansammlung von Tools. Es geht auch nicht um den einen Ansatz oder die eine Vorgehensweise. Ebenso wenig geht es darum, dass die Entwickler alle Betriebsaufgaben übernehmen. Vielmehr geht es um die Implementierung einer neuen Kultur der Zusammenarbeit zwischen Development und Betrieb (Operations). Dazu werden verschiedene Tools und Methoden eingesetzt. Es gilt Engpässe zu eliminieren und die Durchlaufzeiten der einzelnen Tasks soweit als möglich zu minimieren. Dabei hilft unter anderem eine durchgängige Automatisierung aller Aufgaben. Wir werden im späteren Verlauf detaillierter darauf eingehen.

Zuerst konzentrieren wir uns auf die Infrastruktur. Wie stellt man fest ob die schon vorhandene Infrastruktur für DevOps geeignet ist?

## Definition der Anforderungen

Dazu müssen wir zuerst klären, was DevOps genau bedeutet. Wenn man über DevOps spricht, dann stechen drei wichtige Prinzipien hervor (gemäß „The DevOps Handbook“ von John Willis et. al.):

- The Principle of Flow
- The Principle of Feedback
- The Principle of Continual Learning and Experimentation

Das heißt der Arbeitsfluss muss optimiert werden und eine gelebte konstruktive Kritik muss eingeführt werden. Dadurch wird Freiraum für Neues und Experimente geschaffen. Doch welchen Beitrag kann die Infrastruktur dazu liefern? Müssen die bekannten „state-of-the-art“ Kriterien für eine leistungsfähige Umgebung angepasst werden?

Bis jetzt wurden Infrastrukturen auf Basis folgender Kriterien ((Key-Performance-Indikatoren (KPIs)) bewertet:

- Stabilität
- Fehlertoleranz
- Performance
- Security
- Compliance
- Accounting

Wenn man diese Punkte nun gegenüber den drei Prinzipien stellt wird eines schnell klar: Die genannten KPIs sind auch im DevOps Umfeld weiterhin essentiell. So wird es beispielsweise ohne die nötige Stabilität sehr schwierig in einen optimierten Arbeitsprozess zu wechseln. Stattdessen werden weiterhin „Feuerwehreinsätze“ den täglichen Arbeitsalltag der Infrastruktur-Teams prägen. Diese ungeplanten Aufgaben verursachen verzögerte Projektdurchführungen und dadurch Unzufriedenheit bei den Fachbereichen. In der genaueren Betrachtung muss man aber konsequenterweise einigen der KPIs mehr Bedeutung als bisher schenken. Beispielsweise wird durch den Einsatz von Self-Services ein zentrales Logging- und Securitykonzept zwingend benötigt. Damit wird die Nachvollziehbarkeit der Änderungen und die Einhaltung der Security-Richtlinien gewährleistet. Nur so können die Vorteile einer DevOps Umgebung ausgeschöpft werden. An einigen anderen Stellen werden aber zwingend Änderungen notwendig, um eine erfolgreiche Einführung von DevOps gewährleisten zu können.

### **Infrastructure as Code**

Dazu zählt zum einen der wichtige Punkt „Infrastructure as Code“. Unter dem Begriff versteht man die Abbildung betrieblicher Abläufe in Code. Dabei stellt sich die Frage, ob jetzt nur noch Entwickler an der Infrastruktur arbeiten dürfen. Dies ist aber ganz und gar nicht der Fall. Weiterhin ist die Expertise der Administratoren das wichtigste Gut. Es werden aber einige Vorgehensweise und Tools aus der Development-Welt übernommen. Dazu zählt beispielsweise eine Versionskontrolle aller Skripte und Konfigurationsdateien. Damit kann jederzeit nachvollzogen werden, wer wann aus welchen Gründen welche Änderungen an der Infrastruktur beziehungsweise an einzelnen Konfigurationen vorgenommen hat. Außerdem sollten auch im Operations-Team Coding-Guidelines vereinbart werden. Dies erleichtert unter anderem die Wiederverwendbarkeit häufig benötigter Skript-Teile. Eine festgelegte Namenskonvention für die Skripte hilft außerdem bei der Suche nach bereits erstellten Material. All dies sind wichtige Voraussetzungen für den nächsten wichtigen Punkt.

### **Automatisierung**

Der nächste Punkt ist die Automatisierung der anfallenden Aufgaben. Dazu zählt beispielsweise die Provisionierung neuer Server, VMs oder Container. Auch Netzwerkgeräte können automatisch verwaltet werden. Ebenfalls zu den Kernpunkten der Automatisierung zählt die automatisierte Installation und Konfiguration der eingesetzten Softwarekomponenten. Fast genauso wichtig sind automatisierte Tests. Das ist im Entwicklungsbereich durchaus üblich, wird auf Operations-Seite aber oft vernachlässigt. Um

automatisch provisionieren zu können ist eine sehr gute Kenntnis der Umgebung notwendig. Am einfachsten lässt sich dies durch eine konsequente Standardisierung erreichen. Außerdem wird auf der einen Seite ein sehr ausgereiftes Kapazitätsmanagement benötigt, auf der anderen Seite ist aber auch ein zentrales Logging und Monitoring sehr wichtig. Und natürlich muss dies eine Überwachung der Security Vorgaben beinhalten.

### **Self-Services**

Um die Vorteile eines hohen Automatisierungsgrades komplett ausnutzen zu können werden Self-Services benötigt. Mithilfe dieser können Durchlaufzeiten von täglichen Betriebsaufgaben deutlich reduziert werden. Ein Beispiel hierfür ist das Anlegen eines Schemas in einer Datenbank. In klassischen IT-Abteilung durchläuft selbst ein technisch so einfach umzusetzendes Ticket mehrere Teams. So werden leicht Durchlaufzeiten von mehreren Tagen erreicht. Im DevOps Umfeld sollte dieses Vorgehen der Vergangenheit angehören. Ein einfacher Self-Service mit einer automatischen und standardisierten Provisionierung des Workspaces hilft hier dem Operations-Team und auch dem Entwickler. Oracle unterstützt die Administration bereits in vieler Hinsicht, aber auch eigene Tools sind hier relativ einfach zu implementieren. So ist es beispielsweise mit APEX sehr einfach möglich ein Self-Service Portal für eine automatisierte Schemaanlage zu erstellen.

### **DevOps vs. ITIL**

Ein weiterer wichtiger Punkt sind die eingesetzten Prozesse. Dabei fällt immer wieder die Frage ob die bereits eingesetzten ITIL Prozesse noch relevant sind oder ersetzt werden müssen. In einer agilen DevOps Umgebung soll natürlich der Prozessoverhead möglichst minimiert werden. Trotzdem sind die bereits vorhandenen Prozesse wichtig und werden weiterhin benötigt. Eine konsequente Weiterentwicklung dieser ist aber zwingend notwendig. So fällt beispielsweise dem Change Management eine noch größere Bedeutung als bisher zu. Um eine Infrastruktur stabil betreiben zu können ist es zwingend notwendig, alle durchgeführten Änderungen nachvollziehen zu können. Dieser Prozess sollte soweit wie möglich automatisiert werden. Die eingesetzten Automatisierungstools sollten beispielweise über APIs mit dem Changemanagement System kommunizieren können. Auch das Configuration Management und das Release Management werden noch wichtiger. Um in einer automatisierten Infrastruktur betriebsfähig zu bleiben muss jederzeit bekannt sein wo welche Softwarekomponente in welcher Version im Einsatz ist. Hier ist der Einsatz passender Tools eine Grundvoraussetzung um erfolgreich zu sein.

### **Security**

Auch der Bereich Security muss bei der Vorbereitung der Infrastruktur für DevOps in den Fokus gestellt werden. Eine gute und enge Zusammenarbeit mit der Security-Abteilung ist absolut notwendig. Nur mit einer Einbindung der Security-Abteilung bereits zum Start des Projektes können die Weichen richtiggestellt werden. Eine Implementierung von Security-Anforderungen im Nachhinein ist in der Regel nur mit Abstrichen möglich. Weiterhin muss hier der Punkt Monitoring stark mit einbezogen werden. Eine Überwachung der Security-Richtlinien, die Auswertung der Audit-Files und die daraus folgende Alarmierung sind sehr

wichtig um die immer größer werdenden Anforderungen an Security und Compliance gerecht zu werden.

## Einführung von DevOps

Nachdem die wichtigsten Eckpunkte für eine erfolgreiche DevOps Infrastruktur bekannt sind folgt die Planung der Umsetzung. Der Wunsch ist natürlich, dass es einen Schalter DevOps=True gibt. Leider ist das in gewachsenen IT-Landschaften nicht möglich. Bereits vorhandene Strukturen, Prozesse und Vorgehensweisen können nicht von einem auf den anderen Tag geändert werden. Trotzdem ist ein stimmiges Gesamtkonzept vor dem Start in die DevOps-Welt notwendig. Dazu gehören klare Vorstellungen an welchen Komponenten der Umgebung Änderungen durchgeführt werden müssen, um den neuen und erweiterten Anforderungen zu entsprechen. Anschließend ist die Auswahl der geeigneten Tools essentiell.

Die Umsetzung wird Schicht für Schicht erfolgen, damit die notwendige Agilität und Flexibilität gewährleistet wird. Nur so sind ein kontrollierter Rollout und gegebenenfalls eine Anpassung der Vorgehensweise überhaupt möglich. In Abbildung 1 ist eine mögliche Architektur mit den benötigten Komponenten dargestellt.

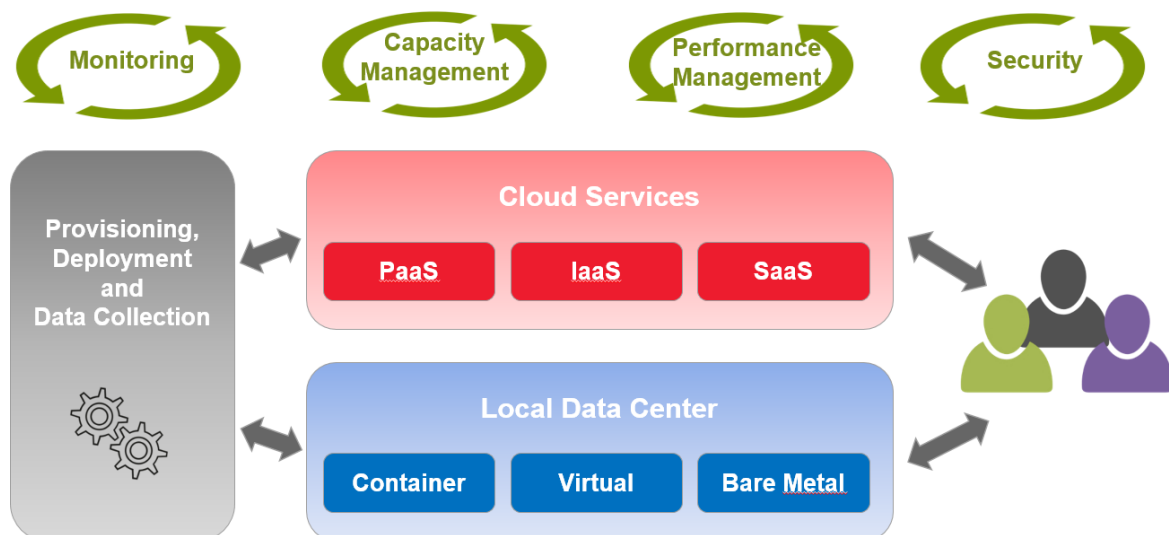


Abbildung 1: Architektur einer DevOps Infrastruktur

Falls der Rollout der für DevOps notwendigen Komponenten nicht von Anfang perfekt klappt, ist dies kein Problem. Grundsätzlich kann jedes IT-Projekt gerettet werden. Die Frage ist immer der Aufwand, sei es die Zeit oder das Geld, das investiert werden muss.

Im DevOps-Umfeld gilt dieser Grundsatz ebenfalls. Falls die Planungen zum Gesamtkonzept in die falsche Richtung laufen, fällt dies spätestens beim Rollout der einzelnen Schichten auf.

Hier ist aufgrund der hohen Flexibilität, die in solchen Projekten gefordert ist, eine Anpassung möglich. Das heißt notwendige Optimierungen am Konzept können auch hier noch durchgeführt werden. Selbst bei einem perfekten Start mit der optimalen Planung, kann es bei der Umsetzung Anpassungsbedarf geben.

**Kontaktadresse:**

Florian Feicht  
Trivadis GmbH  
Lehrer-Wirth-Str 4  
D-81829 München

Telefon: +49 (0) 89 99 27 59 3  
E-Mail [florian.feicht@trivadis.com](mailto:florian.feicht@trivadis.com)  
Internet: [www.trivadis.com](http://www.trivadis.com)