

CLOUD-INTEGRATION UND -MIGRATION



VON LIFT & SHIFT BIS REPLACE WITH SAAS

Die passende Strategie für eine erfolgreiche Cloud-Migration zu finden, kann herausfordernd sein. Verschiedene Ansätze sind erfolgsversprechend. Doch welcher Weg ist der richtige?

Sechs Migrationsszenarien haben sich etabliert, die mit der entsprechenden Konsequenz verfolgt werden müssen, um tatsächlich zu fruchten. Dabei reicht die Spanne von minimalinvasiven Szenarien wie Lift & Shift bis hin zu radikalen Ansätzen wie Rip & Replace oder Replace with SaaS.

Um das ideale Szenario zu identifizieren, hilft die möglichst präzise Beantwortung folgender Fragestellungen:

- 🛡️ Auf wie viel Erfahrung mit der Entwicklung in der Cloud können Sie zurückgreifen?
- 🔍 Wie nah sind Ihre eingesetzten Technologien und Architekturen an gängigen Cloud-Architekturen und -Technologien?
- ⚙️ Wie viel Zeit und Ressourcen können in die Cloud-Migration fließen?
- ☁️ Wollen Sie das volle Potenzial der Cloud direkt ausschöpfen?

Diese Fragen müssen für jede Anwendung, die in die Cloud migriert werden soll, individuell beantwortet werden. Sinnvoll ist es, Abhängigkeiten zu identifizieren sowie Ähnlichkeiten zu erkennen, um Cluster zu bilden. So können Synergien geschaffen werden und Teams können sich gemeinsam der Herausforderung stellen, die Anwendungen auf den Umzug in die Cloud vorzubereiten.

Nutzen Sie Ihre Antworten, um das jeweils passende Szenario herauszufinden.

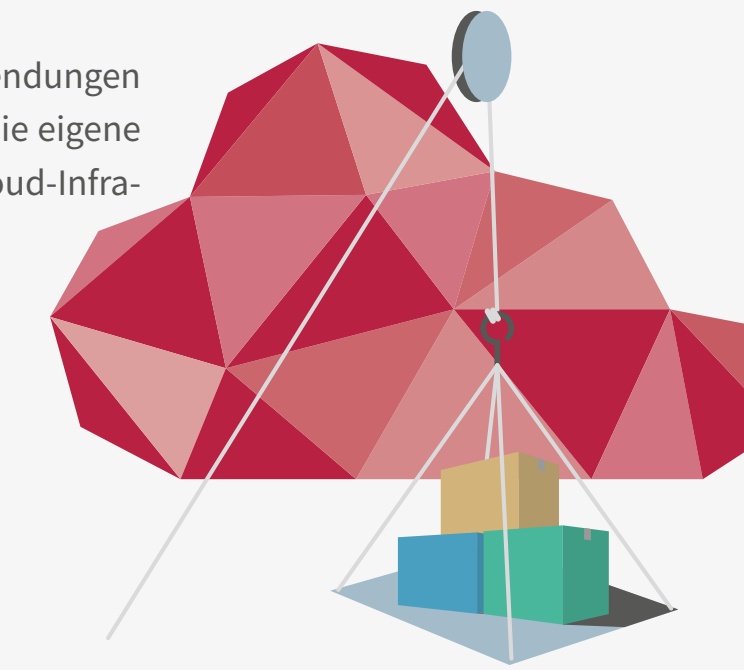
Setzen Sie auf unsere Erfahrung bei der Cloud-Migration! Sprechen Sie uns an!

DIE SECHS VORGEHENSMODELLE DER CLOUD-MIGRATION IM ÜBERBLICK

1 LIFT & SHIFT

Direktes Verschieben von Anwendungen in die Cloud ohne Modifikation. Die eigene Infrastruktur wird durch die Cloud-Infrastruktur abgelöst (IaaS).

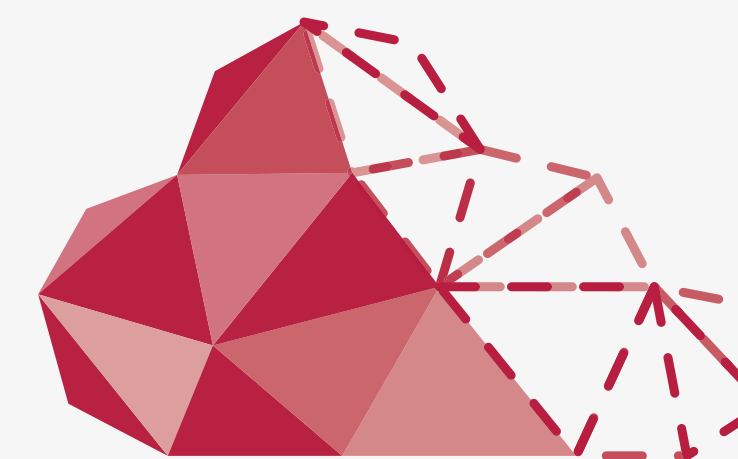
- + Keine Änderungen in Architektur und Code
- + Keine Kosten für Entwicklung und Test
- Native Vorteile der Cloud bleiben ungenutzt (z.B. Skalierbarkeit)



2 LIFT & RESHAPE

Verschieben von Anwendungen mit dem Ziel, die Anwendung direkt in der Cloud zu optimieren. Nutzung von Cloud-Standard-Komponenten, die direkt von der eigenen Software verwendet werden (z.B. Datenbank in der Cloud).

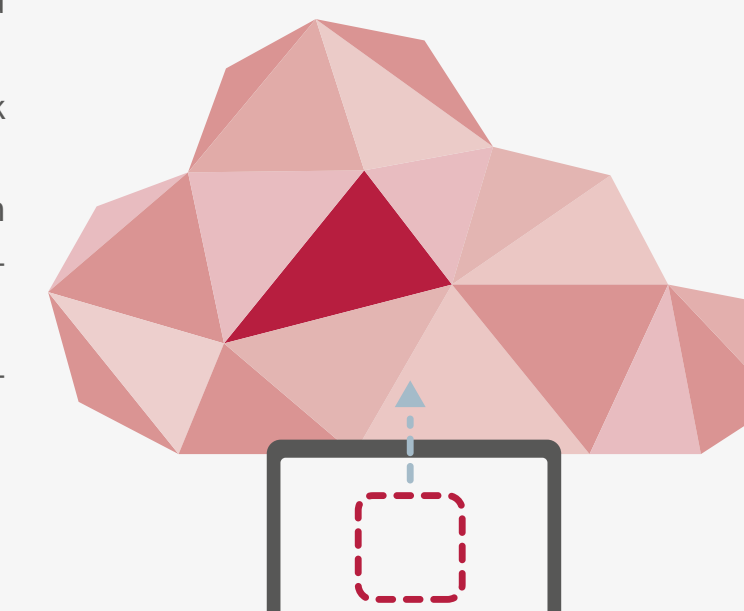
- + Kosteneffizient
- + Erste Nutzung cloud-nativer Funktionalität mit der Möglichkeit, später zu erweitern
- Eventuell Verzettelung mit Ausuferung zum Refactoring
- Viele Vorteile der Cloud bleiben ungenutzt
- ⚠️ Der Fokus liegt auf Umkonfiguration und nicht auf Neuentwicklung von bestehender Software.



3 DECOUPLE & REWRITE

Einzelne Teile werden für die Cloud modernisiert. Der übliche Weg ist die Umstellung auf Microservices mittels Containern. Damit geht die automatisierte Aktualisierung der Cloud-Inhalte einher. CI/CD sorgen für eine kontinuierliche Integration.

- + Nutzung von Cloud-Funktionen führt zu kompaktem, wartbarem Code
- + Software muss nicht abrupt in einem Stück ersetzt werden
- + Potentiale der Cloud können genutzt werden
- + Microservices führen moderne Architekturmuster ein
- + CI/CD sorgen für eine kontinuierliche Integration



- Zeitintensiver als Lift only
- Fachwissen notwendig
- Neuer Code muss getestet werden
- CI/CD-Pipeline muss implementiert werden

4 RE-ARCHITECT

Komplette Überarbeitung der Software, um alle Möglichkeiten der Cloud nativ nutzen zu können. Dabei müssen die Kosten im Blick behalten werden; unter Umständen empfiehlt sich direkt eine vollständige Neuentwicklung (RIP & REPLACE).

- + Nutzung von Cloud-Funktionen, die über den Scope einer einzelnen Komponente hinausgehen
- + Nutzung von Cloud-Funktionen führt zu kompaktem, wartbarem Code
- + Vorteile der Cloud nutzbar
- + Vorteile von Microservices nutzbar
- + CI/CD sorgen für eine kontinuierliche Integration

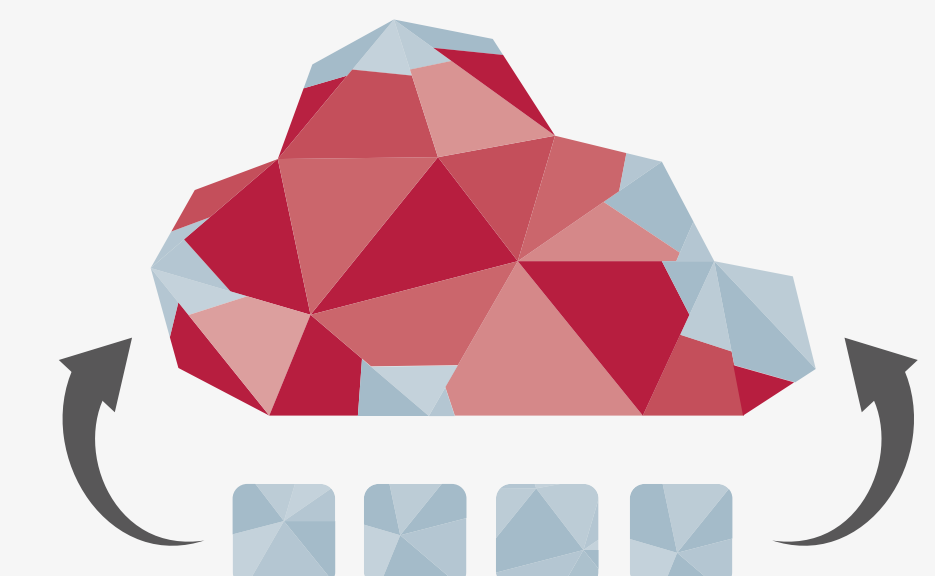


- Zeitintensiver als Decouple & Rewrite
- Fachwissen notwendig
- Neuer Code muss getestet werden
- CI/CD-Pipeline muss implementiert werden

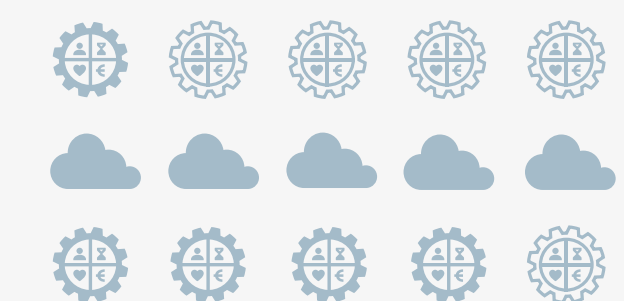


5a RIP & REPLACE

Verwerfen der Software und Neuentwicklung im Hinblick auf die Cloud.



- + Die Vorteile sind denen der Re-Architect-Methode ähnlich. Der Unterschied: Da die Software von Grund auf neu entwickelt wird, kann sie für alle Features der Cloud passend gestaltet werden.
- Die Nachteile sind ebenfalls identisch mit der Re-Architect-Methode. Ein weiterer Nachteil ist jedoch, dass die Software in einem Stück ersetzt werden muss.



5b REPLACE WITH SAAS

Verwerfen der Software und Nutzung der Software des Cloud-Providers (SaaS).



- + Geringster Aufwand
- Beschränkung auf Möglichkeiten des Cloud-Providers
- Gelingt nur mit Standardsoftware, nicht mit Eigenentwicklungen
- Unter Umständen wenig Anpassungsmöglichkeiten
- Vendor Lock-in