

Projektbeschreibung

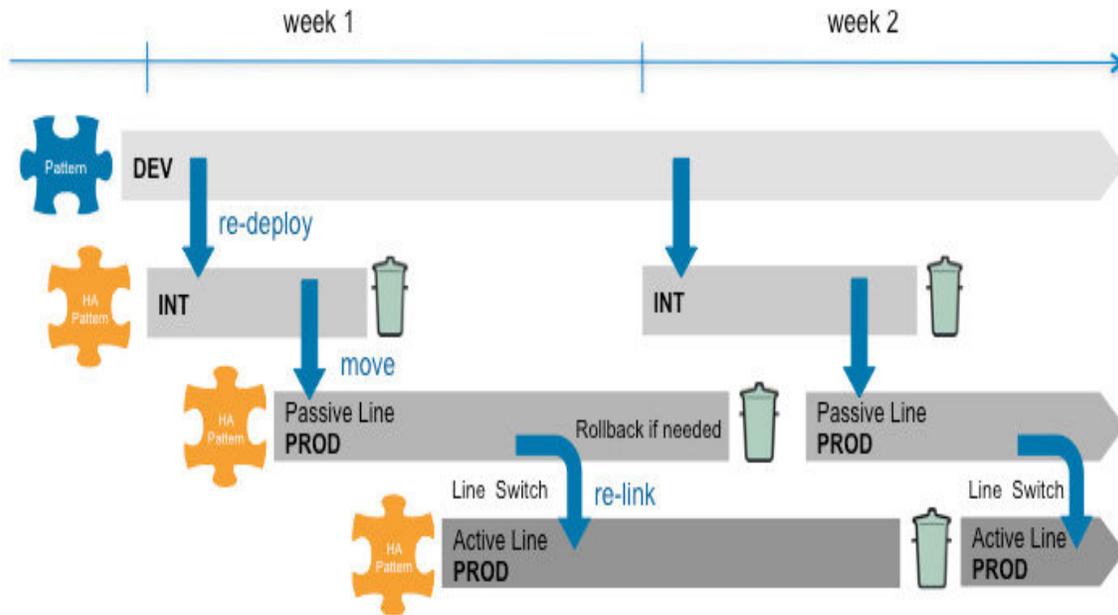
An dieser Stelle wollen wir die beschriebene Theorie durch eine kurze Zusammenfassung eines aktuellen Kundenpilotprojektes aus der Automobilindustrie belegen. Extrem aggressive Wachstumspläne auf globaler Basis sowie eine schnellere Time-to-Market Strategie im Zuge mobiler Dienstleistungen bedingen insbesondere schnellere Entwicklungs- und häufigere Releasezyklen (im Minimum wöchentlich) auf der Seite der mobilen Backend Services, um die vom Fachbereich geforderten Innovationssprünge umsetzen zu können.

Die erforderliche Geschwindigkeit insbesondere in der dynamischen Bereitstellung von erforderlichen Testumgebungen konnte durch den internen IT Provider nicht gewährleistet werden, so dass externe Cloud Services auf I/PaaS im Kontext einer Continuous Delivery Pipeline in Anspruch genommen werden. Die Wahl fällt auf die global verfügbare IBM SoftLayer Lösung, deren Data Center durch einen privaten Backbone mit hoher Bandbreite verbunden sind. Dies ist eine erforderliche Grundvoraussetzung, um u.a. HA und DR Konzepte für den Produktivbetrieb gewährleisten zu können.

Herausforderung des DevOps Konzeptes war die neben der üblichen organisatorischen Trennung von Anwendungsentwicklung und Betrieb die Splittung der beiden Bereiche über Unternehmensgrenzen hinweg. Während die Anwendungsentwicklung in der Verantwortung des Kunden verbleibt, liegt die Betriebsverantwortung in den Händen des Cloud Providers, der IBM.

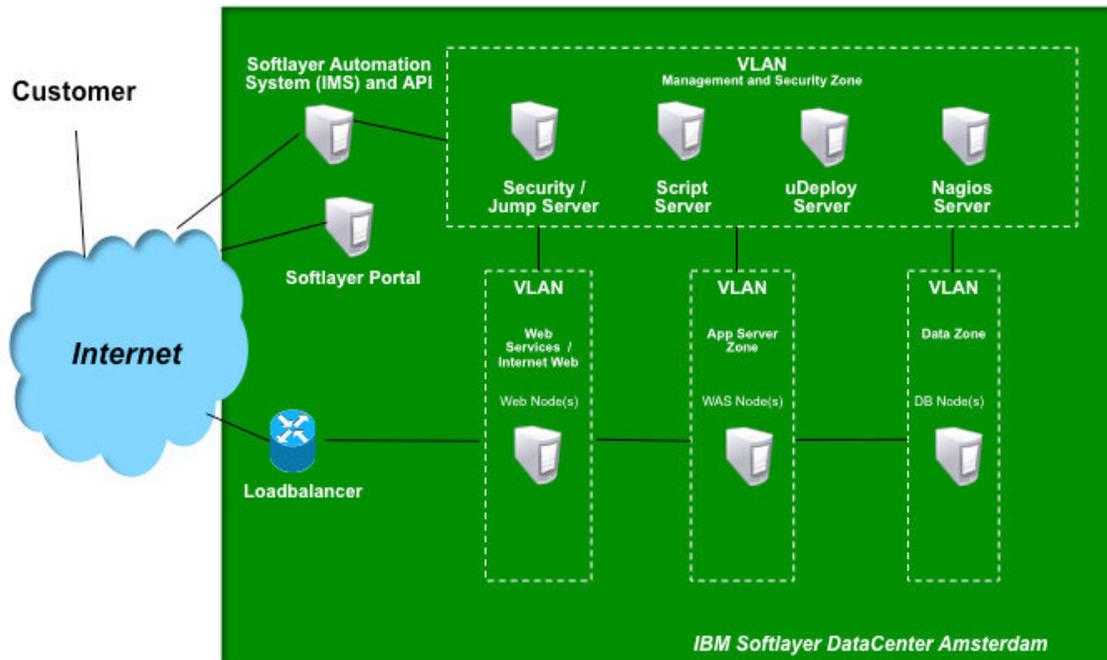
Eines der Grundprinzipien der Implementierung war das Handeln nach dem Prinzip "ein Change ist ein Change", unabhängig davon ob dieser in der AE oder im Betriebsumfeld begründet ist. Jede Änderung verursacht einen Durchlauf des Staging Prozesses (siehe nachfolgendes Diagramm) incl. einer vollständigen Provisionierung des Infrastrukturstacks. Wir bezeichnen dieses Vorgehen auch als „Full Stack Engineering oder Release Ansatz“. Mit anderen Worten, Systeme werden nicht mehr im laufenden Betrieb verändert, es wird das Modell verändert das den Input für die CD Pipeline bildet. Dies setzt selbstverständlich eine extrem schnelle Provisionierung und Deploymentverfahren voraus. Auf Ebene der Produktion wird ein Doppellinienkonzept etabliert. Eine aktive Produktionslinie und eine passive Produktionslinie bilden dieses rotierende Gerüst. In der passiven Linien, die mit jedem neuen zu testenden Release in der Cloud aufgebaut wird, erfolgen die vorbereitenden und finalen Abnahmetests für das Release N+1. Sobald diese Tests und entsprechende Freigabeverfahren erfolgt sind, erfolgt ein vom Deployment Werkzeug gesteuerter Line Switch, der zu einer Umkonfiguration der LoadBalancer in der Cloud führen. Die aktive Produktionslinie kann dabei entweder noch eine Zeit lang deaktiviert vorgehalten oder de-provisioniert werden

Produktionsnahe Umgebungen in hochverfügbarer redundanter Ausprägung konnten in weniger als 30 Minuten lauffähig incl. Der relevanten Anwendungsversion zur Verfügung gestellt werden.



Im obigen Diagramm ist die Rede von Cloud Patterns. Ein Pattern definiert und kapselt alle in der Cloud relevanten Aktivitäten (Serverbereitstellung, Netzwerkkonfiguration, Storageanbindung) um eine Plattform bereitzustellen, die aus Sicht der Anwendungsentwicklung bzw. Testabteilung notwendig ist, um Anwendungen in diese Plattform zu deployen. Cloud Patterns beschreiben ein Modell der instanziierten Infrastruktur und können auch als „Infrastructure as Code“ bezeichnet werden – und auch im Sinne von Versionierung und Code Verwaltung wie Code behandelt werden. Es wurden unterschiedliche Pattern für unterschiedliche Teststufen definiert. Während in frühen Teststufen der PaaS Stack auf einem bis wenigen Systemen angesiedelt ist, definiert das Produktionspattern eine redundant und hochverfügbar ausgelegte Infrastrukturplattform auf Basis einer 3-Tier Web Architektur.

Die folgende Zeichnung vermittelt einen stark vereinfachten Überblick über die Lösungsarchitektur:



Die gesamte Lösung wurde im IBM SoftLayer Data Center Amsterdam aufgebaut und dient als zentraler Management Hub für die Steuerung weltweiter Provisionierung und Anwendungsdeployment Aktivitäten in weiteren Cloud Standorten in USA und Asia Pacific.

Innerhalb eines definierten Management VLANs wurde IBM UrbanCode Deploy als zentrales Deployment Automationswerkzeug implementiert und steuert sowohl den gesamten Anwendungsdeploymentprozess als auch die Infrastrukturprovisionierung innerhalb der SoftLayer Cloud Rechenzentren. Die Integration der Plattformprovisionierung erfolgt über die Nutzung der SoftLayer API's, deren Nutzung gekapselt hinter einem Funktionsaufruf zur Erzeugung einer Development, Test- oder Produktionsumgebung liegt (s.o. Bezug zum Patternprinzip).

IBM UrbanCode Deploy erhält neue Anwendungsartefakte auf dem Continuous Integration / Build Management Prozess und speichert die Versionen der Anwendungsartefakte in einem dafür vorgesehenen Repository.

Zusätzlich wurden folgende weitere Management Systeme im Management VLAN implementiert:

- ein **Nagios Monitoring Server**, der provisionierte Systeme automatisiert in die Überwachung aufnimmt und wieder bei Rückgabe in den Cloud Pool aus selbiger entfernt
- ein **Script Server** auf dem die für das Erzeugen des Plattformstacks in der Cloud relevanten Scripte laufen. Die Scripte nutzen das SoftLayer API zur Provisionierung und Konfiguration der notwendigen Umgebung anhand des definierten Modells / Patterns.
- ein **Security Server** als Zugangssystem (SSL Tunneling) zu den provisionierten Systemen in der Cloud. Sämtliche Testsysteme sind für Deploymentaktivitäten nicht über das Internet sondern ausschliesslich über diesen Security Proxy erreichbar. Die

einigen über das Internet zugänglichen Systeme befinden sich in der aktiven Produktionsline. We pre-provisioned three networking zones / VLANs and a Loadbalancer. All of them were provisioned upfront because they don't change permanently.

Wie bereits oben erwähnt, erfolgt mit jedem Change ein kompletter Staging Prozess über alle Teststufen – im Sinne einer Continuous Delivery Pipeline vollständig automatisiert. Dies schliesst ein automatisiertes Testen nach erfolgreichem Provisionierung der Infrastruktur und anschliessendem Anwendungsdeployoment ein. Nach erfolgreicher Abnahme einer Teststufe erfolgt der Übergang in die nächste Stage.