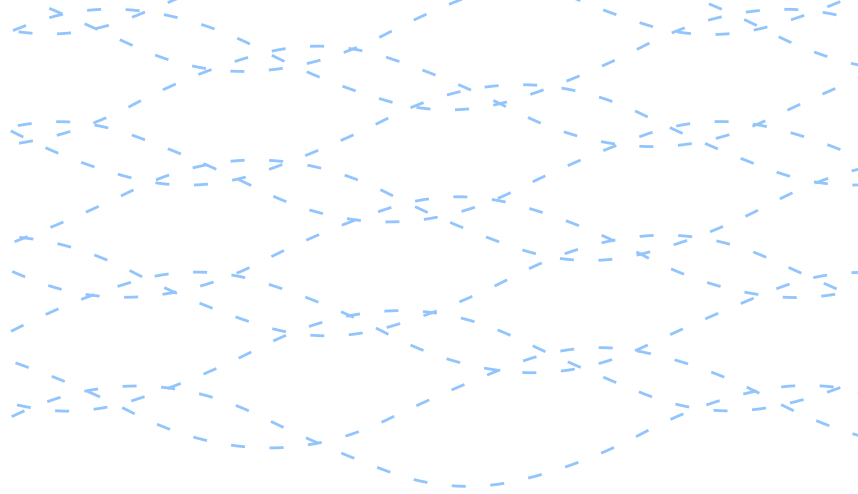




beta systems



JobZ

Workload Automation in Kubernetes – Beta JobZ Cloud Connector als Brücke für moderne IT-Automation

Die Herausforderung der modernen Unternehmens-IT

Die IT-Landschaften in Unternehmen sind heute heterogen wie nie zuvor: z/OS-Mainframes, Unix/Linux-Server, Windows-Systeme, SAP-R/3- und S/4HANA-Plattformen sowie Cloud- und Container-Umgebungen müssen parallel betrieben, gesteuert und überwacht werden. Klassische Workload-Automation-Systeme (WLA) wie **IBM Workload Scheduler (IWS)** haben sich über Jahrzehnte als Rückgrat der unternehmensweiten Prozessautomation etabliert und sorgen für die fehlerfreie Ausführung von Millionen Jobs pro Jahr. Doch mit der zunehmenden Verbreitung von **Container-Technologien**, insbesondere **Kubernetes**, stoßen diese etablierten Systeme an Grenzen: Workloads werden fragmentiert, Logs und Returncodes verstreut, und die einheitliche Steuerung über Plattformgrenzen hinweg ist schwierig.

Hier setzt der **Beta JobZ Cloud Connector** an: Er erweitert die vorhandenen Automationssysteme, integriert Kubernetes-Workloads in bestehende Workflows und stellt sicher, dass alle Jobs – ob auf Mainframe, On-Premises-Systemen oder in Cloud-Umgebungen – zentral gesteuert, überwacht und nachvollziehbar bleiben.

Beta Systems – Wegbereiter plattformunabhängiger Automation

Beta Systems hat sich über Jahre als führender Anbieter für Workload Automation etabliert. Mit Lösungen wie JobZ und dem Cloud Connector modernisiert das Unternehmen klassische Mainframe-Automation und bietet gleichzeitig eine **nahtlose Brücke zu modernen Cloud- und Container-Technologien**. Dies ist ein stabiler und zukunftsweisender Schritt auf dem Weg zu vollständig plattformunabhängigen Automationssystemen.

Workload Automation – eine zentrale Unternehmensaufgabe

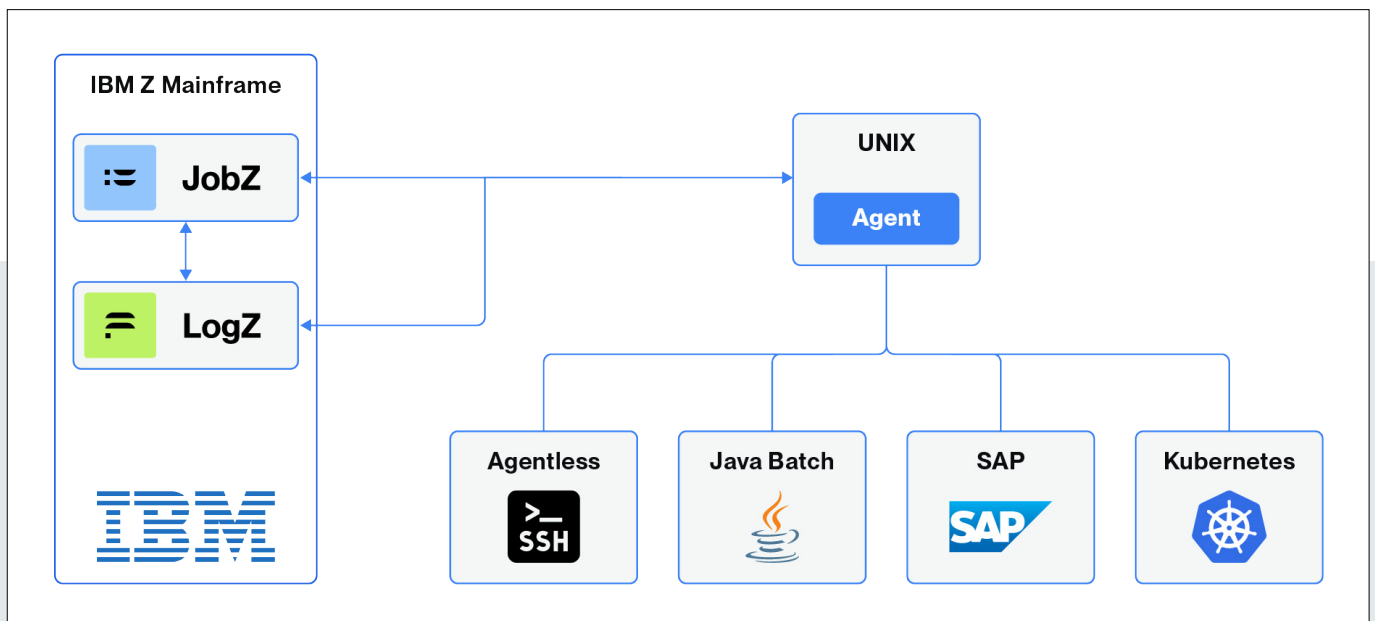
Workload Automation ist mehr als nur Job Scheduling: Sie stellt sicher, dass Geschäftsprozesse reibungslos laufen. In größeren Unternehmen werden über WLA hunderte Millionen Einzelaktivitäten pro Jahr ausgeführt, und die Zuverlässigkeit dieser Prozesse ist mission-critical. IWS beispielsweise zentralisiert die Job-Definitionen, orchestriert Abhängigkeiten und verwaltet Ausführungszeiten über Plattformen hinweg. Ohne diese zentrale Steuerung wäre die Unternehmens-IT anfällig für Fehler, Inkonsistenzen und Ausfallzeiten.

Mit der Containerisierung ändern sich die Anforderungen an WLA grundlegend. Container sind **stateless**, skalierbar und dynamisch – Eigenschaften, die Flexibilität bieten, aber auch neue Herausforderungen für die Automatisierung schaffen.

Containerisierung und Kubernetes – neue Anforderungen

Containerisierung ist mittlerweile Standard in der Unternehmens-IT. Laut einer Studie von techconsult (2024) setzen rund 80 % der Unternehmen Container-Technologien ein oder planen dies. **Kubernetes** ist die dominierende Plattform für Container-Orchestrierung und bietet Services wie Ingress für die Steuerung von HTTP-/HTTPS-Kommunikation in Clustern. Doch während Kubernetes dynamisch und skalierbar ist, fehlt eine einfache Verbindung zu bestehenden zentralen Scheduler-Systemen.

Die zentrale Frage lautet: Wie gelangen Job-Befehle von einem Mainframe-basierten Scheduler wie IWS zu einem Kubernetes-Worker-Pod, wie werden Job-Definitionen übertragen, Logs zurückgemeldet und Abläufe konsistent gehalten?



Grafik 1 – Cloud Connector als Brücke für moderne IT-Automatation



Beta JobZ Cloud Connector – die Brückentechnologie

Der **Cloud Connector** fungiert als Brücke zwischen etablierten WLA-Systemen und modernen Kubernetes-Umgebungen. Er ermöglicht:

- **Zentrale Steuerung:** Jobs aus IWS oder JobZ werden in Kubernetes-Pods ausgeführt, ohne die vorhandene Automationslogik zu verändern.
- **Zentralisierte Rückmeldungen:** Returncodes, Logs und erzeugte Dateien werden gesammelt und dem Scheduler sowie LogZ bereitgestellt.
- **Agentenlose Remote-Ausführung:** Keine zusätzlichen Softwareagenten auf Zielsystemen erforderlich, reduziert Aufwand und Sicherheitsrisiken.
- **Cloud- und plattformunabhängigkeit:** Funktioniert auf On-Premises-Systemen und allen gängigen Cloud-Umgebungen.
- **Kubernetes-native Integration:** Nutzung von Best Practices, Sidecars für Log-Erfassung, Mounten von Persistent Volumes, Git-basierte Job-Definitionen.

Damit wird sichergestellt, dass Workflows **stabil, nachvollziehbar und reversionssicher** bleiben, auch wenn Container dynamisch erstellt oder skaliert werden.

Technische Umsetzung im Detail

1. Ingress & APIs

Der Cloud Connector nutzt Kubernetes-Ingress, um Aufträge vom zentralen Scheduler an die Worker-Nodes zu senden. Dies schafft einen standardisierten, sicheren Zugang ohne manuelle Firewall-Freischaltungen.

2. Stateless Container & Job-Definitionen

Jobs werden nicht fest in Container-Images eingebunden. Stattdessen werden sie über Persistent Volumes bereitgestellt oder direkt aus einem Git-Repository geladen. Das ermöglicht Versionierung, Nachvollziehbarkeit und einfache Updates.

3. Dynamische Laufzeitparameter

Variablen und Ressourcen der Umgebung können zur Laufzeit in die Job-Definition einfließen. So lassen sich Workflows flexibel an wechselnde Cluster-Bedingungen anpassen.

4. Sidecars für Logging und Monitoring

Sidecar-Container erfassen Logs, Returncodes und weitere Output-Daten und leiten sie über API/Ingress zurück an den zentralen Scheduler. So bleibt die Ausführung nachvollziehbar und alle nachgelagerten Jobs können korrekt angesteuert werden.

5. Git-basierte Versionierung

Entwickler können Jobs in ihrer gewohnten Umgebung erstellen, testen und versionieren, ohne dass dies die zentrale Automationsinfrastruktur beeinträchtigt. So wird eine konsistente Pipeline von Entwicklung über Test bis Produktion gewährleistet.

Praxisnutzen und Mehrwert

Durch die Integration von Kubernetes-Workloads in zentrale Scheduler-Systeme profitieren Unternehmen von:

- **Einheitlicher Prozessautomation** über alle Plattformen hinweg
- **Reduzierung von Fragmentierung** und manueller Konsolidierung
- **Zentralisiertem Monitoring & Reporting** mit LogZ und Return Codes
- **Vereinfachtem Sicherheitsmanagement** durch agentenlose Ausführung
- **Skalierbarer Cloud-Integration** ohne Anpassungen oder Plattformwechsel
- **Revisionssicheren, stabilen Workflows** dank Git-basierter Job-Definitionen

Kurz gesagt: Der Cloud Connector verbindet **klassische Stabilität** mit **moderner Flexibilität**, ohne dass Unternehmen ihre etablierten Systeme ersetzen müssen.

Der Cloud Connector bietet eine **effiziente Balance zwischen Automatisierungsaufwand** und **erreichbarer Performance**. Kunden schätzen besonders die Integration in bestehende Scheduler, die zentrale Überwachung und die plattformunabhängige Ausführung.

Ausblick – der Weg in die Zukunft der IT-Automation

Die Digitalisierung beschleunigt den Trend zu Cloud, Containern und dynamischen IT-Umgebungen. Unternehmen, die **Workload Automation über alle Plattformen hinweg integrieren**, sichern sich Stabilität, Transparenz und Effizienz. Der **Beta JobZ Cloud Connector** ermöglicht genau dies: **eine durchgängige Steuerung, konsistente Logs, revisions-sichere Workflows** und **flexible Kubernetes-Integration**. Unternehmen behalten die Kontrolle, reduzieren Risiken und schaffen die Basis für zukünftige Automationsstrategien.

Nutzen Sie die Gelegenheit, Ihre Unternehmens-IT **einheitlich, flexibel** und **zukunftsicher** zu automatisieren.

Kontaktieren Sie Beta Systems für eine **Live-Demo** und erfahren Sie, wie der **Cloud Connector** Ihre Workload Automation auf ein neues Level hebt.