

Proposal

Automatisiertes Deployment von Kundeninstallationen per Konfigurator

Zielsetzung

Mithilfe des Konfigurators soll eine einheitliche, versionierte, auditierbare Installation und Konfiguration von Kundensystem ermöglicht werden. Darüber hinaus soll die vollautomatische Einrichtung von Cloud-Installationen ermöglicht werden.

Ausgangszustand

Die manuelle Installation und Konfiguration von Kundensystemen erschweren erheblich den Rollout von großen Projekten sowie Wartung und Support. Insbesondere die fehlende Nachvollziehbarkeit von Änderungen ist höchst problematisch.

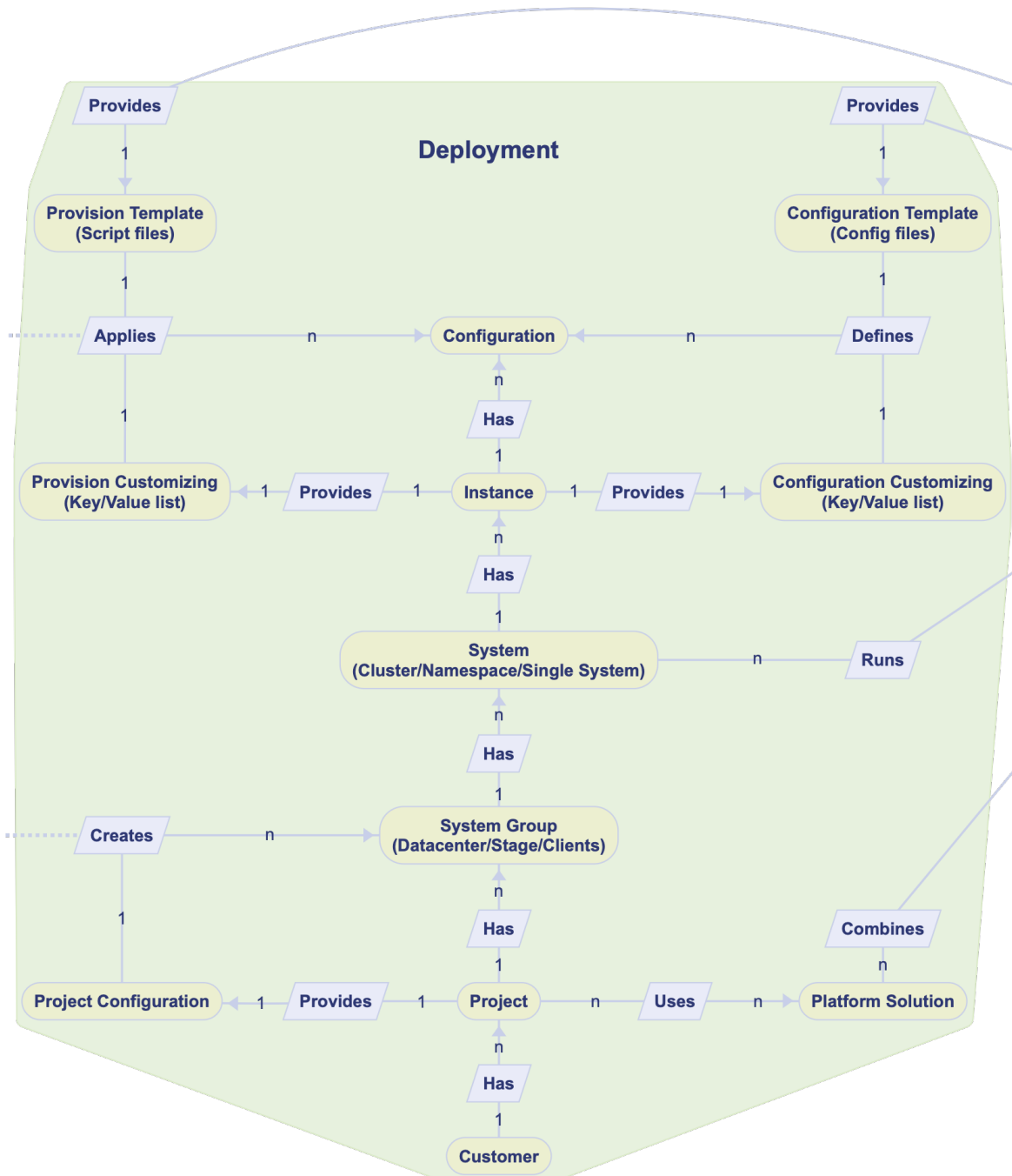
Die Installation und Konfiguration eines Systems teilen sich in mehrere Phasen auf:

- Definieren der Parameter: Hier werden die notwendigen Softwarekomponenten benannt und entschieden, wie die Gesamtlösung auf verschiedene Cluster oder Einzelmaschinen verteilt wird.
- Bereitstellung der Ressourcen: Hier liefert typischerweise der Kunde die im ersten Schritt definierte Infrastruktur. Bei Cloudprojekten müssen entsprechende Ressourcen für unsere Orchestrierungssysteme freigemacht werden.
- Programminstallation: Dies erfolgt manuell für jede Kundeninstanz. In der Cloud werden entsprechende Helm-Charts ausgeführt.
- Programmkonfiguration: Die installierten Programme werden so konfiguriert, dass sie auf den lokalen Instanzen korrekt laufen und die notwendigen Verbindungen zu den anderen Instanzen aufbauen können.
- Kundenspezifische Konfiguration: Nachdem das System lauffähig ist, können die kundenspezifischen Anforderungen in die Konfiguration übernommen werden.

Bisher werden die letzten 3 Phasen typischerweise manuell auf den jeweiligen Maschinen ausgeführt.

Zielzustand

Ziel ist es, möglichst alle manuellen Arbeiten von den Kundeninstanzen zu trennen. Dazu muss das in Phase 1 definierte Systemmodell formalisiert und dessen Konfiguration gespeichert werden. Für die Speicherung eignet sich ein Graph sehr gut. Dieses Proposal schlägt daher eine Erweiterung des in Proposal „Speicherung von Informationen zu Release-Artefakten“ skizzierten Graphen vor. Der vollständige Graph ist im Anhang abgebildet. Hier ist der neu hinzukommende Teil dargestellt:



Für einen „Kunden“ wird ein neues „Projekt“ angelegt, dessen „Projektkonfiguration“ die in der ersten Phase definierten Struktur enthält. Die „Projektkonfiguration“ zur Verdeutlichung hier als eigene Entität dargestellt. Wahrscheinlich werden diese Daten aber als Teil des „Projekts“ gespeichert.

Ein „Projekt“ besteht aus einer Menge von „Systemgruppen“, dies können z.B. Datacenter oder Standorte des Kunden sein. Dort befinden sich ein oder mehrere „Systeme“, welche z.B. ein Cluster in einem Datacenter repräsentieren. Ein „System“ besteht aus mindestens einer „Instanz“, die eine konkrete Maschine (z.B. VM) bezeichnet.

Auf Projektebene gibt es eine Verknüpfung zu „Plattformlösungen“, die im Projekt verwendet werden sollen („Uses“). Eine Lösung beschreibt eine Menge von Produkten, die zusammen eingesetzt werden sollen („Combines“). Dies könnte z.B. die Lösung „Spool-Only“ sein, in der ein P5 mit easyPRIMA, Keycloak und Elastic Stack enthalten ist. Eine Solution definiert dabei z.B. auch, ob ein Produkt auf einem Cluster oder einer Einzelmaschine laufen soll (z.B. P5 im Cluster, easyPRIMA auf einem Einzelrechner).

Auch definiert die Lösung die Möglichkeiten zu kundenspezifischen Anpassungen. Diese können dann auf Projekt- oder System-Ebene realisiert werden (z.B. Stempelkonfiguration).

Die in den gewählten Lösungen enthaltenen Produkte können entsprechend der definierten Rahmenbedingungen mit Systemen verknüpft werden („Runs“). So kann ein P5 im Beispiel nur mit einem Cluster-System verknüpft werden. Dabei kann aus den freigegebenen Versionen gewählt werden, die kompatibel mit den anderen Produkten der Lösung sind.

Eine Produktversion bietet („Provides“) ein für die Lösung passendes „Konfigurationstemplate“ an. Dies sind z.B. Konfigurationsdateien, bei denen bestimmte Inhalte über Variablen bestimmt werden können. Verschiedene Lösungen können ggf. verschiedene Templates enthalten. Die Anzahl der Templates sollte allerdings möglichst klein gehalten werden, um den Aufwand von notwendigen Änderungen zu minimieren.

Ebenso wird ein „Provisionstemplate“ bereitgestellt, in dem z.B. die notwendigen Skripte zur Installation und Konfiguration des Produkts enthalten sind. Auch hier können über Variablen Anpassungen erfolgen.

Eine Instanz bietet die Möglichkeit, die Variablen für „Konfigurationsanpassungen“ festzulegen. Aus dem Template und den Anpassungen wird die endgültige „Konfiguration“ erstellt und z.B. auf unserer Auslieferungsplattform in einem Projektordner abgelegt.

Ebenso können die Variablen für „Provisionsanpassungen“ festgelegt werden. Hieraus entstehen Skripte zum Installieren der Produkte und das Einspielen der Konfiguration. Diese werden ebenfalls im Projektordner für die betreffende Instanz gespeichert.

Das Generieren der Konfigurationen und Provisionsskripte erfolgt über den Workflow des Konfigurators mittels eines entsprechenden Hubs. Dieser Hub muss nicht beim Kunden installiert sein, da er lediglich Daten auf unserer Auslieferungsplattform ablegt. Als Ergebnis existiert für jede Instanz ein Verzeichnis mit allen notwendigen Installations- und Konfigurationsdateien.

Alle Instanz-spezifischen Konfigurationseinträge (z.B. IP-Adresse/Hostname) sind bereits angepasst. Ebenso sind alle benötigten externen Systeme bereits in der Konfiguration referenziert. Auch die kundenspezifischen Anpassungen der Plattformlösungen sind in die Konfigurationsdateien eingeflossen.

Für die Installation muss nur ein Startskript aufgerufen werden. Der PE muss also nur das Verzeichnis herunterladen, entpacken und das Skript aufrufen, um die Instanz zu initialisieren. Ein manueller Eingriff ist nicht notwendig. In einem Cluster ist möglicherweise ein mehrstufiger Prozess notwendig, in dem die einzelnen Stufen manuell gestartet werden, nachdem alle Instanzen des Clusters die letzte Stufe erreicht haben.

Wenn bei einem Kunden ein Konfigurator Hub installiert werden kann, können selbst diese Schritte vollautomatisiert erfolgen.

Im SaaS-Umfeld kann auch die Bereitstellung der Ressourcen über einen entsprechenden Konfigurator Hub erfolgen:

In einem OpenShift/k8s-Cluster („Systemgruppe“) kann dieser z.B. als Controller laufen. Die Konfigurationen werden hier im jeweiligen Namespace („System“) über ConfigMaps den Containern („Instanzen“) zur Verfügung gestellt. Eine klassische Installation ist hier nicht notwendig, da entsprechende Containerimages bereits gebaut wurden.

Auf AWS und vCloud können die gewünschten Systemlandschaften mittels Terraform angelegt werden. Auch hybride Ansätze mit (z.B. On Premises und AWS) sind mit dem hier vorgeschlagenen Ansatz vorstellbar.

Last but not least kann dies die Erstellung von Testsystemen für Kunden wesentlich vereinfachen. Da die gesamte Kundenlandschaft gespeichert ist, können problemlos in der vCloud oder auf AWS entsprechende Systeme (auch teilweise) aufgebaut werden. Diese enthalten automatisch die kundenspezifischen Anpassungen und sind für die jeweilige Umgebung fertig eingerichtet (z.B. IP-Adressen). Ein Klonen oder dauerhaftes Vorhalten von Instanzen ist damit nicht mehr notwendig.

Erste Maßnahmen

Es sollte ein MVP erstellt werden, der die Konfiguration aller Instanzen eines P5-Clusters (ohne kundenspezifische Anpassungen) auf der Auslieferungsplattform ablegt.

Anhang

