

8 Zusammenfassung und weitere Arbeit

Externe Referenzen können Rechnerwolken negativ beeinflussen. Wenn wir ein hohes Niveau der Informationssicherheit oder einen begrenzten Zugang zum Internet haben, können diese externen Abhängigkeiten zu vielen Problemen mit Leistung, Stabilität und Sicherheit führen. Leider greifen viele TOSCA Applikationen während der Bereitstellung auf externe Quellen zu, um Pakete zu installieren oder Dateien herunterzuladen. Das Ziel dieser Arbeit war es, eine Lösung zu entwickeln, indem man externe Abhängigkeiten durch Datenkapselung auflöst, und die Lösung in einen Prototyp zu implementieren.

Es wurde ein Konzept des modularen Frameworks entworfen. Dem Konzept entsprechend werden die Module verschiedene Konfigurationsmanagement-Tools, Paketverwaltungen und Dienstprogramme zum Herunterladen von Files bearbeiten. Jedes Modul ist für die Identifizierung und Auflösung einer bestimmten Art der externen Referenzen verantwortlich. Nach der Auflösung werden die heruntergeladenen Dateien in die Topologie der TOSCA Applikationen integriert. Es wurden mehrere Integrationsmodi präsentiert, die für verschiedene Anwendungsfälle geeignet sein können. Der Prototyp des modularen Frameworks wurde in Java Sprache entwickelt. Das Programm identifiziert und löst Referenzen zu externen Paketen auf. Die Anwendung kann *Bash*-Skripten und *Ansible*-playbooks bearbeiten. Sie hat zwei Arbeitsmodi. Im ersten Modus werden die Abhängigkeiten zwischen den Paketen auf die TOSCA Topologie abgebildet. Das andere Modus dient für die Generierung eines kompakten CSAR mit wenigen Knoten. Das Framework bearbeitet ein CSAR folgendermaßen. Die Struktur des CSAR wird analysiert, um innere Referenzen zu ermitteln. Dann finden die Module externe Abhängigkeiten, entfernen sie und erstellen eine Liste mit den Namen von benötigten Paketen. Jedes Paket wird zusammen mit allen abhängigen Paketen heruntergeladen und ins CSAR integriert. TOSCA Knoten für diese Pakete werden erstellt. Während der Bereitstellung kann die TOSCA Laufzeitumgebung diese Knoten analysieren und die benötigten Pakete installieren.

Um die Erweiterbarkeit des Frameworks anzuzeigen, wurde die Hinzufügung des *aptitude* Paketverwaltungsmoduls dem *Bash* Modul detailliert beschrieben. Es wurde gezeigt, wie man ein Modul erzeugt, das dem Framework hinzugefügt werden kann, wie man seine Grundfunktionen implementiert, Dateien weitergibt und dieses Modul ins *Bash* Modul integriert.

Zum Schluß wurden die Ergebnisse der Ausführung des Frameworks geprüft. Die

ausgegebenen CSARs wurden mit Hilfe von Winery visualisiert und analysiert. Die erzeugten Artefakte wurden verifiziert und ausgeführt.

Weitere Arbeit

Das entwickelte Framework repräsentiert den Prototyp einer Software, die einmal fähig wird, das bearbeitende CSAR völlig in sich geschlossen zu machen. Jetzt kann er erweitert werden oder sogar sein Konzept kann verändert werden. Es gibt viele verschiedene Richtungen vorhanden, um den bestehenden Prototyp zu erweitern. Die Paketverwandlungsmodule können übergearbeitet werden, damit sie die Installationsanweisungen besser verarbeiten. Zum Beispiel kann man die Unterstützung für Variablen dem apt-get Modul für Bash zuzugeben. Einige neue Paketverwandlungs- und Konfigurationsmanagementmodule können entwickelt und dem Framework hinzugefügt werden, um zum Beispiel Shuf, CFEngine, yum, pacman, etc. zu bearbeiten. Die beschriebenen, aber noch nicht implementierten Module für Anwendungen zum Herunterladen von Files können in Zukunft hinzugefügt werden.

Das gesamte Konzept kann übergearbeitet werden, um das höhere Niveau der Abstraktion zu erreichen. Alle Arten der Module können zu einer abstrakten Art gruppiert werden. Das ermöglicht die Hinzufügung von einigen neuen Arten der Komponenten, so wie Archivmodule mit zip oder rar Submodulen, die die Archive ohne große Änderungen in der Struktur der Software getrennt bearbeiten können. Die andere Richtung der Vervollkommnung der Zweckmäßigkeit der Software ist es, eine visuelle Schnittstelle zu implementieren, die es dem Benutzer ermöglicht, die Artefakte, die verarbeitet werden müssen, und die Modi der Verarbeitung getrennt zu wählen. Solche Visualisierung kann auf einem bestehenden Programm aus OpenTOSCA basieren, zum Beispiel auf Winery. Das kann für zusammengesetzte Rechnerwolken, die aus vielen Teilen mit vielfältigen Architekturen bestehen, welche auf unterschiedliche Weisen getrennt bearbeitet werden müssen, nützlich sein.