



Silvan Adrian



Fabian Binna

Diplomanden	Silvan Adrian, Fabian Binna
Examinator	Prof. Beat Stettler
Experte	Urs Baumann
Themengebiet	Internet-Technologien und -Anwendungen

Software Defined Data Center (SDDC)

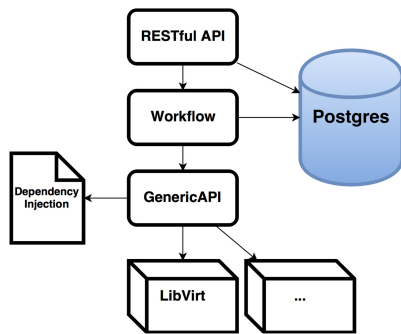
Software Defined Everything

SDDC

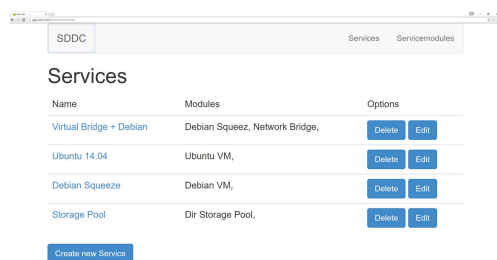
Software Defined Data Center



Im Projekt verwendete Frameworks und Bibliotheken



Übersicht der Architektur



Das Dashboard

Ausgangslage: Unter «Software Defined» versteht man die Zentralisierung der Intelligenz in Controllern. Gerade moderne Data Center werden immer häufiger von Software Controllern gesteuert, damit die Dynamik der Bereitstellung von neuen Services massiv erhöht werden kann. Ziel ist, die Ressourcen Storage, Network und Compute abstrahiert als skalierbare Pools der «Service Ebene» zur Verfügung zu stellen. Dies soll als eine generische Middleware/Application Programming Interface (API) realisiert werden, um verschiedene Controller möglichst einfach in Business Applikationen zu integrieren.

Vorgehen/Technologien: Nach der Analyse verschiedenster Cloud Provider und Libraries wurde eine möglichst systemunabhängige Schnittstelle definiert. Damit dies überhaupt möglich ist, musste ein strikt einzuhaltender Contract erstellt werden. Ein Modul beinhaltet die Konfiguration für eine Ressource und beim erstellen einer Ressource wird ein Identifier zurückgegeben, wobei Modul und Identifier völlig abstrakt sind. Damit die Software in Business Applikationen integriert werden kann wurde eine RESTful API definiert, die Services abstrahiert. Diese beiden Schnittstellen müssen mit einem Workflow zusammengefügt werden. Der Workflow bestimmt, bei Bestellung eines Services, welche Module zum Einsatz kommen und persistiert die Identifier, um die erstellten Ressourcen später wieder anzusprechen.

Ergebnis: Das Resultat ist ein Webservice, der mit Hilfe von Spring implementiert wurde. Die Datenhaltung läuft auf Postgres und wird über ein OR-Mapping an die Domain angehängt. Der Webservice kann über die RESTful API angesprochen werden. Die generische API kann in einer Extensible Markup Language (XML) Datei definiert werden. Beim Start des Webservices wird die generische API mit Spring Dependency Injection zusammengesetzt. So kann die generische API beliebig mit neuen Controllern erweitert und an den Webservice angehängt werden. Für Demonstrationszwecke entstand eine Webseite, die es erlaubt Services zu bestellen, anzusehen, zu bearbeiten und wieder zu löschen.