

SDDC

Software Defined Data Center

Anforderungsspezifikation

Silvan Adrian
Fabian Binna

1 Änderungshistorie

| Datum | Version | Änderung | Autor |
|----------|---------|--|---------------|
| 02.10.15 | 1.00 | Erstellung des Dokuments | Gruppe |
| 02.10.15 | 1.01 | Nicht funktionale Anforderungen | Silvan Adrian |
| 02.10.15 | 1.02 | Use Cases Aktoren + User Stories Aktoren | Silvan Adrian |
| 03.10.15 | 1.03 | Anforderungen API | Fabian Binna |
| 03.10.15 | 1.04 | Anforderungen Dashboards + Mockups eingefügt | Silvan Adrian |
| 03.10.15 | 1.05 | Use Cases fully dressed | Silvan Adrian |

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Änderungshistorie | 2 |
| 2 | Einführung | 5 |
| 2.1 | Zweck | 5 |
| 2.2 | Gültigkeitsbereich | 5 |
| 2.3 | Referenzen | 5 |
| 3 | Anforderungen | 5 |
| 3.1 | API | 5 |
| 3.2 | Customer-Dashboard | 6 |
| 3.2.1 | Homescreen | 6 |
| 3.2.2 | Services Übersicht | 6 |
| 3.2.3 | Service abonnieren | 6 |
| 3.2.4 | Cloud Credentials | 7 |
| 3.3 | Admin-Dashboard | 7 |
| 3.3.1 | Service | 7 |
| 3.3.2 | Servicemodul | 8 |
| 4 | Use Cases | 9 |
| 4.1 | Use Case Diagramm | 9 |
| 4.2 | Aktoren & Stakeholders | 9 |
| 4.2.1 | Customer | 9 |
| 4.2.2 | Admin | 10 |
| 4.3 | Beschreibungen fully dressed | 11 |
| 4.3.1 | UC01: Service abonnieren | 11 |
| 4.3.2 | UC02: Service kündigen | 13 |
| 4.3.3 | UC03: Cloud Login Daten verwalten | 15 |
| 4.3.4 | UC04: Services verwalten | 16 |
| 4.3.5 | UC05: Servicemodule verwalten | 18 |
| 5 | Epics | 20 |
| 5.1 | Customer | 20 |
| 5.2 | Admin | 20 |
| 6 | User Stories | 20 |
| 6.1 | Rollen | 20 |
| 6.1.1 | Customer | 20 |
| 6.1.2 | Admin | 20 |
| 6.2 | Customer | 20 |
| 6.2.1 | Customer greift auf Customer-Dashboard zu | 20 |
| 6.3 | Admin | 21 |
| 6.3.1 | Admin greift auf Dashboard zu | 21 |

| | | |
|----------|---------------------------------------|-----------|
| 6.3.2 | Admin erstellt Servicemodul | 21 |
| 6.3.3 | Admin ändert Servicemodul | 21 |
| 7 | Nichtfunktionale Anforderungen | 22 |
| 7.1 | Menge | 22 |
| 7.2 | Schnittstellen | 22 |
| 7.3 | Qualitätsmerkmale | 22 |
| 7.3.1 | Funktionalität | 22 |
| 7.3.2 | Zuverlässigkeit | 22 |
| 7.3.3 | Benutzerbarkeit | 22 |
| 7.3.4 | Effizienz | 22 |
| 7.3.5 | Änderbarkeit | 22 |
| 7.3.6 | Übertragbarkeit | 23 |

2 Einführung

2.1 Zweck

Dieses Dokument beinhaltet die Anforderung zur Analyse.

2.2 Gültigkeitsbereich

Dieses Dokument ist während des ganzen Projekts gültig.

2.3 Referenzen

-

3 Anforderungen

3.1 API

Die API definiert einen Workflow der einen Service auf einer Cloud erstellt. Es ist offen, ob dieser Service über mehrere Cloud Anbieter hinaus geht. Der Service wird durch ein Konfigurationsfile (z.B. json) definiert. Zusätzlich können Scriptfiles referenziert werden, die die Software auf den Instanzen installieren. Ein Service kann auch wieder gelöscht werden. Es ist nicht die Aufgabe der API existierende Services zu identifizieren. Die API muss Modular sein, das heisst es sollte möglich sein andere oder eigene Programme für die Cloud Kommunikation zu verwenden. Innerhalb der API werden Compute, Storage, Network usw. als ServiceModule bezeichnet. Diese Abstraktion ermöglicht das wiederverwenden und erweitern der API.

3.2 Customer-Dashboard

3.2.1 Homescreeen

Im Homescreeen (sobald der Customer auf das Dashboard zugreift) werden alle zu Verfügung stehenden Services angezeigt. Hier werden die Services Offerings genannt um eine Unterscheidung zwischen Abonnierten Services (Services) und zur Verfügung stehenden Services (Offerings) machen zu können.

Dashboard

Offerings Services Account

Offerings



3.2.2 Services Übersicht

In der Services Übersicht werden dem Customer alle abonnierten Services angezeigt und können hier auch gekündigt werden.

Dashboard

Offerings Services Account

Your Services

| Name | Options |
|--|-----------|
| LAMP Stack | terminate |
| MEAN Stack | terminate |
| <input type="button" value="subscribe Service"/> | |

3.2.3 Service abonnieren

Sobald ein Service auf dem Homescreeen ausgewählt wird muss noch der zuständige Provider gewählt werden (häng auch wieder davon ab von welchem Provider bisher Login Daten hinterlegt wurden). Da momentan noch kein Hybrid Betrieb vorgesehen ist muss ein Account gewählt werden unter welchem der Service verrechnet wird.

Dashboard

Offerings Services Account

LAMP Stack

Account

OpenStack

subscribe Service

3.2.4 Cloud Credentials

Da für jeden Provider wieder Login Daten benötigt werden müssen diese an einer zentralen Stelle gespeichert werden (anhand von denen kann sich auch das Service Angebot ändern).

Dashboard

Offerings Services Account

[Home](#) > [Account](#) > Cloud Credentials

Your Cloud Credentials

| Name | Provider | Options |
|----------------|--------------|-----------------|
| silvan | OpenStack | delete / change |
| sadrian@hsr.ch | Digitalocean | delete / change |

add Credentials

3.3 Admin-Dashboard

Zusätzlich zum Customer-Dashboard soll ein Admin-Dashboard zur Verfügung stellen in welchem der Admin Services und Servicemodule erstellen kann.

3.3.1 Service

Ein Service hat einen bestimmten Namen und jedem Service sind eine gewisse Anzahl Servicemodule zugeteilt, um den Service abbilden zu können. Hier kann der Admin den Service ändern und je nach Anforderung den Service anpassen.

Dashboard

Services **Servicemodules** **Account**

Services

| Name | Aktion |
|---|-----------------|
| LAMP Stack | delete / change |
| MEAN Stack | delete / change |
| Build Server | delete / change |
| <input type="button" value="create Service"/> | |

3.3.2 Servicemodul

Jedes Servicemodul besitzt einen Namen und wird einem Provider zugeschrieben, dabei kann jedes Servicemodule den Typ Compute, Network oder Storage haben.

Dashboard

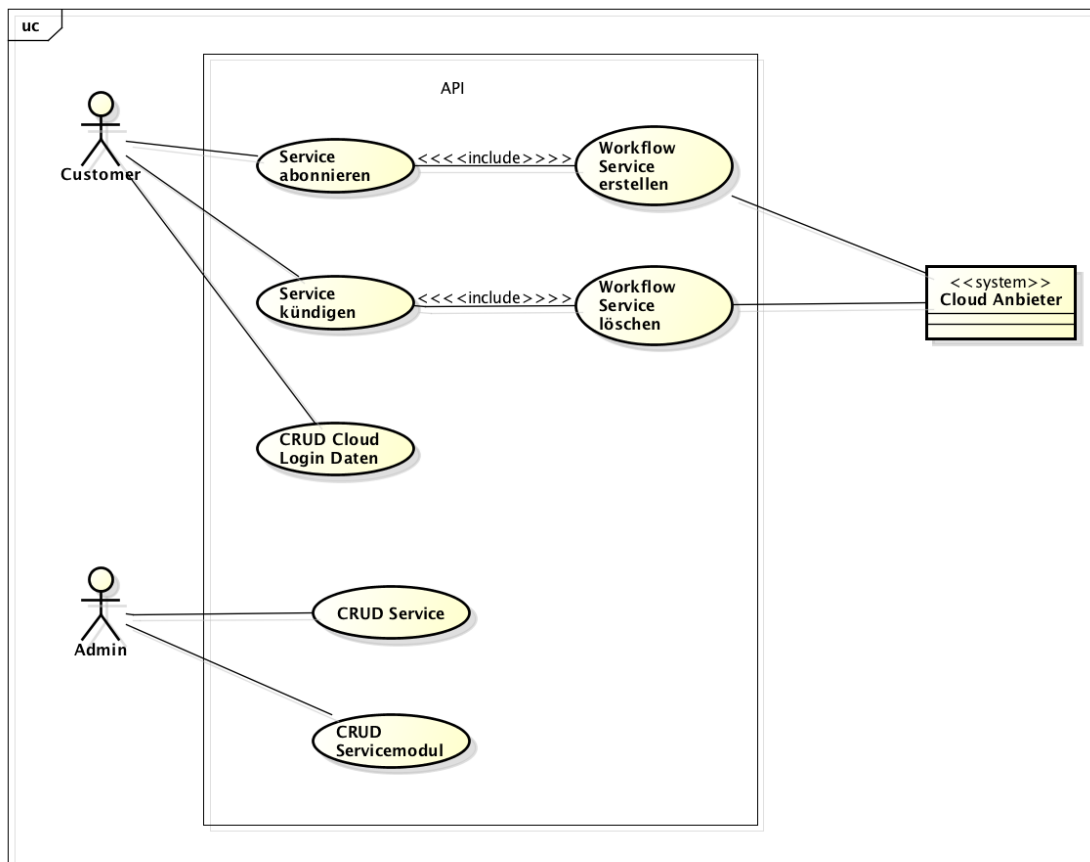
Services **Servicemodules** **Account**

Servicemodules

| Name | Provider | Type | Options |
|---|-------------|---------|-----------------|
| LAMP Instance | OpenStack | Compute | delete / change |
| MEAN Instance | Azure | Compute | delete / change |
| Build Server Instance | Cloud Stack | Compute | delete / change |
| <input type="button" value="create Servicemodule"/> | | | |

4 Use Cases

4.1 Use Case Diagramm



powered by Astah

4.2 Aktoren & Stakeholders

4.2.1 Customer

Als Customer möchte ich meine abonnierten Services verwalten.

| Aktor | Typ | Ziele |
|----------|---------|--------------------------------|
| Customer | Primary | • Service abonnieren |
| | | • Service kündigen |
| | | • Cloud Login Daten hinzufügen |
| | | • Cloud Login Daten anpassen |
| | | • Cloud Login Daten löschen |

4.2.2 Admin

Als Admin möchte ich Services und Servicemodule verwalten können.

| Aktor | Typ | Ziele |
|-------|---------|--|
| Admin | Primary | <ul style="list-style-type: none">• Service erstellen |
| | | <ul style="list-style-type: none">• Service anpassen |
| | | <ul style="list-style-type: none">• Service löschen |
| | | <ul style="list-style-type: none">• Servicemodul erstellen |
| | | <ul style="list-style-type: none">• Servicemodul anpassen |
| | | <ul style="list-style-type: none">• Servicemodul löschen |

4.3 Beschreibungen fully dressed

4.3.1 UC01: Service abonnieren

| | |
|------------------------------------|---|
| Primäraktor | Customer |
| Steakholders und Interessen | Customer: Möchte einen Service abonnieren |
| Vorbedingungen | Das Customer-Dashboard wurde geöffnet, ist bei der API authentifiziert und hat einen Cloud Account hinzugefügt. |
| Nachbedingungen | Die Service Infos wurden gespeichert und der Workflow wurde angestossen |

Standartablauf

1. Der Customer gibt die Webadresse für das Dashboard ein
2. Wiederholen bis kein Service mehr abonniert werden soll
 - a) Der Customer wechselt in die **Offerings Übersicht**
 - b) Der Customer wählt einen der vorhandenen **Services** aus
 - c) Der Customer wählt einen vorhanden **Account** auf welchem er den **Service** abonnieren will
 - d) Der Customer drückt den Button **subscribe Service**
 - e) Der Customer wird in die **Services Übersicht** weitergeleitet
3. Der Customer schliesst das Customer-Dashboard

Alternativer Ablauf

1. a) Der Customer entscheidet sich um und schliesst das Fenster/Tab
2. a) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tab
- b) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tab
 - ii. geht zurück in die **Offerings Übersicht**
- c) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tab
 - ii. geht zurück in die **Offerings Übersicht**
 - iii. wählt einen anderen Account
- d) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tab
 - ii. geht zurück in die **Offerings Übersicht**
 - iii. wählt einen anderen Account

| | |
|--|--------------------------------------|
| Spezielle Anforderungen | siehe nichtfunktionale Anforderungen |
| Technologie- und Datenvarianten | Keine |
| Auftrittshäufigkeit | mehrmals pro Woche |
| Offene Fragen | Keine |

4.3.2 UC02: Service kündigen

| | |
|------------------------------------|---|
| Primäraktor | Customer |
| Steakholders und Interessen | Customer: Möchte einen Service kündigen |
| Vorbedingungen | Das Customer-Dashboard wurde geöffnet, ist bei der API authentifiziert und hat einen Service abonniert |
| Nachbedingungen | Die Service Infos wurden gelöscht und der Workflow wurde angestossen |
| Standartablauf | <ol style="list-style-type: none"> 1. Der Customer gibt die Webadresse für das Dashboard ein 2. Wiederholen bis kein Service mehr gekündigt werden soll <ol style="list-style-type: none"> a) Der Customer wechselt in die Services Übersicht b) Der Customer wählt einen der vorhanden Services aus c) Der Customer drückt auf den link terminate d) Der Customer wird in die Services Übersicht weitergeleitet 3. Der Customer schliesst das Customer-Dashboard |

Alternativer Ablauf

1. a) Der Customer entscheidet sich um und schliesst das Fenster/Tab
 2. a) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tabb) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tab
 - ii. wählt einen anderen Servicec) Der Customer entscheidet sich um
 - i. Schliesst das Fenster/Tab
 - ii. wählt einen anderen Service
-

| | |
|--------------------------------|--------------------------------------|
| Spezielle Anforderungen | siehe nichtfunktionale Anforderungen |
|--------------------------------|--------------------------------------|

| | |
|--|-------|
| Technologie- und Datenvarianten | Keine |
|--|-------|

| | |
|----------------------------|--------------------|
| Auftrittshäufigkeit | mehrmals pro Woche |
|----------------------------|--------------------|

| | |
|----------------------|-------|
| Offene Fragen | Keine |
|----------------------|-------|

4.3.3 UC03: Cloud Login Daten verwalten

| | |
|------------------------------------|--|
| Primäraktor | Customer |
| Steakholders und Interessen | Customer: Möchte ich Cloud Login Daten verwalten |
| Vorbedingungen | Das Customer-Dashboard wurde geöffnet |
| Nachbedingungen | Die Account Infos wurden gespeichert |
| Standartablauf | <ol style="list-style-type: none"> 1. Der Customer gibt die Webadresse für das Dashboard ein 2. Der Customer wechselt in die Account/Cloud Credentials Übersicht 3. Solange wiederholen bis kein Account mehr hinzugefügt werden muss <ol style="list-style-type: none"> a) Customer drückt Button Add Credentials b) Customer füllt gefragt Felder aus c) Customer bestätigt Eingaben mit Klick auf Save 4. Der Customer schliesst das Customer-Dashboard |
| Alternativer Ablauf | <ol style="list-style-type: none"> 3. <ol style="list-style-type: none"> a) Wiederholen bis kein Account mehr geändert werden muss <ol style="list-style-type: none"> i. Account auswählen und auf change Link klicken ii. Felder anpassen iii. Änderungen bestätigen mit Klick auf Button Save b) Wiederholen bis kein Account mehr gelöscht werden muss <ol style="list-style-type: none"> i. Account auswählen und auf delete Link klicken |
| Spezielle Anforderungen | siehe nichtfunktionale Anforderungen |

| | |
|--|--------------------|
| Technologie- und Datenvarianten | Keine |
| Auftrittshäufigkeit | mehrmals pro Woche |
| Offene Fragen | Keine |

4.3.4 UC04: Services verwalten

| | |
|------------------------------------|---|
| Primäraktor | Customer |
| Steakholders und Interessen | Admin: Möchte einen Service verwalten |
| Vorbedingungen | Das Admin-Dashboard wurde geöffnet und Admin eingeloggt, falls Service gelöscht werden soll darf kein Customer mehr den Service abonniert haben |
| Nachbedingungen | Das Admin-Dashboard wurde geschlossen und Änderungen wurden gespeichert |

Standartablauf

1. Der Admin gibt die Webadresse für das Admin-Dashboard ein
2. Wiederholen bis kein neuer Service hinzugefügt werden muss
 - a) Der Admin wechselt in die **Services Übersicht**
 - b) Der Admin drückt auf den button **create Service**
 - c) Der Admin füllt die benötigten Daten ein (**Name, welche Servicemodule**)
 - d) der Admin bestätigt mit Klick auf Button **Save**

Alternativer Ablauf

2.
 - a) Wiederholen bis kein Service mehr geändert werden muss
 - i. Service auswählen und auf Link **change** klicken
 - ii. Daten ändern
 - iii. Durch Klick auf Button **Save** bestätigen
 - b) Wiederholen bis kein Service mehr gelöscht werden muss
 - i. Service auswählen und Link **löschen** auswählen

| | |
|--|--------------------------------------|
| Spezielle Anforderungen | siehe nichtfunktionale Anforderungen |
| Technologie- und Datenvarianten | Keine |
| Auftrittshäufigkeit | mehrmals pro Woche |
| Offene Fragen | Keine |

4.3.5 UC05: Servicemodule verwalten

| | |
|------------------------------------|---|
| Primäraktor | Customer |
| Stakeholders und Interessen | Admin: Möchte ein Servicemodul verwalten |
| Vorbedingungen | Das Admin-Dashboard wurde geöffnet und Admin eingeloggt, falls Servicemodul gelöscht werden soll darf kein Service mehr das Servicemodul verwenden. |
| Nachbedingungen | Das Admin-Dashboard wurde geschlossen und Änderungen wurden gespeichert |

Standartablauf

1. Der Admin gibt die Webadresse für das Admin-Dashboard ein
2. Wiederholen bis kein neues Servicemodul hinzugefügt werden muss
 - a) Der Admin wechselt in die **Servicemodules Übersicht**
 - b) Der Admin drückt auf den button **create Servicemodule**
 - c) Der Admin füllt die benötigten Daten ein (**Name, Provider, Typ**)
 - d) der Admin bestätigt mit Klick auf Button **Save**

Alternativer Ablauf

2.
 - a) Wiederholen bis kein Servicemodul mehr geändert werden muss
 - i. Servicemodul auswählen und auf Link **change** klicken
 - ii. Daten ändern
 - iii. Durch Klick auf Button **Save** bestätigen
 - b) Wiederholen bis kein Servicemodul mehr gelöscht werden muss
 - i. Service auswählen und Link **löschen** auswählen

| | |
|--|--------------------------------------|
| Spezielle Anforderungen | siehe nichtfunktionale Anforderungen |
| Technologie- und Datenvarianten | Keine |
| Auftrittshäufigkeit | mehrmals pro Woche |
| Offene Fragen | Keine |

5 Epics

5.1 Customer

- Service abonnieren
- Service kündigen
- Cloud Login Daten verwalten

5.2 Admin

- Service verwalten
- Servicemodul verwalten

6 User Stories

6.1 Rollen

6.1.1 Customer

Als Customer benutze ich das Dashboard, um für mich einen Service zu abonnieren oder zu kündigen, ebenfalls verwalte ich meine Cloud Login Daten

6.1.2 Admin

Als Admin erstelle ich neue Services und Servicemodule und erweitere diese um neue Funktionen/Verbesserungen.

6.2 Customer

6.2.1 Customer greift auf Customer-Dashboard zu

| | |
|---------------------------|--|
| Prio | Hoch |
| Story Points | 3 |
| Story | Als Customer möchte ich auf das Customer-Dashboard zugreifen können |
| Akzeptanzkriterien | |
| A1 | Der Customer kann den Url des Customer-Dashboard aufrufen und kriegt das Dashboard angezeigt |

6.3 Admin

6.3.1 Admin greift auf Dashboard zu

| | |
|---------------------------|--|
| Prio | Hoch |
| Story Points | 3 |
| Story | Als Admin möchte ich auf das Admin-Dashboard zugreifen können |
| Akzeptanzkriterien | |
| A1 | Der Admin kann den Url des Customer-Dashboard aufrufen und kriegt ein Login angezeigt. |
| A2 | Der Admin kann sich einloggen und kriegt die Service Übersicht angezeigt |

6.3.2 Admin erstellt Servicemodul

| | |
|---------------------------|--|
| Prio | Hoch |
| Story Points | 3 |
| Story | Als Admin möchte ich Servicemodule erstellen können |
| Akzeptanzkriterien | |
| A1 | Das Servicemodule kann nur erstellt werden, falls Admin eingeloggt ist. |
| A2 | Das Servicemodul wird erstellt und wird in der Servicemodule Übersicht angezeigt |

6.3.3 Admin ändert Servicemodul

| | |
|---------------------------|---|
| Prio | Hoch |
| Story Points | 3 |
| Story | Als Admin möchte ich Servicemodule ändern können |
| Akzeptanzkriterien | |
| A1 | Als Admin krieg ich die Übersicht der verfügbaren Servicemodule |
| A2 | Als Admin kann ich Name, Provider anpassen und speichern |
| A3 | Als Admin kann ich die Änderungen einsehen |

7 Nichtfunktionale Anforderungen

7.1 Menge

- Die Software unterstützt mehr als 30 Cloud Anbieter (libcloud)
- Bei jedem Cloud Anbieter bestehen eine gewisse Anzahl Services (von Anbieter zu Anbieter verschieden)

7.2 Schnittstellen

- Die Software wird über HTTP/HTTPS angesprochen
- Zur Interaktion im Admin-Dashboard/Customer-Dashboard werden die herkömmlichen Schnittstellen gebraucht (Maus,Tastatur,Bildschirm)

7.3 Qualitätsmerkmale

7.3.1 Funktionalität

siehe Abschnitt API und Dashboard

7.3.2 Zuverlässigkeit

- Der Workflow zum erstellen eines Services soll entweder durchgeführt und abgeschlossen werden oder falls Unterbruch/Fehler rückgängig gemacht werden.
- Die Software soll verteilt betrieben werden und eine möglichst hohe Verfügbarkeit bieten

7.3.3 Benutzerbarkeit

- Die Software kann über das vorgesehene Admin-Dashboard benutzt werden
- Die API kann auch über die Kommandozeile angesprochen werden

7.3.4 Effizienz

-

7.3.5 Änderbarkeit

Die Software soll modular aufgebaut werden, damit Erweiterungen in Zukunft problemlos möglich sind.

7.3.6 Übertragbarkeit

Das Projekt wird in Python geschrieben ist somit also auf Python mindestens in der Version 2.5 angewiesen, kann allerdings durch den Einsatz eines Docker Containers einfach Übertragbar gemacht werden.