

SDDC

Software Defined Data Center

Configfile

Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
17.10.15	1.00	Erstellung des Dokuments	Gruppe
19.10.15	1.01	Configfile Optionen	Silvan Adrian
07.11.15	1.02	JSON definition + Anpassungen auf neue Anforderungen	Fabian Binna

Inhaltsverzeichnis

1	Änderungshistorie	2
2	Einführung	4
2.1	Zweck	4
2.2	Gültigkeit	4
2.3	Übersicht	4
3	Configfile	4
3.1	Allgemein	4
3.1.1	JSON	4
3.2	Compute	4
3.2.1	Name/uuid	4
3.2.2	Memory	5
3.2.3	vCPU	5
3.2.4	Boot ISO	5
3.2.5	Boot Disk	5
3.2.6	Network	5
3.3	Storage	5
3.3.1	Name	5
3.3.2	Grösse	5
3.3.3	Typ	5
3.4	Network	6
3.4.1	OpenFlow	6
3.5	Syntax	6
3.5.1	Service	6
3.5.2	OrderedService	6

2 Einführung

2.1 Zweck

Dieses Dokument beschreibt das Configfile

2.2 Gültigkeit

Dieses Dokument ist während des ganzen Projekts gültig und wird laufend erweitert

2.3 Übersicht

Dieses Dokument soll eine Übersicht über die Inhalte des Configfiles geben.

3 Configfile

Das Configfile beschreibt einen Service möglichst abstrakt. Die Punkte Compute und Storage werden durch allgemeingültige Größen (S, M, L) beschrieben. Der eigentliche Service ist auf einem Image vorinstalliert. Im Configfile wird dieses Image referenziert und bei aufsetzen des Services auf dem Server installiert.

3.1 Allgemein

Da je nach Bedarf Compute, Storage oder Netzwerk zuerst erstellt werden muss wird dies über eine Art Workflow behandelt, welcher entscheidet welches Servicemodul zuerst erstellt werden soll.

3.1.1 JSON

Das Configfile wird in JSON geschrieben und wird auch so an die Generische API übergeben.

3.2 Compute

In Compute werden Memory, vCPU, Harddisk und Netzwerk (IP) zugewiesen. Es gibt noch einige mehr Funktionen, diese werden zu diesem Zeitpunkt aber noch nicht behandelt.

3.2.1 Name/uuid

Jeder Compute Instanz muss ein eindeutiger Name gegeben werden, dieser wird zu Beginn direkt in das Configfile eingefügt und zu einem späteren Zeitpunkt soll es möglich sein den Namen via Customer-Dashboard festzulegen. Bei libvirt wird ebenfalls eine uuid benötigt, welche eine eindeutige Identifizierung erlaubt -> wodurch das Auffinden und Löschen einer Instanz erleichtert werden soll

3.2.2 Memory

Memory wird entweder als fester Wert mitgegeben oder wird über die Instanzgrösse beim Cloud Anbieter vorbestimmt (micro,medium,large).

3.2.3 vCPU

vCPU's werden entweder als Wert mitgegeben oder durch den Cloud Anbieter vorgegeben (die Instanzgrösse).

3.2.4 Boot ISO

Je nach Anforderung muss der Pfad zu einer Boot ISO mitgegeben werden (für die Installation eines Betriebssystems)

3.2.5 Boot Disk

Ebenfalls muss noch eine Boot Disk übergeben werden, dies kann ein bereits bestehender Storage Pool, Volume oder eine Datei sein. Die Grösse wird hier durch die ausgewählte Disk/Datei bestimmt, bei Cloud Anbietern jedoch durch die Instanzgrösse.

3.2.6 Network

Beim Netzwerk kann eine IP oder MAC Adresse mitgegeben oder automatisch zugewiesen werden. Weitere Netzwerk Einstellungen werden über die SDN Library abgehandelt. Bei Cloud Anbietern wird hier ebenfalls direkt automatisch eine IP und MAC Adresse zugewiesen.

3.3 Storage

3.3.1 Name

Storage besitzt einen festen eindeutigen Namen über welchen der Pool angesprochen werden kann.

3.3.2 Grösse

Der Storage benötigt eine gewisse Grösse um erstellt zu werden oder falls es sich um eine Partition oder Datei handelt wird sie dadurch vorgegeben.

3.3.3 Typ

Der Storage kann sowohl lokaler als auch Netzwerk Speicher sein, hier wird zwischen verschiedenen Typen unterschieden. Z.B.: können NFS Storages eingebunden werden oder GlusterFS bzw. Sheepdog.

3.4 Network

3.4.1 OpenFlow

Netzwerkeinstellungen werden über OpenFlow gehandhabt, somit werden hier die OpenFlow spezifische Konfigurationen unterstützt. Zu diesem Zeitpunkt wird dies allerdings noch nicht behandelt.

3.5 Syntax

3.5.1 Service

```
{  
  id : <number>,  
  serviceName : <string>,  
  image : <string>,  
  
  compute : {  
    size : S | M | L  
  },  
  
  storage : {  
    size : S | M | L  
  },  
}
```

3.5.2 OrderedService

```
{  
  id : <number>,  
  serviceName : <string>,  
  
  identifiers : [ <string>, ... ]  
}
```