Deploying Solaris 11 with OpenStack and LDAP based Puppet

Thorsten Schlump
T-Systems International GmbH

Schlüsselworte
Solaris OpenStack LDAP Puppet

Einleitung
OpenStack ist eine freie Architektur für Cloud Computing. Es besteht aus zusammenhängenden Modulen mit denen sich ein Infrastructure-as-a-Service (IaaS) Servicemodell realisieren lässt. Hierfür verfolgt OpenStack einen generischen Ansatz um zum einen auf unterschiedlichen Systemen lauffähig zu sein, zum anderen diverse Einsatzmöglichkeiten wie Private/Public Cloud Angebote zu ermöglichen. Derzeit nehmen über 500 Firmen am OpenStack Projekt teil, die Nutzerzahl wächst rasant. Im Folgenden unsere Erfahrungen mit der Integration von OpenStack in das Deployment Modell Solaris @ T-Systems.

Vergleich aktuelles Deployment vs. OpenStack
Das aktuelle Deployment setzt auf dem Dynamic Toolset (DT) auf. Dies ist ein von T-Systems entwickeltes Tool zur Definition von Systemen mit folgenden Eigenschaften:

- plattformunabhängiger Ansatz
- Mandantenfähigkeit
- **vollständige** Konfigurationsmöglichkeit
- Verwendung durch **Admin**
- beinhaltet kein Plattformmanagement
- LDAP für User **und Konfigurationen**

Demgegenüber lassen sich einem OpenStack diese Eigenschaften zuschreiben:

- plattformunabhängiger Ansatz
- Mandantenfähigkeit
- **minimale** Konfigurationsmöglichkeit
- Verwendung durch **Kunde** möglich
- Definition von Kontingenten möglich
- LDAP für User möglich, Konfigurationen werden vom OpenStack selbst verwaltet

Die Unterschiede liegen in der Komplexität des DT. Eine durch den Kunden gesteuerte Self Service Cloud ist mit OpenStack besser realisierbar. Unser gesamtes Deployment ist LDAP basiert, weshalb das DT auch LDAP für User und Konfiguration nutzt. Dies ist beim derzeitigen OpenStack nicht der Fall, nur die Userverwaltung liegt im LDAP.

Puppe mit LDAP
Wir verwenden LDAP als sogenannten External Node Classifier (ENC). Dies bedeutet wir können LDAP als Sicherheitsebene für den Zugriff auf den Puppet Master verwenden, und Daten aus dem LDAP sind als Variablen für die Programmierung mit Puppet verfügbar.

Für die erste Sicherheitsebene muss der Domain Name des Full Qualified Domain Name (FQDN) im Puppet Master als gültig markiert sein. Ansonsten erhält der Client kein signiertes Zertifikat.

Für die zweite Sicherheitsebene wird der Domain Name des FQDN wird als Suchpfad im LDAP verwendet:

zz.ux.sys => dc=zz,dc=ux,dc=sys

Hier muss nun eine Konfiguration für den Hostnamen gefunden werden, nur dann wird ein Katalog erstellt.

Der Katalog enthält nicht nur Daten dieser Konfiguration, auch Referenzen sind möglich:

parentnode => cn=192_168_192_0,ou=profile, dc=zz,dc=ux,dc=adm:2

Diese Daten werden dem Katalog hinzugefügt. Letztendlich beinhaltet der Katalog folgende Daten:

Fakten vom System + Attribute aus dem LDAP = zur Verfügung stehenden Daten

Die Fakten vom System entsprechen dem IST Zustand, die Attribute aus dem LDAP dem SOLL Zustand.

Als Beispiel nehmen wir die primäre IP Adresse des Systems:

Fakt vom System:	ipaddress => 172.16.0.12
Attribut aus dem LDAP:	ipHostNumber = 172.16.0.4

Normalerweise müsste nun durch den Puppet Agenten die aktuelle IP Adresse mit den Daten aus dem LDAP überschrieben werden (SOLL auf IST Abgleich). Durch die dynamische IP Adressvergabe im OpenStack ist es aber die Regel, dass der IST Zustand der gültige ist. Dies ist ein Problem der fehlenden LDAP Anbindung vom OpenStack, das wir später erörtern werden.

Installation OpenStack

Für die Installation von OpenStack haben wir ein Puppet Modul geschrieben das in der Lage ist den Control Node sowie Compute Nodes zu installieren. Aktiviert und konfiguriert wird dieses Modul durch zwei zusätzliche LDAP Attribute in der Konfiguration:

puppetvar => mode=osm
puppetvar => osm=zzuxosm1

Der „mode“ Eintrag aktiviert das OpenStack Modul, der „osm“ Eintrag benennt den Namen des Control Nodes. In Abhängigkeit des „osm“ Eintrages werden nun folgenden OpenStack Komponenten installiert:

osm = = Hostname => Controller Node

- Keystone = Identity
- Glance = Image Service
- Nova = Compute
- Horizon = Dashboard
- Cinder = Block Storage
- Neutron = Networking

osm != Hostname => Compute Node

- Nova = Compute

Netzwerk Layout
Wir verwenden derzeit zwei Netzwerke für die Instanzen:

- Ein internes Netz für den Zugriff auf interne Dienste wie Puppet, LDAP etc.
- Ein externes Netz für den Zugriff von außen (auch das Default Gateway der Instanz)

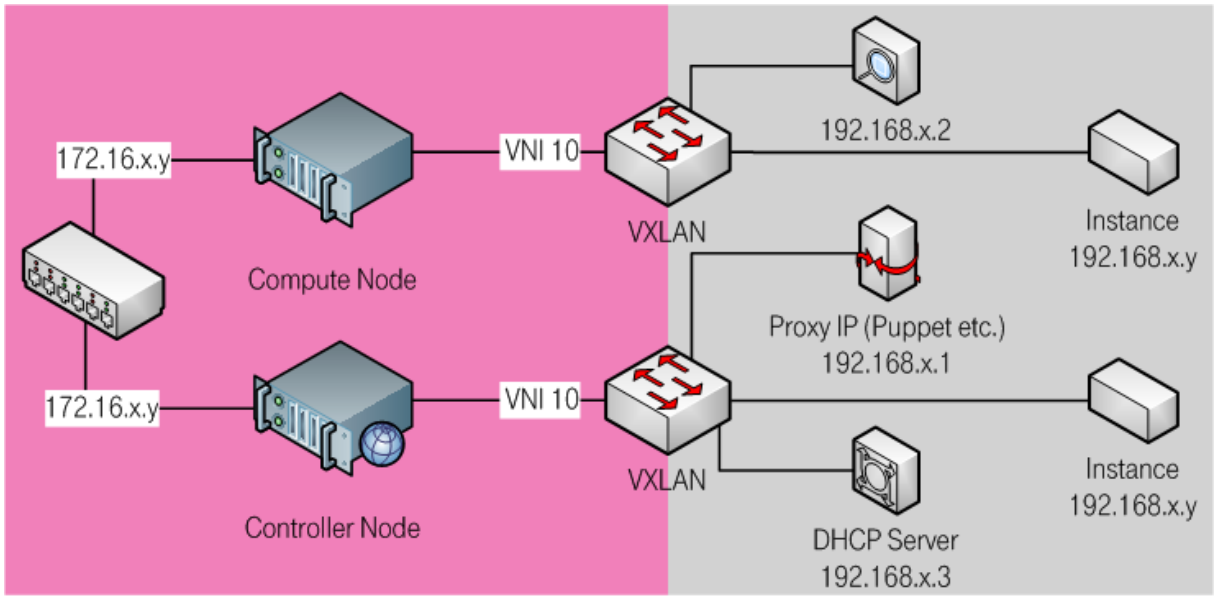

Abbildung 1: Internes Netz

VXLAN = Virtual Extensible LAN
VNI = VXLAN Network Identifier

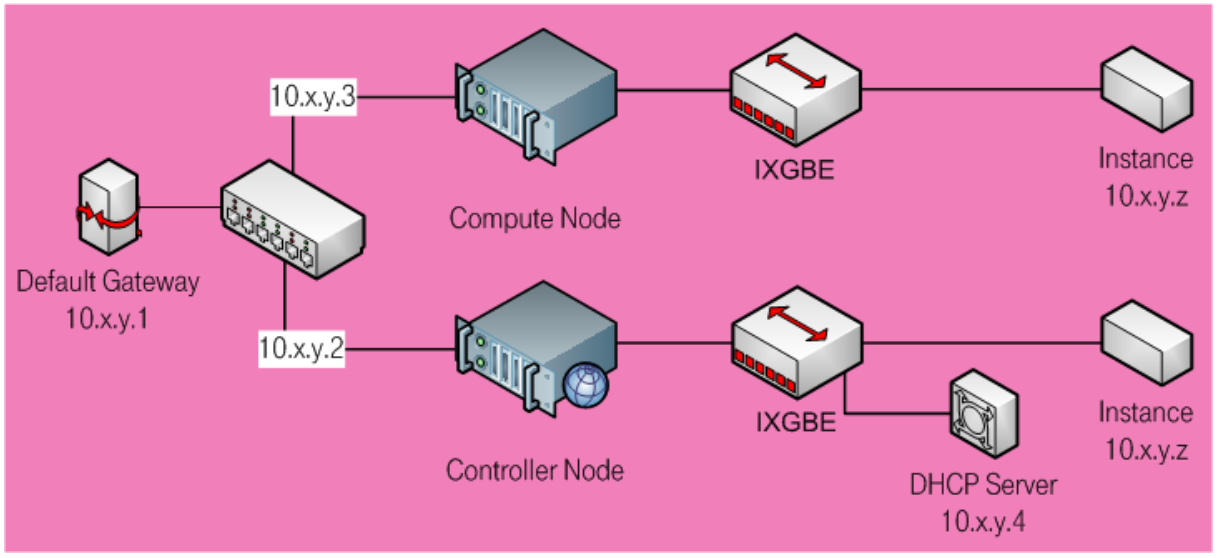

Abbildung 2: Externes Netz

Vom Image zum Unified Archive
Zunächst installieren wir eine Zone mit dem Standard Image über das entsprechende Puppet Modul. Anschließend erfolgt nur noch die Installation eines zusätzlichen IPS Paketes „openstack/ngz“ mit folgender Funktion:

1. Service zur Anpassung der Zone beim Start
    a. Modifikation der Hosttabelle damit IP zum FQDN passt
    b. Löschen aller Puppet Keys / Konfigurationen die nicht dem FQDN entsprechen
    c. Prüfung Routing im internen Netz (Erreichbarkeit Puppet Master)
    d. Verifikation gültiger „root“ Benutzer (Cron)
2. Service um den Puppet Agent vom obigem Service abhängig zu machen

Danach muss nur noch das Unified Archive erstellt werden:

```
# archiveadm create -z zzux0000 -e /var/tmp/ngz-sol11cpu0715-sparc.uar
```

Dieses Image wird anschließend im Glance konfiguriert und steht damit zur Installation von OpenStack Instanzen zur Verfügung.

OpenStack Instanzen im LDAP
Unsere erste inzwischen überholte Variante basiert auf einem Skript, das periodisch auf jedem OpenStack Server gestartet wird und folgende Aktionen durchführt:

1. Auflistung der aktiven Zonen
2. Prüfung ob bereits bearbeitet (lokale Datei und danach Anfrage an LDAP)
3. Zusammenstellung Daten der aktuellen Zone (Name, IP etc.)
4. Erstellung Hostkonfiguration über SSH Automatisierungsschnittstelle des DT

Die Verteilung des Skriptes und der SSH Schlüssel erfolgt über ein IPS Paket „openstack/gz“ im Rahmen der OpenStack Installation.

Das Hauptproblem dieser Variante ist, dass Änderungen an den LDAP Daten zwar realisierbar sind, aber das Löschen der LDAP Konfigurationen in Umgebungen mit mehreren Compute Nodes schwierig ist: Nur weil ein System nicht mehr auf Node A aktiv ist heißt nicht, dass es nicht auf einem anderen Node aktiv sein kann - in so einem Fall ist das Löschen der LDAP Konfiguration gefährlich.

Die Idee diese Funktionalität in der Instanz selber zu implementieren konnte zunächst nicht realisiert werden: Der SSH Zugriff auf das DT limitiert, da er einen Schlüsselaustausch bedingt - dies ist aus mehreren Gründen unvorteilhaft (unter anderem Sicherheit). Seit der neusten Version des DT gibt es aber eine SOAP Schnittstelle (XML über HTTPS), die für den nicht interaktiven Zugriff auf das DT eine höhere Flexibilität bietet.

Somit können wir nun die LDAP Synchronisierung aus der Zone heraus tätigen. Es existiert bereits ein Skript, das die OpenStack Instanz beim Start anpasst. Dieses Skript wird nun erweitert:

- Beim Start wird über die SOAP Schnittstelle die LDAP Konfiguration angelegt
- Beim Stopp wird über die SOAP Schnittstelle die LDAP Konfiguration gelöscht

Damit ist auch das Problem der Änderungen gelöst, da eine Änderung der relevanten Konfigurationsdaten einen Neustart der Instanz bedingt.

Workflow Deployment mit OpenStack
Die folgenden vier Stufen beschreiben den Lebenszyklus einer OpenStack Instanz:

1. Instanz wird in Horizon erstellt/gestartet
2. Service prüft die lokale Konfiguration, erstellt die LDAP Konfiguration, und der Puppet Agent übernimmt die weitere Anpassung
3. Instanz wird in Horizon gelöscht/gestoppt
4. Service löscht die LDAP Konfiguration

Von Stufe vier kann der Zyklus wieder mit Stufe 1 von Neuem beginnen.

Ausblick und Fazit
Folgende Punkte werden derzeit im Proof of Concept bearbeitet:

- Zentraler Speicherort für Systemkonfigurationen ist LDAP. Daher ist bei Systemen ohne native LDAP Anbindung eine Schnittstelle nötig.
- Die derzeitige Skript Schnittstelle funktioniert gut, ist aber nicht perfekt.
- Optimal wäre es wenn OpenStack direkt LDAP für Systemkonfigurationen nutzen könnte. Ein erster Ansatz ist das Modul „nova.network.ldapdns“.
- Ein anderer Ansatz basiert auf einer direkten Steuerung durch OpenStack. Ein Impuls löst über eine Schnittstelle eine Aktion im LDAP aus. Diverse Ansätze wie Logging, Messaging und Telemetrie werden derzeit evaluiert.

Für den Einsatz von OpenStack bei T-Systems sind diese Punkte relevant:

- Mit OpenStack können wir dem Kunden eine Managed Cloud anbieten
- Diese Cloud kann vom Kunden im Rahmen der bestellten Kontingente selbst verwaltet werden
- Es wird das Standard Solaris Image verwendet, daher kein großer Mehraufwand in der Bereitstellung
- Da OpenStack ein generischer Ansatz ist sind Teile auch für andere Derivate nutzbar und erlauben den Austausch von Komponenten

Kontaktadresse:
Thorsten Schlump
T-Systems International GmbH
Heerdter Lohweg 35
D-40549 Düsseldorf

Telefon: +49-(0)-211-50082283
Mobil: +49-(0)-170-5734555
E-Mail Thorsten.Schlump@t-systems.com
Internet: www.t-systems.com