Solaris 11 Deployment mit Puppet

Thomas Rübensaal
T-Systems International GmbH
Bamberg

Schlüsselworte
Solaris 11, Puppet, Deployment, Installation, Konfiguration, Konfigurationsmanagement

Einleitung
Mit der Einführung von Solaris 11 ergeben sich deutliche Unterschiede bei der Installation. Jumpstart für Solaris 10 wird durch den „Automated Installer“ (AI) ersetzt. T-Systems nutzt bisher für das Customizing von Solaris 10 das Tool Sysconf, welches nach der Jumpstart Phase ausgeführt wird. Sysconf hat allerdings den Nachteil, dass die Ausführung sequentiell erfolgt und ein Server Deployment somit relativ viel Zeit benötigt.
Eine Anpassung von Sysconf an Solaris 11 hätte viel Arbeit erfordert, so dass die Wahl des Tools für das Deployment von Solaris 11 ergebnisoffen diskutiert werden konnte.
Die endgültige Wahl fiel dann auf das Tool „Puppet“, das zum einen unseren Anforderungen für ein Service-Provider-Ready Deployment erfüllte als auch durch seinen parallelen Ansatz in Sachen Geschwindigkeit überzeugen konnte.

Puppet – Was ist das?

- Puppet ist ein flexibles, anpassbares Framework, um viele sich wiederholende Tasks, die von einem System Administrator täglich manuell ausgeführt werden müssten, zu automatisieren.
- Es basiert auf Programmiersprache Ruby und ist Open Source.
- Als eine zustandsbeschreibende Vorgehensweise wird ein gewünschter Status definiert.
- Wenn der gewünschte Status einmal definiert ist, installiert Puppet automatisch alle notwendigen Pakete und startet alle zugehörigen Services. Weiterhin wird bei jedem neuen Puppet Run sichergestellt, dass der definierte Zustand vorhanden ist.
- Puppet ist für die meisten UNIX ähnlichen Betriebssysteme (AIX, HP-UX, RedHat, Suse, Solaris, …) und für Microsoft Windows verfügbar.
- Puppet kann sicher und beliebig oft ausgeführt werden, wobei Puppet nur dann Änderungen auf dem Client durchführt, wenn der aktuelle System Status vom definierten System Status abweicht.
- Puppet unterstützt LDAP. Alle Konfigurationsdaten können in einer LDAP Datenbank gespeichert werden. Nur der Puppet Master Server benötigt einen Zugriff auf den LDAP. Von daher benötigen die Puppet Clients keinen zwingenden LDAP Zugriff.

Puppet Run – Wie funktioniert es?

- Auf dem Client wird ein Puppet Run angestoßen.
- Der Client sendet normierte Daten über sich selbst zum Puppet Master Server.
- Der Puppet Master Server liest diese Daten (Facts) vom Client als auch optional Konfigurationsdaten vom LDAP Server und erstellt daraus einen Katalog. Dieser Katalog spezifiziert, wie der Client konfiguriert werden soll.

- Der Katalog wird nun auf dem Client ausgeführt. Der Client meldet dem Puppet Master zurück, wenn die Konfiguration komplett ausgeführt wurde.
- Über die API von Puppet können optional auch Daten zu Third Party Tools gesendet werden.

Wie sicher ist Puppet?

- Puppet Agenten (Clients) benötigen keinen Zugriff auf Manifest Dateien, und können auch Konfigurationsinformationen außerhalb ihres eigenen Katalogs nicht lesen.
- Die Kommunikation zwischen dem Puppet Master Server und den Clients erfolgt verschlüsselt (SSL Verschlüsselung des kompletten Datenverkehrs).
- Per Default wird TCP Port 8139 (Server → Client = Puppet Kick) und 8140 (Client → Server) verwendet.
- Puppet wurde von der Group IT Security der Deutschen Telekom als geeignetes Tool für das Konfigurationsmanagement eingestuft und kann somit als sehr sicher betrachtet werden.

Puppet Einsatz bei T-Systems

T-Systems Tool-Vergleich Solaris 10 ←→ Solaris 11

- Für Solaris 10 wurde "Sysconf" zur Client Konfiguration genutzt. Weger der seriellen Ausführung von Sysconf sind Deploymentvorgänge hier sehr langsam.
- Für Solaris 11 findet "Puppet" Verwendung. Auf Grund paralleler Ausführbarkeit erfolgt das Deployment hier deutlich schneller.
- Bisher mussten für Solaris 10 eigene Jumpstart- und Sysconf-Server an jeder größeren Lokation (verteilt auf ca. 10 Rechenzentren) vorgehalten werden.
- Für Solaris 11 ist nur noch ein zentraler Automated Installer und ein zentraler Puppet Master vorhanden. Für die Anbindung von weiteren Rechenzentren kommen Proxy Caches zum Einsatz.
- Weiterhin ist eine LDAP Server Infrastruktur sowie ein eigen-entwickeltes web-basierendes Konfigurationstool (Dynamic Toolset) vorhanden, das sowohl für Solaris 10 als auch Solaris 11 genutzt wird.

T-Systems Solaris 11 Architektur Überblick

- Es existiert eine LDAP Server Infrastruktur mit einem zentralen LDAP Master und lokalen LDAP Master an jeder Lokation (inkl. Master – Master Replikation). Des weiteren sind LDAP Slave Server für lokale Redundanzen vorhanden.
- Ebenso gibt es einen zentralen Automated Installer, Solaris 11.x Repositories und Puppet Master. (Server Layer, nur an einer Lokation)
- Die remote Lokationen sind mit je einem Apache Proxy Cache und Netcat zur zentralen Lokation angebunden. (Proxy Layer)
- Solaris 11 Clients (an jeder Lokation) verwenden den lokalen Proxy Layer (Client Layer)
- Für den PXE-Boot von x86 Clients werden bestehende lokale Solaris 10 Installserver (DHCP) mitgenutzt.
- Für die LDAP Funktionalität der Solaris Clients werden die LDAP Server im jeweiligen RZ genutzt.

- Das Dynamic Toolset (ein eigenentwickeltes zentrales Konfigurationstool mit Weboberfläche) dient zur Erfassung und Verwaltung der Client-Daten.

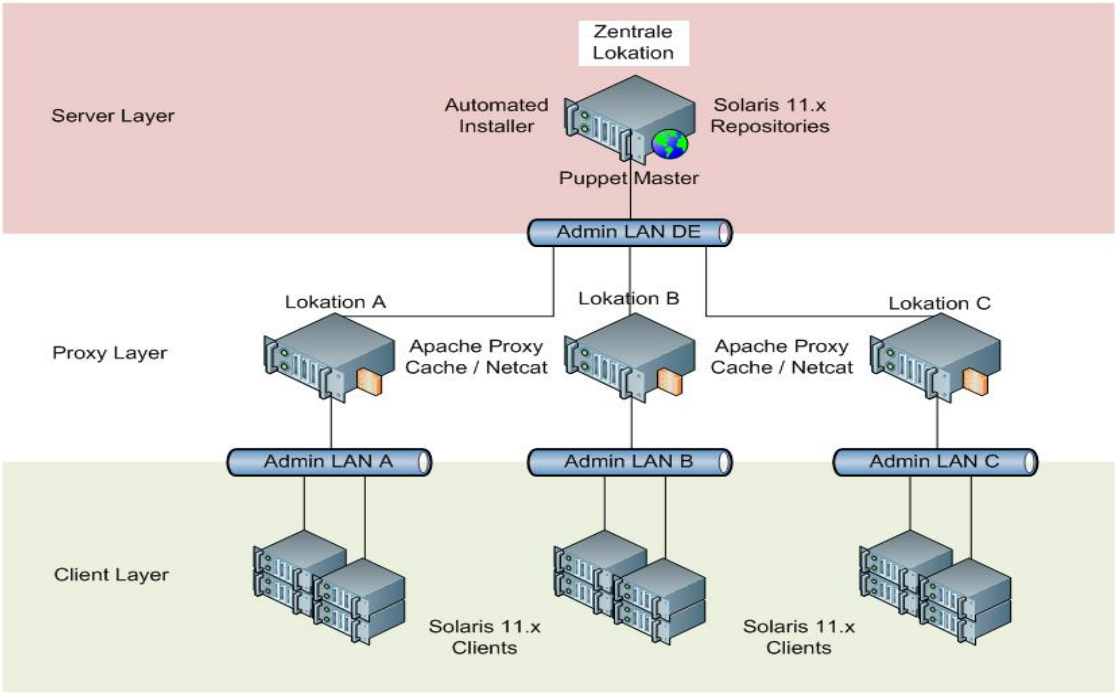

Abb. 1: T-Systems Solaris 11 Infrastruktur

Ablauf einer Solaris 11 Installation

- Solaris 11 Client wird mit einem Web basierenden T-Systems internen Konfigurationstool (Dynamic Toolset) konfiguriert. Alle relevanten Konfigurationsdaten der Solaris Clients (z.B. Rechnername, MAC Adresse, IPs, Filesysteme, definierte Puppet Module,…) werden hierbei erfasst.
- Die Solaris Client Konfiguration wird im Hintergrund zentral in einer LDAP Datenbank (zentraler LDAP Master) gespeichert. Weiterhin wird der zentrale WAN-Boot-Server (für SPARC Clients) bzw. ein lokaler DHCP Server (auf bestehender Solaris 10 Infrastruktur, für x86 Clients) konfiguriert.
- Der WAN-Boot (SPARC) des Solaris 11 Clients erfolgt über den lokalen Apache Proxy Cache. Dieser holt sich ggf. nicht vorhandene oder veraltete Daten vom zentralen Automated Installer.
- Beim PXE-Boot (x86) wird der DHCP-Request von einem lokal vorhandenen Solaris 10 Installationsserver beantwortet. Nach Laden des Miniroots erfolgt die eigentliche Installation über den lokalen Apache Proxy Cache (analog zu SPARC).
- Anschließend erfolgt die Automated Installer Installation (inklusive Installation des Puppet Agenten) wieder über den lokalen Apache Proxy Cache als Gateway zum zentralen Repository.

- Der Puppet Agent startet als letzter Service und konfiguriert sich selbst. Die Kommunikation der lokalen Puppet Clients erfolgt über einen Netcat als Gateway zum zentralen Puppet Master.
- Nun folgt der initiale Puppet Run (alle definierten Puppet Module werden ausgeführt), der bei Erfolg das System auch rebootet.
- Optional kann nach dem Neustart automatisiert ein zweiter selektiver Puppet Run zur Bereitstellung von “Non Global Zones” und / oder “ORACLE VM for SPARC Guests” erfolgen.
- Alle zusätzlichen Konfigurationsläufe werden vom Administrator bei Bedarf initiiert (wenn zum Beispiel eine weitere nicht-globale Zone im LDAP definiert wird oder andere Attribute geändert werden).

Verfügbare Repositories
- Solaris (Oracle Solaris 11.x full repository)
- HA-Cluster (optional, für Oracle Cluster Installationen)
- tsys (für T-Systems-eigene Pakete)
- Symantec (optional, für Symantec Pakete)

Release Plan (Solaris Versionierung für IPS Pakete und Puppet Module)
- Zwei Releases im Jahr Pflicht
- Zwei weitere Releases im Jahr optional
- Releases orientieren sich an den Oracle CPUs

Weitere Puppet Besonderheiten bei T-Systems
- Als Load Balancer für den Puppet Master kommt Passenger zum Einsatz.
- Es wurden verschiedene Umgebungen für Solaris 11 Repositories und Puppet Module definiert (analog zu Release –Plan, Umgebungen für Test und Produktion)
- Erweiterung der Parent Node Funktionalität um die Fähigkeit, mehrere Parent Nodes in definierter Reihenfolge einzubinden (Definition von Puppet Modulen in verschiedenen Hierarchiestufen, z.B. je Subnetz oder je RZ)
- Installation erfolgt in mehreren aufeinander aufbauenden Phasen
- Eigene Ressource-Typen für spezielle Solaris 11Anforderungen wurden erstellt (z.B. für Solaris 11 Netzwerkkonfiguration)

Ausblick bei T-Systems
Sobald die Solaris 11 Durchdringung zukünftig hinreichend ist, können die Deployment Umgebungen für Solaris 10 lokationsabhängig zurückgefahren bzw. eingestellt werden. Alternativ kann die Bereitstellung des Jumpstart Dienstes für Solaris 10 auch unter Solaris 11 erfolgen. Dies könnte dann auch einen Wechsel von Sysconf zu Puppet für Solaris 10 zur Folge haben.

Kontaktadresse:
Thomas Rübensaal
T-Systems International GmbH
Gutenbergstraße 13
D-96050 Bamberg

Telefon: +49 (0) 951 1336-5427
Fax: +49 (0) 391 580228-475
E-Mail thomas.ruebensaal@t-systems.com
Internet: http://www.t-systems.com