Konzept neue Plattform: Basics

Dokumentenname	PFM_Konzept_Basics
Autor	Thomas Bader
Status	Erster Entwurf, bereit zur Diskussion
Letzte Änderung	20.09.2015

Übersicht

Die Dienste werden mit virtualisierten Maschinen bereitgestellt. Dazu werden zwei physische Server als Cluster konfiguriert.

Die Konfiguration der einzelnen Services bzw. virtuellen Maschinen wird gesondert dokumentiert. Das vorliegende Dokument dient der Konzeptionierung und einer Übersicht über die Infrastruktur der Virtualisierung.

Betriebssysteme

RHEL wird für die beiden Cluster-Nodes bzw. Hypervisoren verwendet (**TODO**: Sponsoring bei Redhat anfragen). Der Cluster-Stack von Redhat ist sehr weit fortgeschritten. Da es sich um Open Source Komponenten handelt, kann er zwar auch auf anderen Betriebssystemen verwendet werden - RHEL hat jedoch langen Support (RHEL7 wurde in 2014 released und ist bis 2024 supported). Der Life-Cycle der Distribution der Hypervisoren sollte denjenigen der Hardware übersteigen. Upgrades der Hypervisoren sind sehr zeitaufwendig, da das Clustering besonders viele Tests benötigt. Er soll deshalb nur stattfinden, wenn die Hardware ersetzt wird.

Debian wird für die restlichen virtuellen Maschinen eingesetzt.

Komponenten des Clustering

Die Konfiguration des Clusters erlaubt den Failover von virtuellen Maschinen sowie der Storage. Dazu müssen unterschiedliche Komponenten konfiguriert werden.

cman

cman ist der Cluster-Manager und dient hauptsächlich als Quorum-Provider. Er prüft ob die Voraussetzungen für den Betrieb gegeben sind und start/stoppt den Cluster.

Bei trash.net sind weniger als 3 Nodes im Cluster. Deshalb kann der Betrieb nicht per Quorum realisiert werden. Der Cluster-Manager wird deshalb den Cluster ohne Prüfung von Vorbedingungen starten.

corosync

Durch corosync wird die Basisfunktionalität des Clusters realisiert. Per Multicast tauschen die Nodes Heartbeats untereinander aus und individuelle Services des Clusters werden automatisch gestoppt oder gestartet.

Der Austausch der Heartbeats ist kritisch. Wenn dieser nicht sichergestellt ist und sich beide Cluster-Nodes nicht mehr „sehen“, dann werden beide Nodes versuchen sämtliche Dienste bei sich zu starten und zu failovern. Dies führt dazu, dass beide Nodes alle virtuellen Maschinen bei sich lokal starten, diese parallel laufen und auf die Storage schreiben, und anschliessend keine Zusammenführung der Daten mehr möglich ist. Um dies zu verhindern sind im Cluster zwei Massnahmen implementiert:

1. Die Heartbeats werden zusätzlich auch per RS232-Link übertragen, um bei Problemen mit den Netzwerkinterfaces keinen Failover im Cluster auszulösen
2. Mit Fencing können die Nodes sich gegenseitig abschalten, sofern nicht automatisch behebbare Probleme auftreten

corosync führt Controlled Process Groups (CPG) ein. Innerhalb einer CPG werden Nachrichten immer in geordneter Reihenfolge ausgetauscht. Dazu wird ein Totem in einem Ring ausgetauscht, jede Node darf nur Locks anfordern, wenn sie das Totem bei sich hat. Da im hier beschriebenen Cluster nur zwei Nodes sind, wird das Totem zwischen den Nodes hin und her transferiert.

Rgmanager / Pacemaker

Die beiden Komponenten Rgmanager (Redhat-Cluster Stack) und Pacemaker (Linux-HA) stellen das Ressourcen-Management des Clusters bereit. In ihnen sind die Services des Clusters hinterlegt (z.B. welche virtuelle Maschinen auf welcher Node läuft). Durch corosync festgestellte Änderungen am Cluster-Membership wird an den Resource-Manager signalisiert. Dieser prüft ob Services gestartet oder beendet werden müssen.

Welcher Resource-Manager eingesetzt wird ist zu definieren.

DRBD

Mit DRBD werden drei Block-Devices als Mirror zwischen den Maschinen synchronisiert:

- Ein Mirror dient als Shared-Storage um die Konfiguration/Skripte des Clusters abzulegen
- Ein Mirror dient als Basis für Clustered-LVM, um eine Volume-Group für virtuelle Maschinen der Node 1 abzulegen
- Ein Mirror dient als Basis für Clustered-LVM, um eine Volume-Group für virtuelle Maschinen der Node 2 abzulegen

Jede Node hat jeweils ihre eigene Volume-Group aktiv. Sobald eine Node ausfällt, wechselt übernimmt sie VG und virtuelle Maschinen der anderen Node.

DLM

Der Distributed Lock Manager stellt das Locking im Cluster bereit. Bei Shared-Ressourcen (Clustered-LVM und GFS2) muss sichergestellt werden, dass keine parallelen Zugriffe auf dieselben Daten stattfinden. DRBD synchronisiert lediglich die Datenblöcke der Disks und stellt nicht sicher, ob nicht Prozesse auf beiden Nodes parallen dieselben Datenblöcke schreiben wollen (was zu inkonsistenten Daten führt).

Durch die von corosync eingeführten CPG können für eine Ressource niemals zwei Locks parallel vergeben sein.

Mit DLM werden im Cluster zwei Dinge sichergestellt:

- Mit GFS2 kann ein Dateisystem auf beiden Nodes mounted sein und die Locks im DLM stellen sicher, dass nicht beide Nodes auf dieselben Dateien schreibend zugreifen
- Bei Clustered LVM stellt DLM sicher, dass eine Volume-Group nur auf einem System gleichzeitig aktiv ist

GFS2

Mit GFS2 wird der DRBD-Mirror der Shared-Storage auf beiden Nodes auf /shared mounted. Dies dient dazu, dass die beiden Nodes Konfigurationsdateien, Skripte und Installationsimages teilen können.

Clustered LVM

Im Cluster werden zwei Volume-Groups realisiert. Jede Node verfügt über eine eigene VG und das darunterliegende Physical-Volume wird per DRBD repliziert.

Betrieb des Clusters

Start des Clusters

Bei einem Start des Clusters laufen folgende Schritte ab:

1. Über cman wird das Quorum des Clusters festgestellt. Es sind nur zwei Nodes im Cluster, deshalb beschränkt sich dies darauf die Erreichbarkeit der jeweils anderen Node zu prüfen. Eine Node hat eine höhere Priorität im Clusters - sofern diese die andere Node nicht sieht, schaltet sie diese via Fencing ab. Dies verhinder Split-Brain Situationen.
2. cman signalisiert corosync, dass der Cluster in Betrieb gehen kann und die Nodes beginnen den Austausch von Heartbeats
3. corosync signalisiert dem Resource-Manager eine Änderung im Cluster-Membership
4. Resource-Manager nimmt anhand seiner lokalen Konfiguration alle Services in Betrieb - DRBD-Blockdevice aktivieren, CLVM-VG aktivieren, virtuelle Maschinen starten
5. CLVM holt sich beim DLM Locks und sperrt die jeweils eigene Volume-Group
6. GFS2 holt sich bei Bedarf (d.H. bei jedem Schreibzugriff) Locks vom DLM und gibt diese bei Ende der Operation wieder frei

Der Resource-Manager weiss, welcher Service auf welcher Node läuft. Sofern beide Nodes aktiv sind, startet jede Node jeweils ihre eigenen Services. Falls die beiden Nodes einander nicht erreichen

können, startet die übrig gebliebende Node alle Services bei sich lokal.

Das Fencing ist deshalb wichtig, da bei einem Verbindungsproblem zwischen den Nodes verhindert werden muss, dass jede Node bei sich selber alle Services lokal startet.

Ausfall einer Node

Beim Ausfall einer Node finden folgende Schritte statt:

1. Falls die innerhalb von x Heartbeats wieder online ist, wird der Cluster ohne Veränderung weiterbetrieben
2. Falls die Node innerhalb von x Heartbeats immer noch offline ist, prüft cman ob ein Quorum vorhanden ist (nicht notwendig bei Clustern mit zwei Nodes)
3. parallel dazu wird DLM signalisiert, dass keine Locks mehr vergeben werden dürfen
4. sobald das Fencing durchgeführt wurde, wird DLM signalisiert, wieder Locks vergeben zu dürfen
5. ebenso wird dem Resource-Manager signalisiert, alle lokalen Services zu prüfen
6. Resource-Manager startet auf Node lokal alle Services, welche im Cluster nicht aktiv sind

Wenn die ausgefallene Node wird verfügbar ist, prüft cman erneut das Quorum, nimmt die Node wieder in die Verteilung des Totem auf und signalisiert dem Resource-Manager die Prüfung der Resources. Dieser wird so konfiguriert, dass er Services nicht automatisch migriert. Nach dem Ausfall einer Node soll zuerst die korrekte Funktion dieser manuell verifiziert werden und die Migration der Services manuell vorgenommen werden.

Virtualisierung

Die Virtualisierung wird per KVM realisiert. Jede VM erhält ein Logical-Volume und wird an die Bridge verbunden, welche Zugang zum relevanten Netz bereitstellt.

Alternativ können auf einem Cluster in einer lokalen Storage (/local) auch virtuelle Maschinen abgelegt werden, welche vom Fail-Over ausgeschlossen sind.

Speicherplatz der Mitglieder

Die Daten der Mitglieder werden auf einem Fileserver abgelegt, welcher die Homeverzeichnisse der Benutzer per NFS verteilt. Dadurch können die Mitglieder auf allen Maschinen ihr Homeverzeichnis nutzen. Dies bedeutet:

- Auf einem Shell-Server editierte Daten sind auch auf dem Webserver verfügbar
- Mails können auf beiden Shell-Servern gelesen werden (setzt Maildir voraus)
- Webseiten können auf mehreren Webservern genutzt werden (bei Migrationen können z.B. die Webseiten stückweise auf den Webserver mit den neueren Komponenten migriert werden)

Benutzerverwaltung

Sämtliche virtuelle Maschinen werden so konfiguriert, dass sie über PAM die Benutzerdaten ab dem LDAP-Verzeichnis beziehen. Die Authentifizierung erfolgt über Kerberos.

Der Zugriff auf einzelne virtuelle Maschinen wird über die Gruppenmitgliedschaft kontrolliert. Für die Systemadministration wird so bspw. festgelegt, welcher Administrator auf welche der virtuellen Maschinen einloggen darf.

Die Verwaltung des LDAP-Verzeichnis wird mit dem LDAP Account Manager (LAM, https://www.ldap-account-manager.org/) vorgenommen.

Auf LDAP werden auch die Maildomains bzw. Aliases verwalten. Der Mailserver bezieht die Angaben zur Zustellung von Mails aus dem Verzeichnis.

Systemadministration

Die /etc-Verzeichnisse der Installationen werden mit etckeeper (http://joeyh.name/code/etckeeper/) in ein git-Repository eingecheckt. Jeder Administrator muss seine Änderungen zusammen mit einem Kommentar einchecken.

Ein Monitoring-Server mit OMD bzw. check_mk (http://omdistro.org/, http://mathias-kettner.de/checkmk.html) wird dazu verwendet, die Verfügbarkeit sämtlicher Dienste zu prüfen und zu alarmieren.

Die Shellserver dienen den Administratoren als Jumphosts. Der direkte Login per SSH auf die restlichen VMs mit Diensten ist nur von ihnen aus möglich. Um im Falle eines Ausfalls der Shellserver zumindest Zugriff auf die Hypervisor-Hosts zu bekommen, wird auf ihrer host-based Firewall Port-Knocking implementiert (http://www.shorewall.net/PortKnocking.html).

Security

Jeder Host ist mit einer host-based Firewall ausgestattet (Shorewall, http://www.shorewall.net/).

Die Logs der Systeme werden zentral auf einem Logserver gesammelt (Syslog per TCP). Auf jeder virtuellen Maschine wird die Aufbewahrungsdauer für Logs verkürzt. Stattdessen dient der Logserver der längeren Aufbewahrung der Logs. Der Zugriff auf den Logserver wird eingeschränkt.

Die Aktionen der Systemadministratoren werden, sobald ein Wechsel zu einem privilegierten Account stattgefunden hat, aufgezeichnet. Bevorzugt soll sudo verwendet werden, welches sämtliche Kommandos loggt. Sofern eine Shell als privilegierter Benutzer benötigt wird, dann werden die abgesetzten Kommandos aufgezeichnet.

Netzwerkzonierung

Die Infrastruktur verfügt über eine rote Zone, das Internet. Alle Server, welche von aussen erreichbar sein müssen, sind mit einem Interface in die Zone verbunden.

Die grüne Zone ist das private Netz von trash.net. Mit ihr werden virtuelle Maschinen verbunden, welche kritische Dienstleistungen (z.B. LDAP, Fileserver) bereitstellen und keinen Zugriff ins Internet benötigen.

Die virtuellen Maschinen von trash.net benötigen Zugriff auf LDAP, Fileserver und andere. Deshalb sind einige Maschinen (diejenigen, welche gleichzeitig Zugriff auf rot und grün benötigen) Dual-Homed an beide Zonen verbunden.

Für Mitglieder und Projekte ist die blaue Zone vorgesehen. Mit ihr können private Netze zwischen virtuellen Maschinen realisiert werden. Maschinen von Mitgliedern können, analog zu trash.net, nur mit der blauen Zone verbunden sein oder Dual-Homed rot/blau sein. Pro Projekt/Mitglied wird in der blauen Zone jeweils ein eigenes VLAN erstellt.

Die Nutzung der blauen Zone ist optional. Die virtuelle Maschine eines Projekts kann auch alleine mit der roten Zone verbunden sein. Ein eigenes VLAN in der blauen Zone wird nur zugewiesen, wenn das jeweilige Projekt die Voraussetzungen dafür erfüllt:

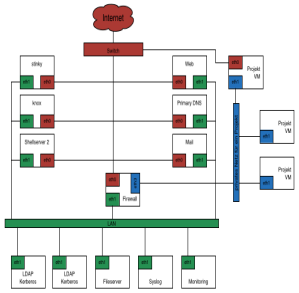
- Wenn ein Projekt besonders viele VM benötigen sollte, dann ist die Vorgabe, dass eine virtuelle Maschine im Internet als Reverse Proxy o.ä. betrieben wird, damit die VM mit privaten IP-Adressen adressiert werden können
- Wenn ein Projekt eine Datensynchronisation o.ä. zwischen virtuellen Maschinen benötigt, dann soll diese über die blaue Zone realisiert werden. Hohes Trafficaufkommen zwischen virtuellen Maschinen soll nicht über die rote Zone transferiert werden.

Neben der roten, grünen und blauen Zone existieren noch weitere Zonen, welche exklusiv für die Verbindung zwischen den Hypervisor-Hosts vorgesehen sind:

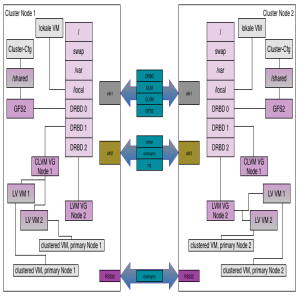
- grau als Storage-Netz
- gelb als Back-Channel / Cluster-Management
- orange für IPMI

Details der Konfiguration

Zonierung



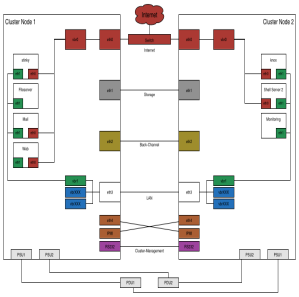
Storage und Cluster-Komponenten



Netzwerk

Der Uplink zum Internet wird über einen Switch bereitgestellt. Er stellt deshalb einen single-point of failure. Weitere Verbindungen sind als Crosslink zwischen den Nodes realisiert. Ein Link dient der Synchronisation der Storage und ein Link dient dem Cluster-Management. Ein weiterer Link ist als 802.1q-Trunk umgesetzt und erlaubt den Aufbau von privaten Netzen zwischen virtuellen Maschinen. Die IPMI-Interfaces der beiden Server werden über Kreuz mit Crosslinks verbunden.

Zone	Interface	Typ
Internet (rot)	eth0	Link der beiden Cluster-Nodes an den Switch im Internet
Storage (grau)	eth1	Cross-Link zwischen beiden Nodes für Replikation der Storage
Back-Channel (gelb)	eth2	Cross-Link zwischen beiden Nodes für das Cluster-Management
LAN	eth3	Cross-Link zwischen den beiden Nodes für private Netze, als 802.1q-Trunk
trash.net private net (grün)	vbr1, eth3.1	VLAN als privates Netz von trash.net
beliebige Netze	vbrXXX, eth3.XXX	beliebige private Netze für Kunden
IPMI	eth4	Cross-Link, jede Node zum IMPI der anderen Node
RS232	ttyS0	serieller Cross-Link, für Cluster-Management



Storage

Pro Node wird ein RAID-Array auf lokalen Disks betrieben. Diese werden wie folgt partitioniert:

Partition	Filesystem	Mountpoint	Nutzung	Replikation
1	etx4	/	Root-Filesystem der Node	n
2	swap	-	Swap pro Node	n
4	ext4	/var	var-Filesystem der Node	n
5	ext4	/local	lokale Datastorage der Node	n
6	DRBD	-	Shared-Storage zwischen den Nodes	y

Partition	Filesystem	Mountpoint	Nutzung	Replikation
7	DRBD	-	PV für die Volume-Gruppe der Node 1	y
8	DRBD	-	PV für die Volume-Gruppe der Node 2	y

Über DRBD werden die folgenden Mirrors eingerichtet:

Mirror	Partition	Filesystem	Mountpoint	Nutzung
0	6	GFS2	/shared	Shared-Storage zwischen den Nodes
1	7	Clustered-LVM	-	PV für die Volume-Gruppe der Node 1
2	8	Clustered-LVM	-	PV für die Volume-Gruppe der Node 2

Virtuelle Maschinen

Host	OS	Nutzung	Cluster Node	Failover
stinky	?	Shell-Server	Node 1	n
tbd	?	Shell-Server	Node 2	n
knox	?	secondary DNS	Node 2	n
tbd	?	primary DNS	Node 1	n
Webserver	?	Webserver	Node 1	y
Mail	?	Mailserver (SMTP, POP, IMAP)	Node 1	y
Fileserver	?	NFS-Server	Node 1	y
Monitoring	?	Node für Monitoring	Node 2	y
LDAP/Kerberos Primary	?	Authentifizierung und Authorisierung für alle Nodes	Node 1	n
LDAP/Kerberos Secondary	?	Authentifizierung und Authorisierung für alle Nodes	Node 2	n
Syslog	?	zentraler Syslog-Server	Node 1	y

Anmerkung: nicht alle virtuellen Maschinen sind mit einem Failover ausgestattet. Sofern ein Dienst bereits auf Applikationsebene redundant ist (DNS), dann muss kein Failover auf Ebene der Virtualisierung realisiert werden.

From:
https://wiki.trash.net/ - **Trash.Net Wiki**

Permanent link:
https://wiki.trash.net/plattform-migration:konzept

Last update: **2015/09/20 13:56**