vSphere

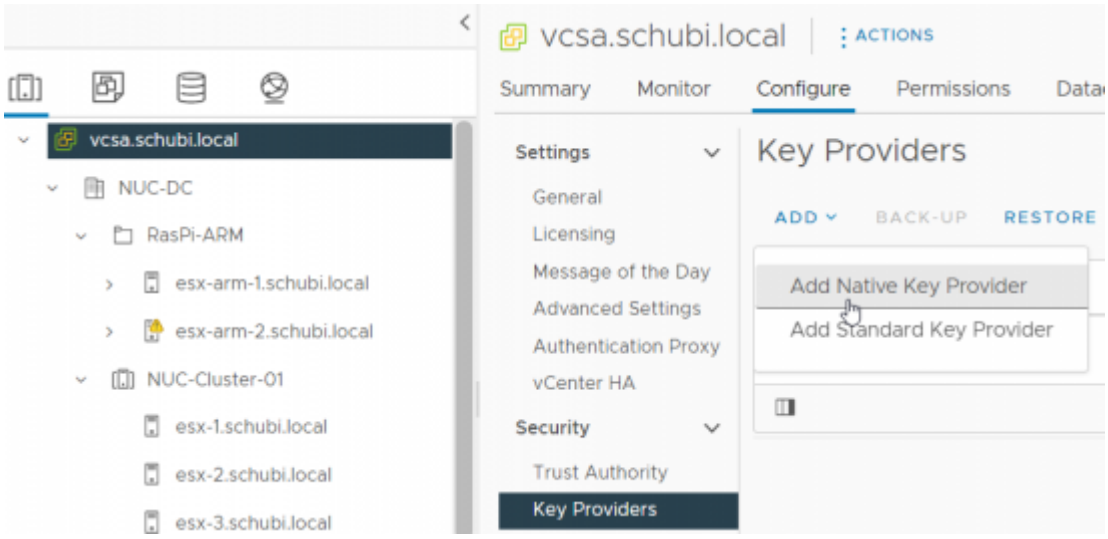
Windows mit Secure Boot und VBS

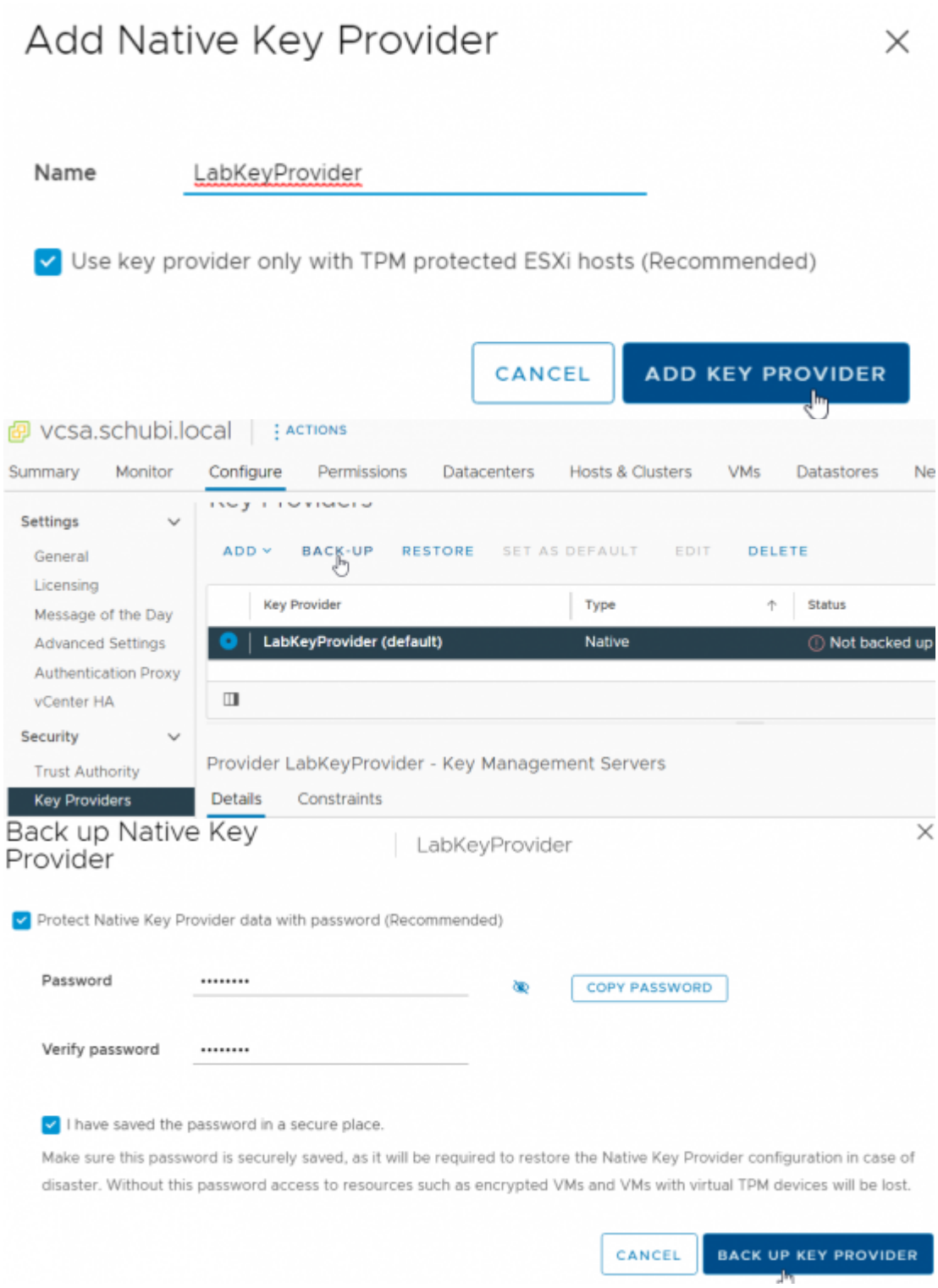
Da das Thema immer mehr nachgefragt wird, hier eine kurze Übersicht, wie man seine VMware Umgebung und VM für UEFI, Secure Boot und Virtual Based Securety (VBS) umstellt.

vTPM & Key Provider

Die meisten aktuellen Server haben einen TPM Chip on Board. Dieser ist jedoch niemals dafür design worden, für 50 oder mehr VMs eine Schlüsselverwaltung zu übernehmen. Der Chip ist schlichtweg damit überfordert.

Deshalb wird von VMware ein vTPM - ein virtuelles Trusted Platform Modul Device - für jede VM bereitgestellt. Die entsprechenden Schlüsselinfos werden in der .nvram Datei der VM gespeichert. Damit nicht “Jeder” den Schlüssel von dort auslesen kann, muss die VM verschlüsselt werden. Um eine VM verschlüsseln zu können, braucht man aber einen Key Provider. Dieser kann ein beliebiger kompatibler Key Provider eines Dritthersteller oder der von VMware bereitgestellte “Native” Key Provider sein. Ich beziehe mich auf den VMware Native Key Provider.





vMotion Problem zwischen vCenter

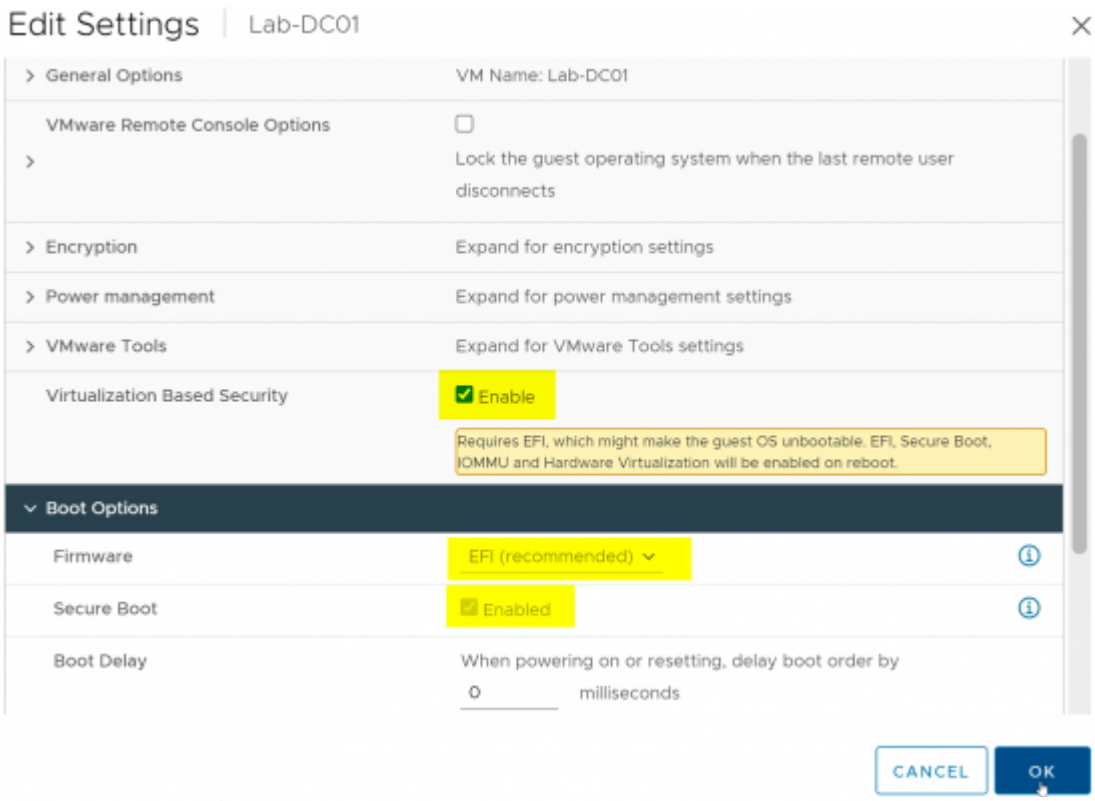
```
config.vpxd.network.allowVmotionBetweenLogicalSwitches = true
```

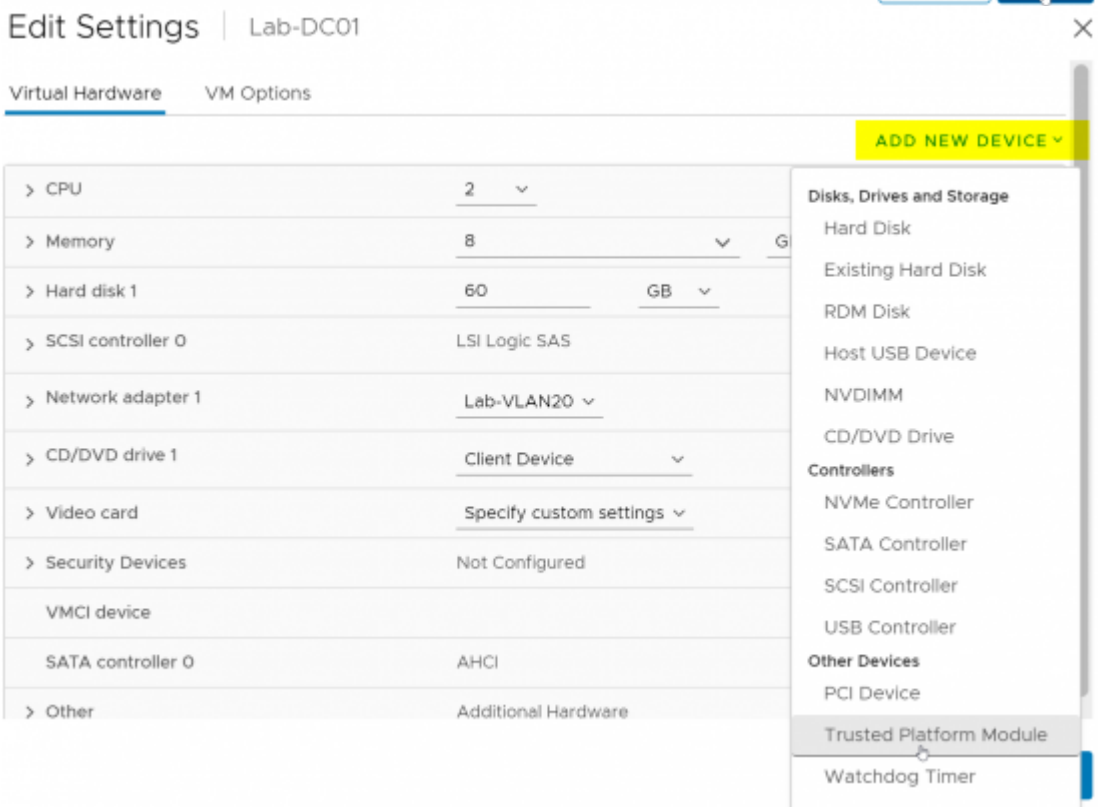
VM auf UEFI umstellen

Wie von BIOS auf UEFI umgestellt wird, habe ich im Windows Teil beschrieben:

windows_evaluation_vhd_zu_vmdk_und_volllizenzierung

VM Parameter für vTPM & VBS





VBS in der VM einschalten

Ich habe mich hier der MS Einführung bedient:
https://docs.microsoft.com/en-us/windows/security/identity-protection/credential-guard/credential-guard-manage

VMFS Datastore per PowerCLI updaten

```
$vcs = Connect-VIServer vcsa.yourdomain.de
$tds = (Get-Datastore svc_temp_vmfs_datastore)
$sds = (Get-Datastore svc_esx_datastore_01)
Update-VmfsDatastore -Datastore $sds -TemporaryDatastore $tds -
TargetVmfsVersion 6 -Server $vcs -Force
```

VMFS on USB

https://www.virten.net/2016/11/usb-devices-as-vmfs-datastore-in-vsphere-esxi-6-5/

Test Kernelports auf Jumbo Frames

```
vmkping -d -s 8972 x.x.x.x

vmkping -I vmkX x.x.x.x

bei VXLAN:
esxcli network diag ping --netstack=vxlan --host <vmknic IP> --df --
size=<packet size>
```

nützliche ESXi Console vDS Befehle

https://kb.vmware.com/s/article/1008127

CPU vs. Core vs. NUMA

Sehr gute Zusammenfassung unter
https://blogs.vmware.com/performance/2017/03/virtual-machine-vcpu-and-vnuma-rightsizing-rules-of-thumb.html

Ich binde direkt die Übersichtstabelle von Mark Achtemichuk ein.

Physical Processor	vCPUs Required	VM Configuration		Resulting vNUMA Nodes Presented
		Resulting Sockets	Core per Socket	
Intel 2 Sockets, 10 Cores per Socket 40 Logical Processors	1	1	1	1
	2	1	2	1
	3	1	3	1
	4	1	4	1
	5	1	5	1
	6	1	6	1
	7	1	7	1
	8	1	8	1
	9	1	9	1
	10	1	10	1
	11	Sub-optimal		
	12	2	6	2
	13	Sub-optimal		
	14	2	7	2
	15	Sub-optimal		
	16	2	8	2
	17	Sub-optimal		
	18	2	9	2
	19	Sub-optimal		
	20	2	10	2

Windows 2012 auf vCenter Installation vorbereiten

Bei Windows 2012 müssen vor der Installation einige Sachen vorbereitet werden. Das es bei mehreren Kunden und auch in unserer eigenen Umgebung sporatisch mit IPv6 Probleme gab, wird auch der IPv6 Stack abgeschaltet.

Hier alle PowerShell Kommandos als Block:

set_w2k12.ps1

```
# IPv6 Stack abschalten
New-ItemProperty -Path
HKLM:\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters -Name
DisabledComponents -PropertyType DWord -Value 0xffffffff

# Flash nachinstallieren
import-module ServerManager
Add-WindowsFeature -name Desktop-Experience,qWave  -
IncludeAllSubFeature -IncludeManagementTools

# .Net 3.5 nachinstallieren (Installquelle unter D:)
Install-WindowsFeature NET-Framework-Core -Source D:\sources\sxs

# UAC deaktivieren (wird beim Upgrade vom SRM gebraucht)
Set-ItemProperty -Path
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system -Name
EnableLUA  -Value 0x00000000

# jeder boot tut gut :-)
Restart-Computer
```

IPv4 über IPv6 bevorzugen:

set_IPv4.ps1

```
# IPv4 vor IPv6
New-ItemProperty -Path
HKLM:\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters -Name
DisabledComponents -PropertyType DWord -Value 0x20

Set-ItemProperty -Path
HKLM:\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters -Name
DisabledComponents -Value 0x20
```

vSAN Cluster auflösen

Man kann nicht einfach ein vSAN Memberhost in ein anderes vCenter oder Cluster übernehmen. Sondern der Host und die Platten müssen aus dem vSAN entfernt werden. Derzeit geht dies nur über esxcli Kommandos.

Das heisst, erst mal die Disks auflisten, dann den automode deaktivieren (sonst sind die Disks gleich wieder drin ), die vSAN Disks entfernen, den Cluster verlassen und prüfen, ob alles chic ist...

Die zu entfernenden Disks kann man über verschiedene Arten referenzieren. Device ID, UUID etc. Ich bevorzuge die UUID. (danach grepen und das Kommando ausführen...)

Hier die Kurzübersicht der Kommandos:

```
esxcli vsan storage list

esxcli vsan storage automode set --enabled false

esxcli vsan storage list | grep "VSAN UUID:"

esxcli vsan storage remove -u 52ec07c4-b4c6-2818-3888-XXXXXXXXXXXX

esxcli vsan cluster leave

esxcli vsan cluster get
```

vSAN Beta in der vSphere 5.5 Umgebung

Für einen Test leider nicht der optimale RAID Controller. Für vSAN sollten alle Platten als JBOD bereitgestellt werden. Der Fujitsu D3116C (LSI2208 V3.0) kann das jedoch nicht.

Also alle SSDs als RAID 0 zusammengefasst, alle HDDs als RAID 0 zusammengefasst. Problem dabei

ist, dass beide Volumes als “non SSD” erkannt werden 😕

```
~ #  esxcli storage core device list
.....
naa.600300570116acb019d80932e01f72c1
   Display Name: Local LSI Disk (naa.600300570116acb019d80932e01f72c1)
   Has Settable Display Name: true
   Size: 5147712
   Device Type: Direct-Access
   Multipath Plugin: NMP
   Devfs Path: /vmfs/devices/disks/naa.600300570116acb019d80932e01f72c1
   Vendor: LSI
   Model: MR SAS 6G 1GB
   Revision: 3.22
...
   Is Local: true
   Is Removable: false
   Is SSD: false
...

naa.600300570116acb019d8089cd7308369
   Display Name: Local LSI Disk (naa.600300570116acb019d8089cd7308369)
   Has Settable Display Name: true
   Size: 379520
...
   Is SSD: false
...
```

Da wird man wohl ein Volume “manuell” zu einer SSD machen müssen. Da hilft zum Glück der VMware KB “Enabling the SSD option on SSD based disks/LUNs that are not detected as SSD by default (2013188)” weiter.

```
~ # esxcli storage nmp satp rule add --satp VMW_SATP_LOCAL  --device
naa.600300570116acb019d8089cd7308369 --option "enable_ssd"
~ # esxcli storage core claiming unclaim --type device --device
naa.600300570116acb019d8089cd7308369
~ # esxcli storage core claimrule load
~ # esxcli storage core claiming reclaim -d
naa.600300570116acb019d8089cd7308369
~ # esxcli storage core claimrule run
~ # esxcli storage core device list -d naa.600300570116acb019d8089cd7308369
naa.600300570116acb019d8089cd7308369
   Display Name: Local LSI Disk (naa.600300570116acb019d8089cd7308369)
   Has Settable Display Name: true
   Size: 379520
...
   Is SSD: true
...
```

oder mit einem kleinem Script:

```
#!/bin/sh

echo $1
esxcli storage nmp satp rule add --satp VMW_SATP_LOCAL  --device $1 --option
"enable_ssd"
esxcli storage core claiming unclaim --type device --device $1
esxcli storage core claimrule load
esxcli storage core claimrule run
esxcli storage core device list -d $1
```

Um vSAN in Betrieb zu nehemen, muss man eine Lizenz im vCenter einspielen und diese Lizenz auch dem Cluster zuweisen. Sonst kann man Disks hinzufügen, die Aktion wird mit success beendet aber keine Platten benutzt. Das hat bei mir eine Weile gedauert, das Problem zu finden (Dazu auch ein kurzer Artikel von Duncan Epping http://www.yellow-bricks.com/2013/09/25/initialized-disks-used-vsan-completed-successfully-disks-added/)

ausgegrauter Datastore lässt sich nicht löschen

Wenn der Storageadmin zu schnell war und einfach eine Datastore LUN entfernt hat, ohne diesen Datastore voher von den ESXi Servern zu unmounten, belibt der Datastore grau und man kann ihn nicht mehr Löschen.

Abhilfe gibt's dann nur noch per SQL Statements. Vorher bitte den vCenter Service stoppen.

Folgendes Script hilft (natürlich ohne jegliche Fehlerbehandlung):

```
DECLARE @DATASTORE VARCHAR(50)
SET @DATASTORE='Datastorename'

DECLARE @DATASTORE_ID INT

SELECT @DATASTORE_ID=ID FROM VPX_ENTITY WHERE NAME = @DATASTORE;

DELETE FROM VPX_DS_ASSIGNMENT WHERE DS_ID=@DATASTORE_ID;
DELETE FROM VPX_VM_DS_SPACE WHERE DS_ID=@DATASTORE_ID;
DELETE FROM VPX_DATASTORE WHERE ID=@DATASTORE_ID;

DELETE FROM VPX_ENTITY WHERE ID=@DATASTORE_ID;

-- es sollten keine Ausgaben kommen
SELECT * FROM VPX_DS_ASSIGNMENT WHERE DS_ID=@DATASTORE_ID;
SELECT * FROM VPX_VM_DS_SPACE WHERE DS_ID=@DATASTORE_ID;
SELECT * FROM VPX_DATASTORE WHERE ID=@DATASTORE_ID;
SELECT * FROM VPX_ENTITY WHERE ID=@DATASTORE_ID;
```

Danach den vCenter Service starten

vCenter Datenbank zu Groß?

Mehrfach hatten wir schon bei Kunden und bei uns vCenter Datenbanken, die im Verhältnis zur Anzahl der Hosts und VMs zu groß waren. Nicht immer sind das überqellende Performancecounter. Oft gibt es ein kleines Problem in der Umgebung, dass dafür sorgt, dass die Eventtabellen und davon abhängigen Tabellen mit vielen gleichen Einträgen zugemüllt wird.

Ich versuche hier mal ein paar SQL Queries für die Suche zu sammeln.

Es gibt für das wachsende Event Log Problem auch einen VMware KB (Purging old data from the database used by VMware vCenter Server 4.x and 5.x).

Wo liegt das Problem?

Als erstes muss man mal schauen, welche Tabellen groß werden:

getTableInfos.sql

```
USE vcenter01
GO

SELECT
    t.NAME AS TableName,
    s.Name AS SchemaName,
    p.rows AS RowCounts,
    SUM(a.total_pages) * 8 AS TotalSpaceKB,
    SUM(a.used_pages) * 8 AS UsedSpaceKB,
    (SUM(a.total_pages) - SUM(a.used_pages)) * 8 AS UnusedSpaceKB
FROM
    sys.tables t
INNER JOIN
    sys.indexes i ON t.OBJECT_ID = i.object_id
INNER JOIN
    sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id =
p.index_id
INNER JOIN
    sys.allocation_units a ON p.partition_id = a.container_id
LEFT OUTER JOIN
    sys.schemas s ON t.schema_id = s.schema_id
WHERE
    t.NAME NOT LIKE 'dt%'
    AND t.is_ms_shipped = 0
    AND i.OBJECT_ID > 255
GROUP BY
    t.Name, s.Name, p.Rows
ORDER BY
      UsedSpaceKB DESC
    --t.Name
```

Das ergibt dann ggf. so ein Ergebnis:

```
TableName           SchemaName   RowCounts        TotalSpaceKB
UsedSpaceKB   UnusedSpaceKB
VPX_EVENT_ARG         dbo        108753600          39336304        39330792
5512
VPX_EVENT             dbo         15208355           8305024         8303096
1928
VPX_HIST_STAT2_9      dbo          1331228             79856           79672
184
VPX_HIST_STAT2_17     dbo          1328635             79648           79488
160
VPX_HIST_STAT2_16     dbo          1328274             79656           79480
176
VPX_HIST_STAT2_11     dbo          1327713             79656           79448
208
VPX_HIST_STAT2_12     dbo          1327717             79656           79448
208
VPX_HIST_STAT2_13     dbo          1327474             79528           79424
104
```

Man sieht, dass es viele Events gibt. Jetzt herausbekommen, welche problematisch sind. Da ich nicht weiss, in welchem Zeitlichen Rahmen dies geht, lasse ich mir die derzeitige höchste ID der Eventtabelle geben.

```
SELECT MAX(event_id) FROM VPX_EVENT
```

Dann schaue ich, welche die letzten Events sind, indem ich einfach von der max. ID ein paar Tausend abziehe und eine neue Abfrage mit diesem Wert starte.

```
USE vcenter01
GO
SELECT
*
FROM VPX_EVENT
WHERE EVENT_ID > 15330000
GO
```

Dann sieht man ggf. sehr häufig auftretende Events z.B.

```
EVENT_ID            CHAIN_ID                                    EVENT_TYPE
EXTENDED_CLASS ...
15332944            15332944          esx.problem.net.vmknic.ip.duplicate
2       ...
```

Oder diese etwas bessere Version 

```
USE [vcenter_ifb-vc-01]
GO
```

```
SELECT EVENT_TYPE, COUNT(EVENT_TYPE) AS EventRows
FROM dbo.VPX_EVENT
GROUP BY EVENT_TYPE
ORDER BY EventRows DESC
GO
```

Das ergibt z.B.

```
EVENT_TYPE                                      EventRows
vim.event.UserLoginSessionEvent                    217741
vim.event.UserLogoutSessionEvent                   217616
vim.event.AlarmStatusChangedEvent                   24173
vim.event.TaskEvent                                  5769
com.vmware.vc.VmDiskConsolidatedEvent                1575
vim.event.CustomFieldValueChangedEvent               1524
vim.event.VmResourceReallocatedEvent                 1084
com.symantec.netbackup.backup.success.v2              952
vim.event.ScheduledTaskStartedEvent                   754
vim.event.ScheduledTaskCompletedEvent                 742
vim.event.VmReconfiguredEvent                         657
com.symantec.netbackup.backup.success                 567
...
```

Bereinigung

Danach gibt es verschiedene Möglichkeiten diese Events zu löschen. Alle betroffene Events mit einmal zu löschen ist nicht zu empfehlen. Das Transaktion Log kann sehr, sehr groß werden. Man kann dies jedoch “Stückeln”.

```
DELETE FROM VPX_EVENT WHERE EVENT_ID IN
             (SELECT TOP 100000 EVENT_ID FROM VPX_EVENT
              WHERE
              EVENT_TYPE='vim.event.UserLoginSessionEvent'
              )
GO
```

Das muss man wiederholen, bis alles gelöscht ist. Oder man automatisiert dies. Eine Anleitung ist in der SQL Section zu finden Transaction Rows verkleinern.

Bereinigung der EventLogs für Faule

Für Faule gibts das folgende Script. Es setzt temporär das Event und Task Aufbewahrungsalter auf max. 30 Tage und alles darüber hinaus wird TransLog-freundlich gelöscht, was will man mehr 🙂

fuerFaule.sql

```
USE [ifb-vc-02]
GO

DECLARE @maxAgeEvent nvarchar(255), @maxAgeTask nvarchar(255),
@maxAgeEventEnabled nvarchar(255), @maxAgeTaskEnabled nvarchar(255);

SELECT @maxAgeEvent=VALUE FROM dbo.VPX_PARAMETER WHERE
NAME='event.maxAge'
SELECT @maxAgeTask=VALUE FROM dbo.VPX_PARAMETER WHERE
NAME='task.maxAge'
SELECT @maxAgeEventEnabled=VALUE FROM dbo.VPX_PARAMETER WHERE
NAME='event.maxAgeEnabled'
SELECT @maxAgeTaskEnabled=VALUE FROM dbo.VPX_PARAMETER WHERE
NAME='task.maxAgeEnabled'

SELECT * FROM dbo.VPX_PARAMETER WHERE NAME='event.maxAge' OR
NAME='task.maxAge' OR NAME='event.maxAgeEnabled' OR
NAME='task.maxAgeEnabled'

UPDATE dbo.VPX_PARAMETER SET VALUE='30' WHERE NAME='event.maxAge'
UPDATE dbo.VPX_PARAMETER SET VALUE='30' WHERE NAME='task.maxAge'
UPDATE dbo.VPX_PARAMETER SET VALUE='true' WHERE
NAME='event.maxAgeEnabled'
UPDATE dbo.VPX_PARAMETER SET VALUE='true' WHERE
NAME='task.maxAgeEnabled'


SELECT * FROM dbo.VPX_PARAMETER WHERE NAME='event.maxAge' OR
NAME='task.maxAge' OR NAME='event.maxAgeEnabled' OR
NAME='task.maxAgeEnabled'

-- This purges the data from the vpx_event, vpx_event_arg, and vpx_task
tables based on the date specified for maxAge.
-- Diese SP läuft max. 1 Stunde. Wenn sehr viele Events vorhanden sind,
gg. nochmals das ganze Script starten
EXEC dbo.cleanup_events_tasks_proc

UPDATE dbo.VPX_PARAMETER SET VALUE=@maxAgeEvent WHERE
NAME='event.maxAge'
UPDATE dbo.VPX_PARAMETER SET VALUE=@maxAgeTask WHERE NAME='task.maxAge'
UPDATE dbo.VPX_PARAMETER SET VALUE=@maxAgeEventEnabled WHERE
NAME='event.maxAgeEnabled'
UPDATE dbo.VPX_PARAMETER SET VALUE=@maxAgeTaskEnabled WHERE
NAME='task.maxAgeEnabled'

SELECT * FROM dbo.VPX_PARAMETER WHERE NAME='event.maxAge' OR
NAME='task.maxAge' OR NAME='event.maxAgeEnabled' OR
NAME='task.maxAgeEnabled'
GO
```

Egal wie man löscht, sollte man anschließend das Translog verkleinern. Siehe dazu Transactionlog verkleinern.

Single Sign On (SSO) Service von vSphere 5.1

Mit vSphere 5.1 hat sich einiges bei der Authentifikation von Nutzern geändert. Das neue Herzstück ist ein Single Sign On (SSO) Service. Ein solcher Dienst ermöglicht es, die einmalige Anmeldung des Nutzers für viele Anwendungen und Plugins bereit zustellen. Es können mehrere SSO Dienste installiert werden und ggf. hinter Loadbalancern aufgestellt werden.

So muss man beim vCenter Orchestrator 5.1 nicht mehr umständlich die LDAP Einstellungen vornehmen, sondern nutzt den SSO Service mit wenigen Klicks.

Die Autorisierung, spricht, welche Rechte ein Nutzer im vCenter hat, wird weiterhin wie gewohnt festgelegt und verwaltet.

Ein weitere Vorteil ist die granulare Festlegung, welche Directory Services sollen genutzt werden und sind z.B. lokale Nutzer überhaupt berechtigt. Somit kann man einfach Nutzer gegen verschieden ADs etc. prüfen. Das hat zur Folge, das man an vSphere 5.1 bei der Anmeldung die entsprechende Domäne mit angeben muss.

Bedeutung des Master Passwords des SSO Services

Da der SSO Dienst von zentraler Bedeutung ist, wird bei der Installation ein Default SSO Administrator festgelegt (admin) mit einem s.g. “Master Password”. Dieses Masterpassword ist *sehr, sehr* wichtig 🙂

Nach der Installation ist nur dieser Nutzer berechtigt, Konfigurationen am SSO Service vorzunehmen. Was ist beim SSO Dienst zu konfigurieren?

- Welche Verzeichnisdienste werden zur Authentifikation genutzt
- Welche Nutzerkennung wird zur Kommunikation mit den Verzeichnisdiensten genutzt
- Welche Nutzer oder Gruppen sind neben dem Build IN Admin noch berechtigt, den SSO Serveice zu konfigurieren

Ein SSO Admin wird auch benötigt, wenn andere Dienste für die Nutzung von SSO initial konfiguriert werden.

Noch mal zur Erinnerung – nach der Installation ist auch ein Domänenadmin *kein* SSO Admin!

SSO Masterpassword weg - eine __unsupportete__ Lösung

Das Masterpassword ist ist ja essentiell. Kurz nach dem Erscheinen con vSphere 5.1 habe ich eine unsere vCenter Umgebungen auf 5.1 geupgrated. Da Dokumentation lesen nur was für Weichlinge ist 🙂 habe ich zu dem Zeitpunkt die Bedeutung des Master Password nicht richtig eingeordnet. Ich war der Meinung, dass ich bei konfigurierten Domänenauthentifikation als Domänenadmin schon alles

richten könnte…

Weit gefehlt! Nach 10 Tagen wollte ich meinen auf 5.1 angehobenen vCenter Orchestrator auf SSO umstellen und es ging mit dem Domänenadmin nicht ☹ Also wollte ich die Einstellungen prüfen. Kurzer Blick in die Doku → Masterpassword! Da hatte ich so eine Ahnung. Übliches Password mit Zahl und Sonderzeichen erweitert. Aber wie war es genau? Das Durchprobieren des Passwods wird dadurch erschwert, dass nach 3-maliger falschen Eingabe erst einmal 15 Minuten gewartet werden muss.

Wie erwartet hatte ich keinen Erfolg. Da kam mir die Idee den SSO Dienst einfach neu zu installieren und dabei das Password neu zu setzten. Schön gedacht, aber man braucht zum Deinstallieren das Masterpassword ☹

Somit habe ich einen SR bei VMware eröffnet, ob es einen Weg aus dieser Dilemma gibt. Die recht schnelle Antwort, die über das Enineering abgesichert wurde, lautete *“There is no supported way to change the master password, if you forgot it.”*

Im Klartext bedeutet das, die komplette vCenter Umgebung neu gegen die bestehende vCenter Datenbank auf einem neuem Betriebssystem zu installieren – mit allem was daran hängt.

Hier eine Lösung, die aber vom VMware Support ausdrücklich als ***unsupportet*** bezeichnet wird, wie man wieder Zugriff auf den SSO Service bekommen kann.

Die Schritte in der Übersicht:

1. Einrichten einer neuer SSO Datenbank
2. Installieren einen neuen SSO Services gegen dieses Datenbank
3. Masterpassword merken!
4. Auslesen des sha256 gehashten Passwords dieses Nutzers aus dieser neuen Datenbank
5. Herunterfahren des produktiven SSO Dienstes
6. Update des Password Hash in der produktiven Datenbank
7. Neustart der kompletten vCenter Umgebung
8. Anmeldung an die produktive Umgebung mit Nutzer admin und dem neuen Password
9. Hinzufügen weitere SSO Admins ☺

Den Punkt 4) und 6) möchte ich hier etwas näher für den SQLServer zeigen. Ich empfehle die SQL Query im Management Studio des SQL Servers auszuführen. Voraussetzung ist natürlich die SQL Berechtigung für den Zugriff auf die produktive SSO DB und die Dummy SSO DB.

Der Password Hash kann aus der neuen Dummy DB mit Hilfe des folgenden Query abgefragt werden:

get_sso_masterhash.sql

```
SELECT
    [PASSWORD]
FROM
    [dbo].[IMS_PRINCIPAL]
WHERE
    LOGINUID = 'admin'
```

```
AND
    PRINCIPAL_IS_DESCRIPTION = 'Admin'
```

Als Ergebnis sollte dieser Query nur eine Zeile mit dem Hash zurück liefern. Dieser Password Hash sieht so aus {SSHA256}RwKqhx0O4owMZv……I3svAd5bRNlNHYZoetfk7uVg== .

Diesen Hash muss man sich merken und gegen die produktive DB folgenden Query ausführen:

set_sso_masterhash.sql

```
UPDATE
    [dbo].[IMS_PRINCIPAL]
SET
    [PASSWORD] =
'{SSHA256}RwKqhx0O4owMZv.....I3svAd5bRNlNHYZoetfk7uVg=='
WHERE
    LOGINUID = 'admin'
AND
    PRINCIPAL_IS_DESCRIPTION = 'Admin'
```

Bei meinem Versuch hat es nicht gereicht, nur den SSO Service zu starten. (Verbindungsfehler beim Anmelden mit dem Web Client) Ich habe kurzerhand die komplette vCenter VM restartet und dann ging alles.

gemischter Zertifikatsbetrieb

Anscheinend gibt es bei 5.1 Probleme, wenn manche Teile der vSphere Umgebung mit selfsign Certs arbeiten und manche mit CA Certs.

Aus meiner Sicht spricht nichts dagegen, jedoch habe ich immer wieder Probleme mit dem WebClient oder vCenter in solchen Konstellationen. Man sollte deshalb konsequent entweder selfsign Certs oder CA Certs verwenden.

Mit dem neuen VMware Tool lassen sich Zertifikate auch einfacher austauschen.

vCenter Hostnamen nachträglich ändern

http://kb.vmware.com/kb/5850444

http://communities.vmware.com/message/732613

http://timjacobs.blogspot.com/2008/05/renaming-virtualcenter-25-server.html

ESXi mit weniger als 2GB RAM betreiben

1. ESXi mit 2 GB oder mehr RAM installieren
2. `/etc/vmware/esx.conf` editieren und folgende Zeile einfügen

```
/vmkernel/minMemoryCheck = "false"
```

3. reboot

iSCSI Multipathing bzw. welche Kernelports werden für iSCSI genutzt

Es gibt immer mal wieder bei VMware ESX(i) mit iSCSI das Problem, dass man festlegen muss, welche Kernelports für den iSCSI Traffic genutzt werden.

Meist ist der Grund dafür, dass die Console im selben Netzsegment liegt, wie der iSCSI Storage oder es werden separate vSwitche mit unterschiedlicher Config genutzt. Leider kann man mit der GUI nicht festlegen, welche Porsts verwendet werden sollen (falls doch, bitte kurze eMail mit einem Hinweis an mich). Somit muss man sich auf das CLI verlassen und brav ein paar Kommandos eintibbern. Nebenbei kann man gleich noch Multipathing für die iSCSI Lun konfigurieren.

Es gibt meiner Meinung nach drei wichtige Möglichkeiten iSCSI ausfallsicher anzubinden

Anbindung	**Beschreibung**	**Vorteile**	**Nachteil**
ein Kernelport + pNics mit virtual Port ID	Es wird ein Kernelport für iSCSI erstellt. Wegen Redundanz werden mehrere pNics zugeordnet und im Failovermode “Virtual Port ID” betrieben.	Einfach und schnell eingerichtet, arbeitet sehr zuverlässig, keine Anpassungen auf pSwitch Seite.	Da bei einem einzigen iSCSI Target nur eine pNic verwendet wird, ist man auf pNic Durchsatz begrenzt.
ein Kernelport + pNIcs im Channel (Trunk)	Es wird ein Kernelport für iSCSI erstellt. Wegen Redundanz werden mehrere pNics zugeordnet. Die pNics werden zu einem Channel zusammengefasst (VMware: Trunk)	je nach pSwitch Ausfälle auf einzelnen pNics gut abgefangen, max. Durchsatz = Anzahl pNics * pNic Durchsatz	Mehr Aufwand bei der Config - vorallem auf pSwitch Seite, Verbindungen über mehrere Switche nicht mehr ohne weiteres möglich, bei manchen pSwitche nicht sehr robust
mehrere Kernelports + Single pNic + Multipathing	Es werden mehrere Kernelports für iSCSI erstellt. Jeder Port erhält nur eine pNic. Wegen Redundanz wird Multipathing über diese Kernelports benutzt.	max. Durchsatz = Anzahl pNics * pNic Durchsatz, robust durch LUN Multipathing, pNics können mit unterschiedlichen pSwitche verbunden werden, keine Anpassungen auf pSwitch Seite.	Konfiguration z.Z. nur über CLI möglich, für jede LUN muss Pathpolicy auf RoundRobin umgestellt werden.

Auf die letzte Art der Anbindung möchte ich nun eingehen, da sie zudem noch die Wahl der KernelPorts zulässt.

Für die Konfiguration muss man auf das CLI, ist aber nicht schlimm. Folgende Punkte müssen abgearbeitet werden

1. es müssen entsprechende Kernelports konfiguriert werden
2. jedem Port wird eine pNic zugeordnet, sowie eine IP und GW
3. SW iSCSI wird eingeschaltet
4. die Kernelports werden dem SW iSCSI HBA zugeordnet
5. das iSCSI Target wird discovered
6. die Luns werden gescanned

Sieht also sehr übersichtlich aus.

Ich habe mal eine CLI Session mitgelogged. Dabei wurde ESX an eine EqualLogic[1)] angebunden. Für den iSCSI Traffic wurden 3 pNics verwendet und JumboFrames eingeschaltet.

Alle Befehle für das iSCSI Multipathing ohne Ausgabe und Kommentar

prepare_iscsi.sh

```
esxcfg-vswitch -a swiscsi
esxcfg-vswitch -m 9000 swiscsi
esxcfg-vswitch -A iSCSI1 swiscsi
esxcfg-vswitch -A iSCSI2 swiscsi
esxcfg-vswitch -A iSCSI3 swiscsi
esxcfg-vmknic -a -i 10.3.4.111 -n 255.255.0.0 -m 9000 iSCSI1
esxcfg-vmknic -a -i 10.3.4.112 -n 255.255.0.0 -m 9000 iSCSI2
esxcfg-vmknic -a -i 10.3.4.113 -n 255.255.0.0 -m 9000 iSCSI3
esxcfg-vswitch -L vmnic1 swiscsi
esxcfg-vswitch -L vmnic2 swiscsi
esxcfg-vswitch -L vmnic3 swiscsi
esxcfg-vswitch -p iSCSI1 -N vmnic2 swiscsi
esxcfg-vswitch -p iSCSI1 -N vmnic6 swiscsi
esxcfg-vswitch -p iSCSI2 -N vmnic1 swiscsi
esxcfg-vswitch -p iSCSI2 -N vmnic6 swiscsi
esxcfg-vswitch -p iSCSI3 -N vmnic1 swiscsi
esxcfg-vswitch -p iSCSI3 -N vmnic2 swiscsi
esxcfg-swiscsi -e
esxcfg-scsidevs -a
esxcfg-vmknic -l
vmkiscsi-tool -V -a vmk1 vmhba38
vmkiscsi-tool -V -a vmk2 vmhba38
vmkiscsi-tool -V -a vmk3 vmhba38
vmkiscsi-tool -D -a 10.3.4.22 vmhba38
esxcfg-rescan vmhba38
esxcli nmp device list | grep naa | grep -v Device
esxcli nmp device setpolicy --device
naa.6090a08850de3fef31d22401000040be --psp VMW_PSP_RR
exit
```

Alle Befehle für das iSCSI Multipathing mit Ausgabe und Kommentar

Damit die gesamte Befehlskette ausgeführt werden kann, habe ich den Ausgaben der Befehle Kommentarzeichen vorangestellt.

prepare_iscsi_02.sh

```
# wir bauen einen iSCSI Switch
esxcfg-vswitch -a swiscsi
# iSCSI soll mit Jumboframes ab  laufen, somit wird der Switch
Jumbotauglich gemacht
esxcfg-vswitch -m 9000 swiscsi
# Jetzt jetzt brauch ich an diesem switch 3 Kernelports  , die
ebenfalls Jumboframes unterstützen
# dazu wird erst eine neue Portgruppe erstellt und dann dieser
Portgruppe eine IP  /GW plus Jumbo MTU zugeteilt
esxcfg-vswitch -A iSCSI1 swiscsi
esxcfg-vswitch -A iSCSI2 swiscsi
esxcfg-vswitch -A iSCSI3 swiscsi
esxcfg-vmknic -a -i 10.3.4.17 -n 255.255.0.0 -m 9000 iSCSI1
esxcfg-vmknic -a -i 10.3.4.18 -n 255.255.0.0 -m 9000 iSCSI2
esxcfg-vmknic -a -i 10.3.4.19 -n 255.255.0.0 -m 9000 iSCSI3
# da wir Multipathing nutzen, erhält jeder Portgruppe nur eine
physische Netzwerkkarte
# über Commandozeile geht das am einfachsten, wenn erst dem Switch
komplett alle Netzwerkkarten
# zugewiesen werden und danach bei jeder Portgruppe die nicht
benötigten Nics entfernt werden
esxcfg-vswitch -L vmnic1 swiscsi
esxcfg-vswitch -L vmnic2 swiscsi
esxcfg-vswitch -L vmnic6 swiscsi
esxcfg-vswitch -p iSCSI1 -N vmnic2 swiscsi
esxcfg-vswitch -p iSCSI1 -N vmnic6 swiscsi
esxcfg-vswitch -p iSCSI2 -N vmnic1 swiscsi
esxcfg-vswitch -p iSCSI2 -N vmnic6 swiscsi
esxcfg-vswitch -p iSCSI3 -N vmnic1 swiscsi
esxcfg-vswitch -p iSCSI3 -N vmnic2 swiscsi
# so sollte dann das ergebnis aussehen
esxcfg-vswitch -l
#Switch Name      Num Ports   Used Ports  Configured Ports  MTU
Uplinks
#vSwitch0         128         3           128               1500
vmnic0
#
#  PortGroup Name        VLAN ID  Used Ports  Uplinks
#  VM Network            0        0           vmnic0
#  Management Network    1002     1           vmnic0
#
#Switch Name      Num Ports   Used Ports  Configured Ports  MTU
Uplinks
```

```
#swiscsi          128         7           128               9000
vmnic1,vmnic2,vmnic6
#
#  PortGroup Name        VLAN ID  Used Ports  Uplinks
#  iSCSI3                0        1          vmnic6
#  iSCSI2                0        1          vmnic2
#  iSCSI1                0        1          vmnic1
#
# jetzt kann man den SW iSCSI Initiator einrichten
# zu  erst einschalten
esxcfg-swiscsi -e
# Enabling software iSCSI...
# nun kommt das eigentlich spannende, es müssen die richtigen
Kernelports dem
# iSCSI Adapter beigebracht werden.
# erst einmal nachschauen, was wir brauchen:
esxcfg-scsidevs -a
#vmhba38 usb-storage       link-n/a  usb.vmhba38
() USB
#vmhba39 iscsi_vmk         online    iscsi.vmhba39
iSCSI Software Adapter
#vmhba0  ata_piix          link-n/a  sata.vmhba0
(0:0:31.2) Intel Corporation PowerEdge R610 SATA IDE Controller
#vmhba32 usb-storage       link-n/a  usb.vmhba32
() USB
#vmhba33 bnx2i             unbound   iscsi.vmhba33
Broadcom iSCSI Adapter
#vmhba34 bnx2i             unbound   iscsi.vmhba34
Broadcom iSCSI Adapter
#vmhba35 bnx2i             unbound   iscsi.vmhba35
Broadcom iSCSI Adapter
#vmhba36 bnx2i             unbound   iscsi.vmhba36
Broadcom iSCSI Adapter
#vmhba37 ata_piix          link-n/a  sata.vmhba37
(0:0:31.2) Intel Corporation PowerEdge R610 SATA IDE Controller
# wir benötigen den vmhba39
# und wir müssen wissen, welche Kernelports...
esxcfg-vmknic -l
# Interface  Port Group/DVPort   IP Family IP Address
Netmask         Broadcast       MAC Address       MTU     TSO MSS
Enabled Type
# vmk0       Management Network  IPv4      10.3.4.103
255.255.0.0     10.3.255.255    f0:4d:a2:07:67:d6 1500    65535
true    STATIC
# vmk1       iSCSI1              IPv4      10.3.4.17
255.255.0.0     10.3.255.255    00:50:56:74:44:44 9000    65535
true    STATIC
# vmk2       iSCSI2              IPv4      10.3.4.18
255.255.0.0     10.3.255.255    00:50:56:75:02:ab 9000    65535
true    STATIC
# vmk3       iSCSI3              IPv4      10.3.4.19
```

```
255.255.0.0     10.3.255.255    00:50:56:7a:05:c4 9000    65535
true    STATIC
# in unserem Fall vmk1 bis vmk3
# dann werden die jetzt an den iSCSI Adapter gebunden
vmkiscsi-tool -V -a vmk1 vmhba39
# Adding NIC vmk1 ...
# Added successfully.
vmkiscsi-tool -V -a vmk2 vmhba39
# Adding NIC vmk2 ...
# Added successfully.
vmkiscsi-tool -V -a vmk3 vmhba39
#Adding NIC vmk3 ...
#Added successfully.
# ab der stelle wird es einfacher mit der GUI - die Eqiuallogic Group
IP eintragen und
# LUNs discovern.
#Danach bei den Multipathingeigenschaften von Fixed auf RoundRobin
stellen
# ich versuchs trotzdem über CLI...
vmkiscsi-tool -D -a 10.3.4.22 vmhba39
esxcfg-rescan vmhba39
esxcfg-mpath -L
# vmhba32:C0:T0:L0 state:active mpx.vmhba32:C0:T0:L0 vmhba32 0 0 0 NMP
active local usb.vmhba32 usb.0:0
# vmhba38:C0:T0:L0 state:active mpx.vmhba38:C0:T0:L0 vmhba38 0 0 0 NMP
active local usb.vmhba38 usb.0:0
# vmhba39:C2:T0:L0 state:active naa.6090a08850de3fef31d22401000040be
vmhba39 2 0 0 NMP active san iqn.1998-01.com.vmware:vm3-vmware-
console-5883c9c3 00023d000003,iqn.2001-05.com.equallogic:0-8a0906-
ef3fde508-be4000000124d231-vmfs1,t,1
# vmhba39:C1:T0:L0 state:active naa.6090a08850de3fef31d22401000040be
vmhba39 1 0 0 NMP active san iqn.1998-01.com.vmware:vm3-vmware-
console-5883c9c3 00023d000002,iqn.2001-05.com.equallogic:0-8a0906-
ef3fde508-be4000000124d231-vmfs1,t,1
# vmhba39:C0:T0:L0 state:active naa.6090a08850de3fef31d22401000040be
vmhba39 0 0 0 NMP active san iqn.1998-01.com.vmware:vm3-vmware-
console-5883c9c3 00023d000001,iqn.2001-05.com.equallogic:0-8a0906-
ef3fde508-be4000000124d231-vmfs1,t,1
# schön die LUN hat 3 Pfade (c0:t0l0:l0l0; c1:7 t0:l0; c2:t0:lo 0)
# welches   D e v i c e   e n t s p r i c h t   d i e s e   L U N
esxcli nmp device list | grep naa | grep -v Device
# welche Multipathingpolicy nutzt diese LUN
esxcli nmp device list
# mpx.vmhba32:C0:T0:L0
#     Device Display Name: Local USB Direct-Access
(mpx.vmhba32:C0:T0:L0)
#     Storage Array Type: VMW_SATP_LOCAL
#     Storage Array Type Device Config: SATP VMW_SATP_LOCAL does  not
support device configuration.
#     Path Selection Policy: VMW_PSP_FIXED
#     Path Selection Policy Device Config:
```

```
{preferred=vmhba32:C0:T0:L0;current=vmhba32:C0:T0:L0}
#     Working Paths: vmhba32:C0:T0:L0
#
# mpx.vmhba38:C0:T0:L0
#     Device Display Name: Local USB CD-ROM (mpx.vmhba38:C0:T0:L0)
#     Storage Array Type: VMW_SATP_LOCAL
#     Storage Array Type Device Config: SATP VMW_SATP_LOCAL does not
support device configuration.
#     Path Selection Policy: VMW_PSP_FIXED
#     Path Selection Policy Device Config:
{preferred=vmhba38:C0:T0:L0;current=vmhba38:C0:T0:L0}
#     Working Paths: vmhba38:C0:T0:L0
#
# naa.6090a08850de3fef31d22401000040be
#     Device Display Name: EQLOGIC iSCSI Disk
(naa.6090a08850de3fef31d22401000040be)
#     Storage Array Type: VMW_SATP_EQL
#     Storage Array Type Device Config: SATP VMW_SATP_EQL does not
support device configuration.
#     Path Selection Policy: VMW_PSP_FIXED
#     Path Selection Policy Device Config:
{preferred=vmhba39:C2:T0:L0;current=vmhba39:C2:T0:L0}
#     Working Paths: vmhba39:C2:T0:L0
# VMW_PSP_FIXED ist nicht was wir wollen, sondern RoundRobin, also
setzen wir das...
esxcli nmp device setpolicy --device
naa.6090a08850de3fef31d22401000040be --psp VMW_PSP_RR
esxcli nmp device list
# mpx.vmhba32:C0:T0:L0
#     Device Display Name: Local USB Direct-Access
(mpx.vmhba32:C0:T0:L0)
#     Storage Array Type: VMW_SATP_LOCAL
#     Storage Array Type Device Config: SATP VMW_SATP_LOCAL does not
support device configuration.
#     Path Selection Policy: VMW_PSP_FIXED
#     Path Selection Policy Device Config:
{preferred=vmhba32:C0:T0:L0;current=vmhba32:C0:T0:L0}
#     Working Paths: vmhba32:C0:T0:L0
#
# mpx.vmhba38:C0:T0:L0
#     Device Display Name: Local USB CD-ROM (mpx.vmhba38:C0:T0:L0)
#     Storage Array Type: VMW_SATP_LOCAL
#     Storage Array Type Device Config: SATP VMW_SATP_LOCAL does not
support device configuration.
#     Path Selection Policy: VMW_PSP_FIXED
#     Path Selection Policy Device Config:
{preferred=vmhba38:C0:T0:L0;current=vmhba38:C0:T0:L0}
#     Working Paths: vmhba38:C0:T0:L0
#
# naa.6090a08850de3fef31d22401000040be
#     Device Display Name: EQLOGIC iSCSI Disk
```

```
(naa.6090a08850de3fef31d22401000040be)
#     Storage Array Type: VMW_SATP_EQL
#     Storage Array Type Device Config: SATP VMW_SATP_EQL does not
support device configuration.
#     Path Selection Policy: VMW_PSP_RR
#     Path Selection Policy Device Config:
{policy=rr,iops=1000,bytes=10485760,useANO=0;lastPathIndex=1:NumIOsPend
ing=0,numBytesPending=0}
#     Working Paths: vmhba39:C0:T0:L0, vmhba39:C1:T0:L0,
vmhba39:C2:T0:L0
# :-) nun gehts mit RoundRobin los.
# somit sollte mit bis zu 300 MB geschrieben oder gelesen werden können
#
```

ESXi im ESXi mit 64Bit Inner Guest

Interessant ist der Ansatz, ein ESXi in einer ESXi VM zu betreiben. Mit ein paar Optionen können darin dann sogar 64 Bit VMs betrieben werden. Damit lassen sich wunderbar “transportable” Test- und Demoumgebungen bauen.

Was muss ich machen:

Auf dem “echten” ESXi Host:

- ESXi 5.x installieren
- Netzwerk-Portgruppe für die ESXi VM in den Promiscuous Mode schalten
- um den virtuellen Hardware Virtualisierungsmode (Ja - die Hardware Virtualisierung Unterstützung aktueller CPUs kann wiederum virtuell bereitgestellt werden 🙂) einzuschalten (wichtig für 64 Bit nested VMs) muss der Parameter `vhv.allow = “TRUE”` in die `/etc/vmware/config` des ESXi Hosts gesetzt werden.

Anlegen der VM für die ESXi Installation:

- neue VM mit Hardware Layer 8 und Typ `Linux Redhat 5 64Bit`
- 2 CPUs und mindestens 2 GB RAM (es geht auch eine CPU…)
- 1GB Thin Provisioned Storage
- 2 x NICs E1000 (eine würde auch gehen, nur Warnung “no Redundancy” fürs Management)
- nach dem Erstellen der VM
  - `Options→General` Guest System ändern zu `Others…ESXi 5`
  - CPU/MMU Virtualisierung von `Automatic` auf `Intel VT` bzw. `AMD …` stellen

Diese VM kann man vor dem Installieren als Template abspeichern.

Mittels vMotion können 64 Bit VMs auch bei gleichem Shared Storage zu “physischen” Hosts verschoben werden.

Sonstiges :-)

timcarman.net/as-built-report/

https://cormachogan.com/2020/01/10/getting-started-with-vmware-cloud-foundation-vcf/

https://www.virtuallyghetto.com/2018/07/new-sddc-certificate-replacement-fling.html

https://www.virten.net/vmware/vsphere-version-comparison/

https://virtualpad.wordpress.com/2015/09/11/check-if-your-vms-are-swapping-fast-checking/

https://blogs.vmware.com/vsphere/2016/05/load-balancing-vsphere-clusters-with-drs.html

https://www.storagereview.com/review/how-to-raspberry-pi-as-a-vsan-witness

Zertificatsproblem VCSA https://kb.vmware.com/s/article/59555
https://web.vmware-labs.com/scripts/check-trust-anchors
https://www.ferroquesystems.com/resource/issue-vcenter-7-and-vsan-unable-to-extract-requested-data-trust-anchor-errors/

1)
bei EqualLogic gibt es ein wunderbares Programm, was die Einrichtung von iSCSI und dem EqualLogic Multipathing Modul erlaubt. Wurde hier aber nicht genutzt.