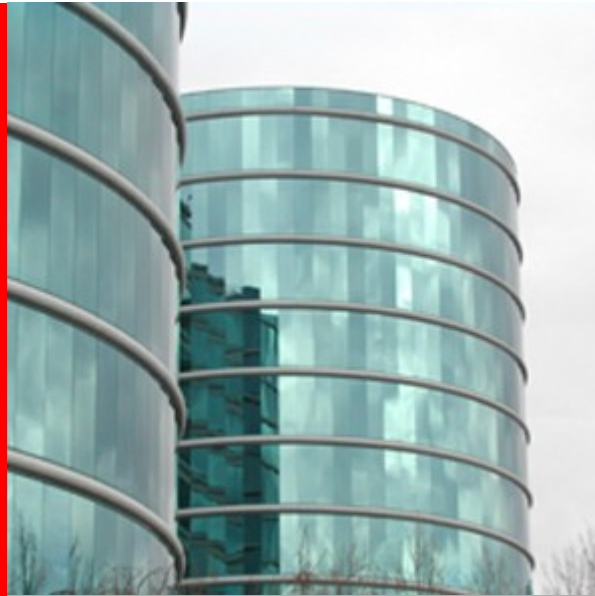


ORACLE®



ORACLE®

**Minimale Downtime -
beim Patchen von Failover (Flying) Oracle Solaris Containern**

Detlef Drewanz, Hartmut Streppel
Principal Sales Consultants
Oracle Deutschland

Hardware and Software

ORACLE®

Engineered to Work Together

Agenda

- The Patching Drama
- Ein Lösungsvorschlag
- Anwendung auf Failover Container im Cluster
- Zusammenfassung
- Fragen und Antworten

Oracle Solaris Zonen

Containers

ORACLE®

SOLARIS

SPARC & x86

Patchen von Systemen mit vielen Zonen

- Patchen erfordert sorgfältige Update Strategie
- Systeme mit vielen installierten Zonen benötigen eine(!) gemeinsame Downtime für alle Services in den Zonen
- Solche Planungen sind schwer zu koordinieren

Herausforderungen

- Viele installierte Zonen, aber minimale Downtime ?
- Updates Zone für Zone möglich ?
- Wie verhält sich die Anwendung beim Update ?
- Automatisierung des Update Prozesses ?
- Fallback möglich ?

Wie werden lokale Zonen unter Solaris gepatcht?

- Lokale Zonen müssen den gleichen Patch- und Paketstand haben wie die globale Zone
- Patch-Werkzeuge erzwingen dies
 - Patchen der globalen Zone bewirkt Patchen der lokalen Zone
- Wie werden „importierte“ Zonen auf den Stand der globalen Zone gebracht?
 - Update-on-attach oder „`zoneadm -z <name> -U`“
 - Vergleicht Patch- und Paketstände und führt Update durch
 - Downgrade nicht möglich

Standard Patch-Methoden für lokale Zonen

- Standard Upgrade
 - Vollständige Downtime des Systems
 - Nicht mehr möglich mit S10 8/11
- Live Upgrade
 - Patchen der globalen und der lokalen Zone im laufenden Betrieb
 - Erzeugen einer alternativen Bootumgebung
 - Aktivieren durch einen Reboot → benötigt eine Downtime für alle Zonen

Alternative Patch-Methoden für lokale Zonen

- Patchen einer Kopie einer lokalen Zone
 - Live Upgrade
 - Klonen der Zone und nachfolgendes Update-on-attach
- Erzeugen einer neuen, identischen lokalen Zone
 - Kopie einer Referenzzone und nachfolgender Update
 - Geskriptetes Erzeugen einer neuen Zone auf einem System mit neuerem Softwarestand
- **ACHTUNG:** Anwendungsdaten sollten immer auf einem getrennten Dateisystem abgelegt werden, das bei Bedarf an die neue Version der Zone angehängt wird

Alternative Patch-Methoden für Zonen

Update process	Effort to Copy the Zone	Survival of Zone-customizations	Zone-by-Zone update possible	Second system USED
Live Upgrade	Little, lucreate command	copy through Live Upgrade	no	no
Clone and update the zone	little, clone the zone	clone customizations	yes	yes
Update a copy of a reference zone	Medium, maintain sample zone, copy sample zone	maintain in sample zone	yes	yes
Create a personalized zone on updated system	High, maintain the script, create the zone by script	maintain by updating the script	yes	yes

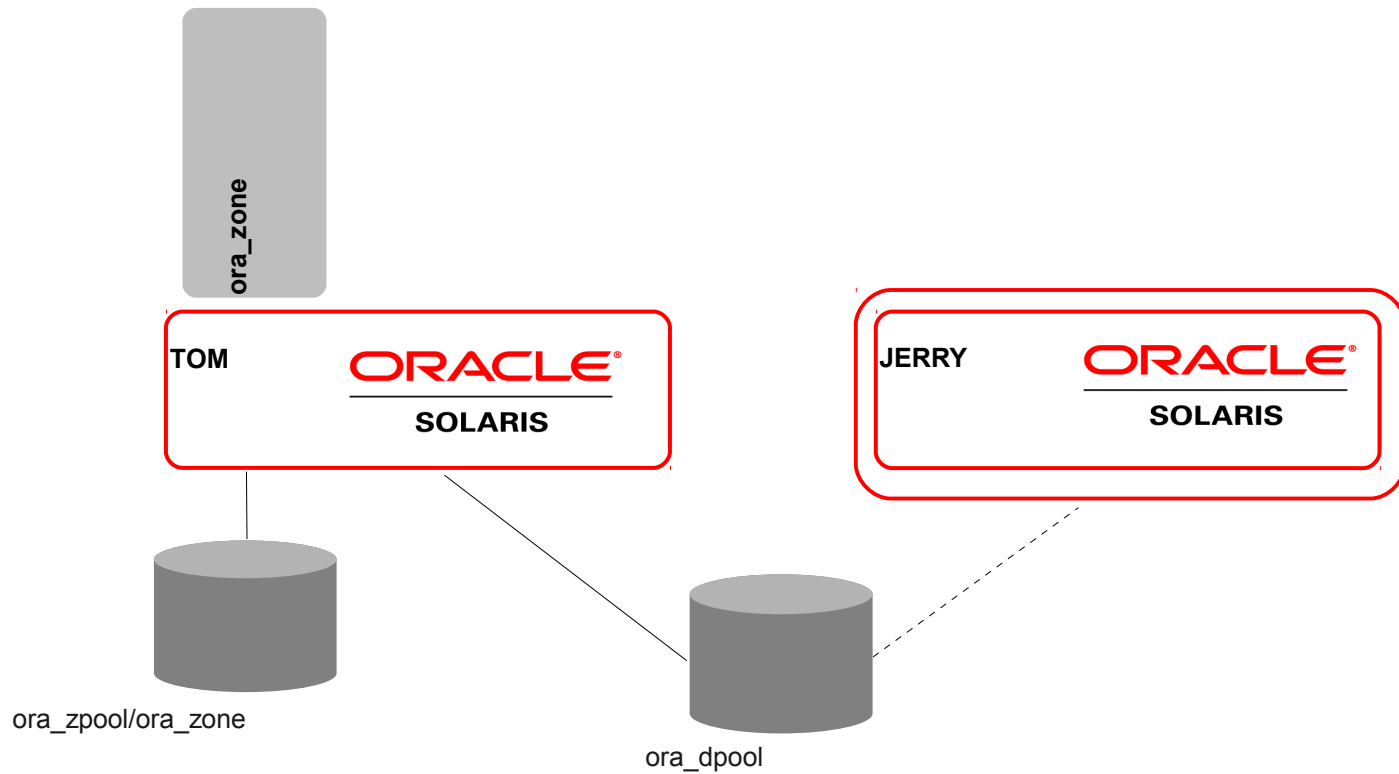
Agenda

- The Patching Drama
- **Ein Lösungsvorschlag**
- Anwendung auf Failover Container im Cluster
- Zusammenfassung
- Fragen und Antworten

Konfigurationsmodell

- TOM
 - Aktuelles Produktivsystem mit ora_zone
- JERRY
 - System mit neuerem Patch-Stand
- Je Zone ein zpool: ora_zpool
- Dataset für Zoneroot: ora_zpool/ora_zone
- Anwendungsdaten in einem separaten Pool: ora_dpool
 - ora_dpool von TOM oder JERRY importierbar

Konfigurationsskizze

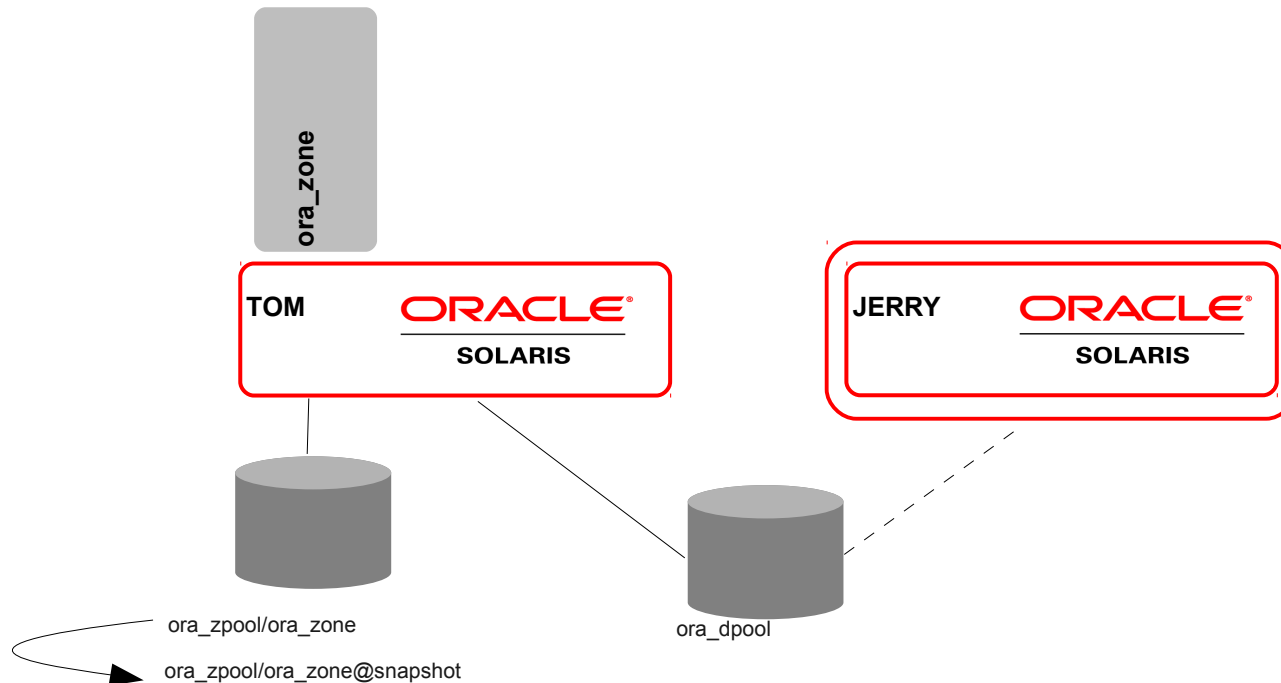


Methode in 5 Schritten

- Snapshot der laufenden Produktionszone
- Klonen des Snapshots auf separater LUN
- Update der geklonten Zone
- Booten und Testen der „neuen“ Zone
- Rollentausch
 - Stoppen der Produktionszone
 - Umhängen des Daten zpools
 - Starten des Klons
- Downtime?
 - Nur während des Rollentauschs :-)

Schritt 1: ZFS Snapshot der produktiven Zone

- Phy. Snapshot mit ZFS

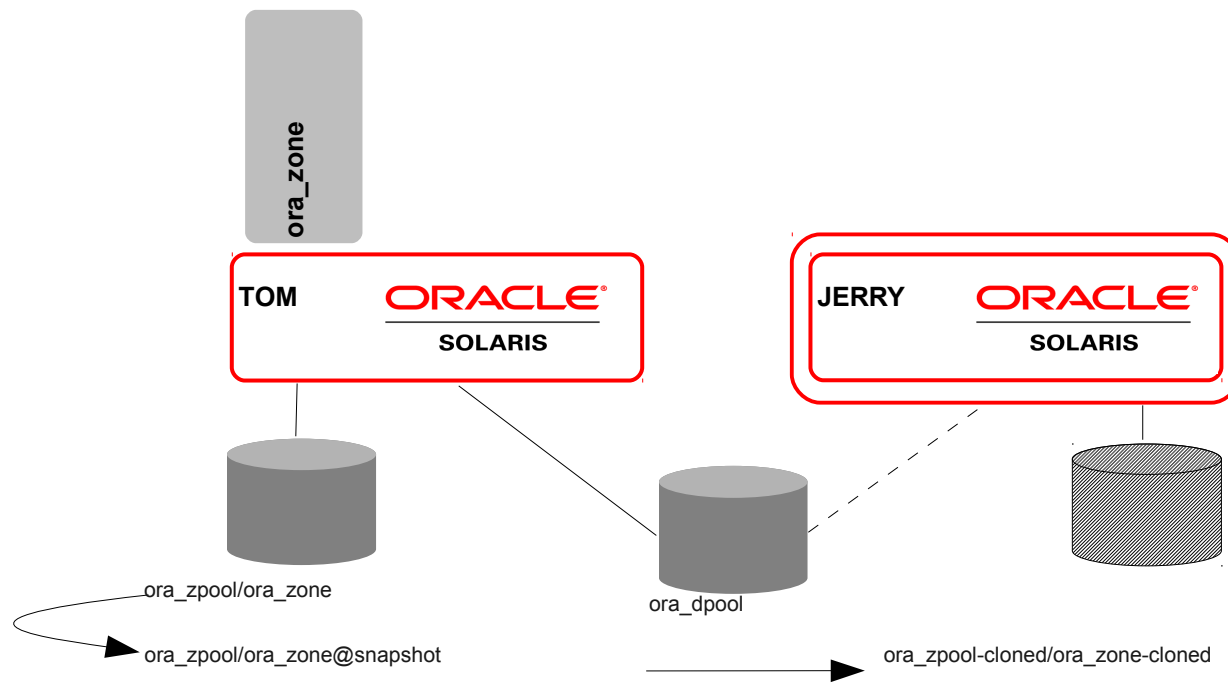


Schritt 1: Die Kommandos

```
[TOM:root] zoneadm -z ora_zone detach -n > /zones/ora_zone/SUNWdetached.xml  
[TOM:root] zfs snapshot ora_zpool/ora_zone@snapshot  
[TOM:root] rm /zones/ora_zone/SUNWdetached.xml
```

Schritt 2: Cloned Zoneroot erzeugen

- Aus snapshot dataset in neuem zpool erzeugen
- ora_zpool_cloned zu JERRY verschieben



Schritt 2: Die Kommandos

```
[TOM:root] zpool create -m none ora_zpool-cloned \  
            c3t600A0B8000347ED600000C654DE3598d0
```

```
[TOM:root] zfs send -p ora_zpool/ora_zone@snapshot | \  
            zfs receive -u ora_zpool-cloned/ora_zone-cloned
```

```
[TOM:root] zfs destroy ora_zpool/ora_zone@snapshot
```

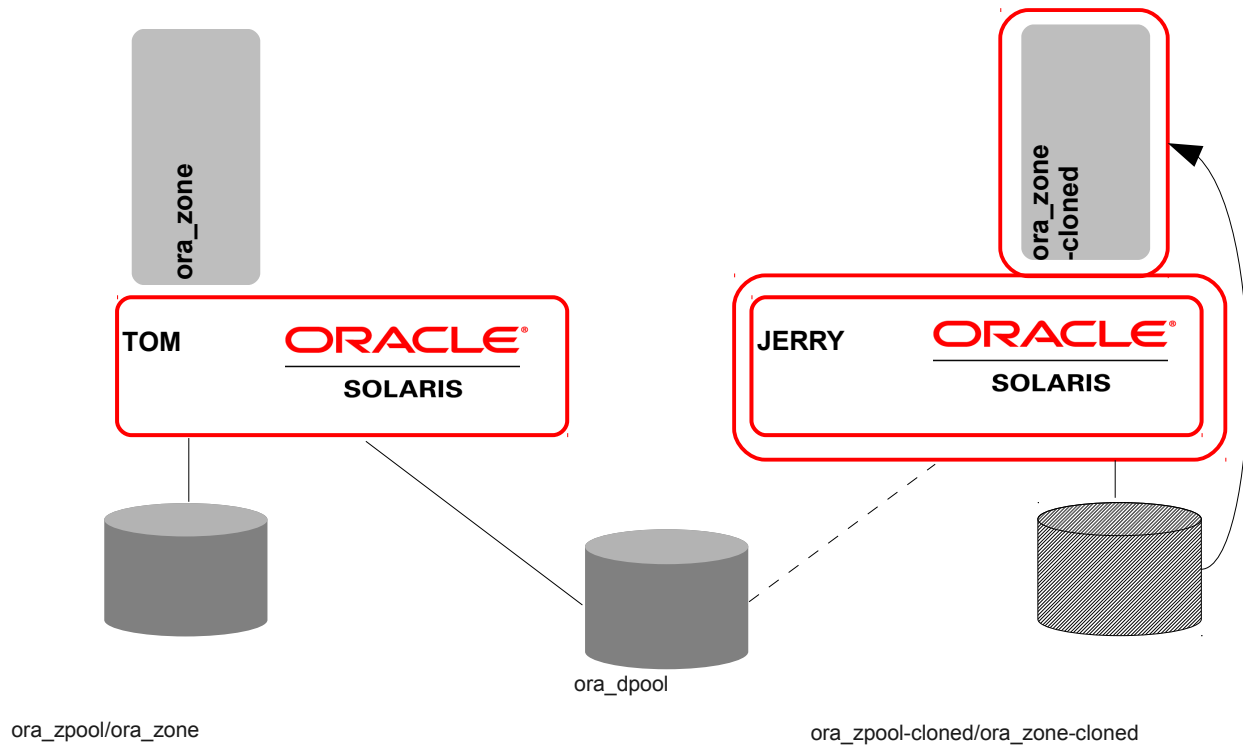
```
[TOM:root] zfs destroy ora_zpool-cloned/ora_zone-cloned@snapshot
```

```
[TOM:root] zpool export ora_zpool-cloned
```

```
[JERRY:root] zpool import ora_zpool-cloned
```

Schritt 3: Erzeugung und Update der Zone

- Update-on-attach der cloned-zone
- Konfiguration der Zone beachten



Schritt 3: Die Kommandos

```
[JERRY:root] zonecfg -z ora_zone-cloned create -a /zones/ora_zone
```

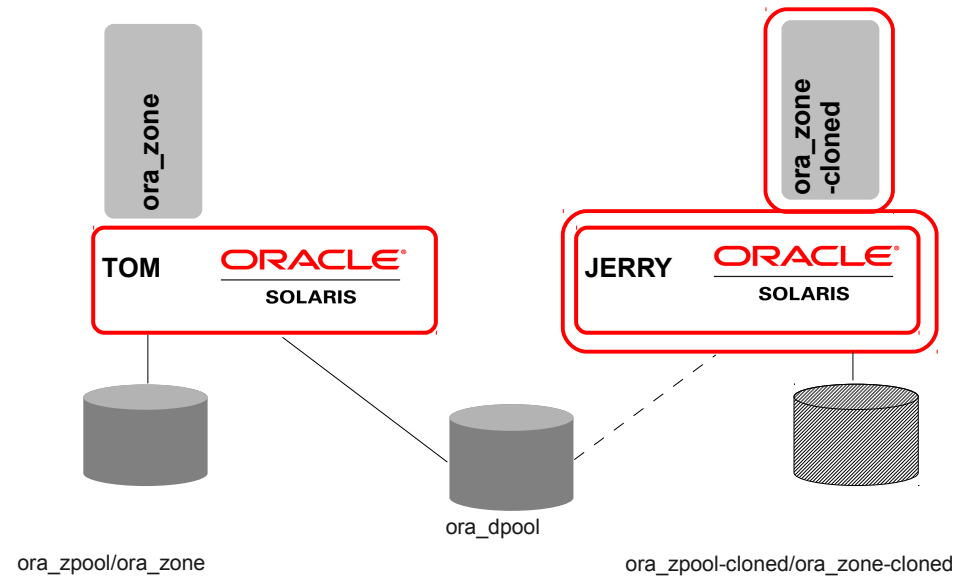
```
[JERRY:root] mkfile 64M /tmp/tmp_dpool
```

```
[JERRY:root] zpool create -m none ora_dpool /tmp/tmp_dpool
```

```
[JERRY:root] zoneadm -z ora_zone-cloned attach -U
```

Schritt 4: Boot und Test der Cloned Zone

- Testdaten bereitstellen
- Zone booten
- Zone stoppen
- Testdaten entfernen



Schritt 4: Die Kommandos

```
[JERRY:root] zoneadm -z ora_zone-cloned boot
```

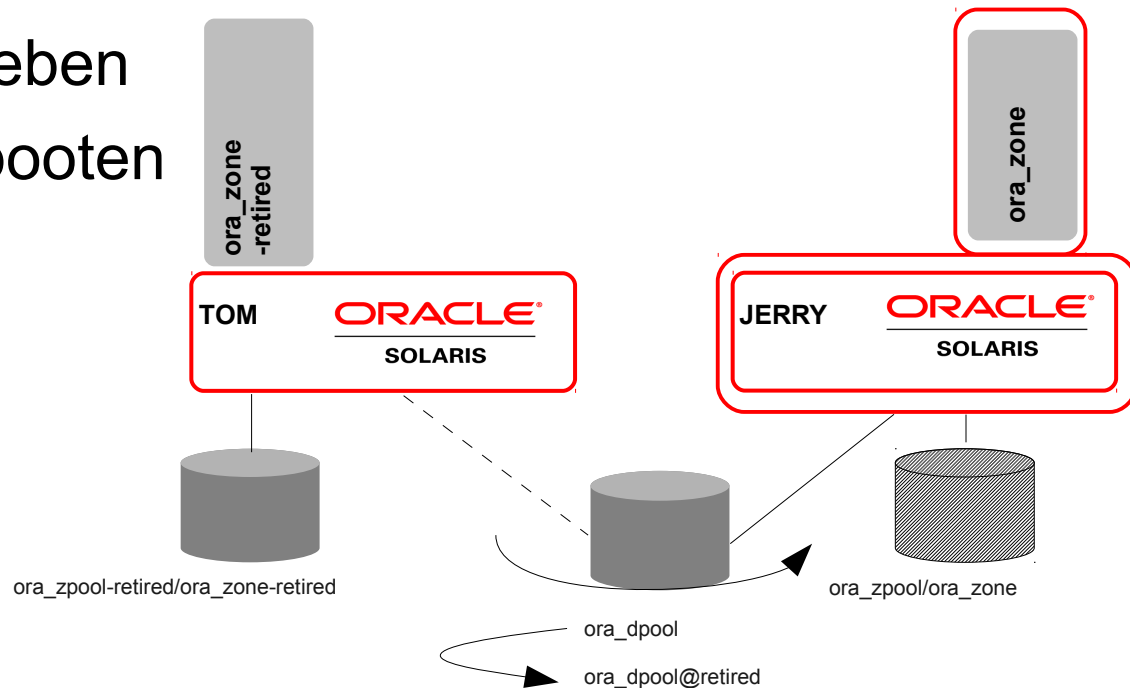
```
[JERRY:root] zlogin ora_zone-cloned init 5
```

```
[JERRY:root] zpool destroy ora_dpool
```

```
[JERRY:root] rm /tmp/tmp_dpool
```

Schritt 5: Rollentausch der Zonen (Downtime)

- Snapshot der Produktionsdaten
- Zpools umbenennen
- ora_zone data sets umbenennen
- ora_dpool verschieben
- Production Zone booten



Schritt 5: Die Kommandos

```
[TOM:root] zlogin ora_zone init 5
```

```
[TOM:root] zfs snapshot -r ora_dpool@retired
```

```
[TOM:root] zpool export ora_dpool
```

```
[TOM:root] zfs rename ora_zpool/ora_zone ora_zpool/ora_zone-retired
```

```
[TOM:root] zpool export ora_zpool
```

```
[TOM:root] zpool import ora_zpool ora_zpool-retired
```

```
[TOM:root] zonecfg -z ora_zone set zonename=ora_zone-retired
```

```
[JERRY:root] zonecfg -z ora_zone-cloned set zonename=ora_zone
```

```
[JERRY:root] zpool export ora_zpool-cloned
```

```
[JERRY:root] zpool import ora_zpool-cloned ora_zpool
```

```
[JERRY:root] zfs rename ora_zpool/ora_zone-cloned ora_zpool/ora_zone
```

```
[JERRY:root] zpool import ora_dpool
```

```
[JERRY:root] zoneadm -z ora_zone boot
```

Agenda

- The Patching Drama
- Ein Lösungsvorschlag
- **Anwendung auf Failover Container im Cluster**
- Zusammenfassung
- Fragen und Antworten

Failover Zone unter Oracle Solaris Cluster Kontrolle

- Was ist anders bei Failover Zonen unter Oracle Solaris Cluster Kontrolle
 - Zonenkonfiguration ist auf allen Knoten bekannt
 - Nodelist Property der Ressourcegruppe muss angepasst werden
 - Beim Rollentausch ist nur eine Änderung der HAStoragePlus Ressource notwendig; dann kann die Ressourcegruppe geschwenkt und die „neue“ Zone gestartet werden
 - Fallback durch Rücksetzen der HAStoragePlus Ressource

Rollentausch bei Zone unter Cluster-Kontrolle

```
echo "START of downtime for service at `date`"
```

```
echo "Stopping production zone"
```

```
1. clrs disable ${ZONENAME}-rs
```

```
echo "Setting nodelist to all nodes to allow for storage failover"
```

```
2. clrg set -p nodelist=$OLD_NODE,$NEW_NODE ${ZONENAME}-rg
```

```
echo "Changing the HASP resource to include the new cloned zpool"
```

```
3. clrs set -p Zpools=$ZONEPOOLNAMECLONED,$DATAPOOLNAMECLONED ${ZONENAME}-hasp
```

```
echo "Switching over the rest of the zone"
```

```
4. clrg switch -n $NEW_NODE ${ZONENAME}-rg
```

```
echo "Resetting the nodelist to prevent fallback to old node"
```

```
5. clrg set -p nodelist=$NEW_NODE ${ZONENAME}-rg
```

```
echo "Starting the zone resource"
```

```
6. clrs enable ${ZONENAME}-rs
```

```
echo "Back online with patched (cloned) zone at `date`"
```

Fallback bei Zone unter Cluster-Kontrolle

```
echo "START of downtime for service at `date`"

echo "Stopping production zone"
clrs disable ${ZONENAME}-rs
echo "Setting nodelist to all nodes to allow for storage failover"
clrg set -p nodelist=$OLD_NODE,$NEW_NODE ${ZONENAME}-rg
echo "Changing the HASP resource to include the new cloned zpool"
clrs set -p Zpools=$ZONEPOOLNAME,$DATAPOOLNAME ${ZONENAME}-hasp
echo "Switching over the rest of the zone"
clrg switch -n $OLD_NODE ${ZONENAME}-rg
echo "Resetting the nodelist to prevent fallback to old node"
clrg set -p nodelist=$OLD_NODE ${ZONENAME}-rg
echo "Starting the zone resource"
clrs enable ${ZONENAME}-rs

echo "Failed back to old zone at `date`"
```

Agenda

- The Patching Drama
- Ein Lösungsvorschlag
- Anwendung auf Failover Container im Cluster
- **Zusammenfassung**
- Fragen und Antworten

Zusammenfassung

- Hochkonsolidierte Umgebungen kennen keine Downtime
- Einfache Methode, Zonen einzeln auf einen neuen Softwarestand zu bringen
 - Ohne und mit Oracle Solaris Cluster
- Minimale Downtime
 - Vom Stoppen der Produktionszone bis Starten der geklonten Zone
- Randbedingungen müssen beachtet werden

Mehr Informationen

- „Minimizing Downtime While Updating Oracle Solaris Containers in Clustered and Non-Clustered Environments“
- Verweis auf andere DOAG Präsentationen
 - Solaris Life Cycle Engineering, Gerry Haskins
 - Oracle Solaris Software Updates, Thomas Wagner
- Solaris Dokumentation

Agenda

- The Patching Drama
- Ein Lösungsvorschlag
- Anwendung auf Failover Container im Cluster
- Zusammenfassung
- Fragen und Antworten

ORACLE®