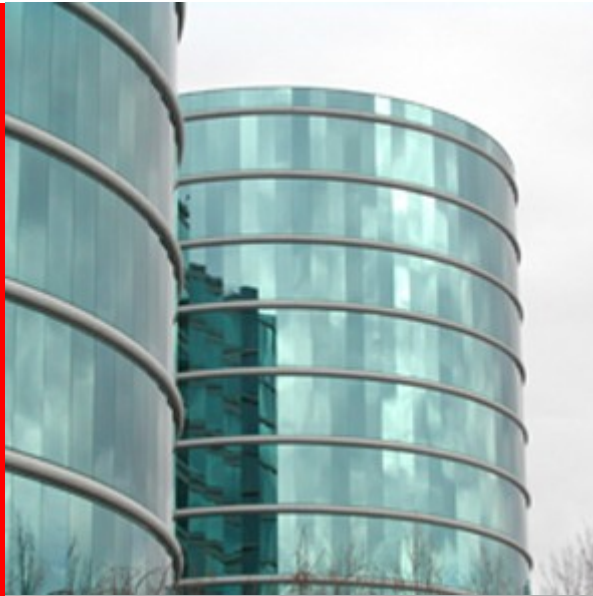


ORACLE®



**ORACLE®**

**Flying Container -  
mit Oracle Solaris Container und Oracle Solaris Cluster**

Hartmut Streppel  
Principal Sales Consultant  
Oracle Deutschland

# Hardware and Software

ORACLE®

# Engineered to Work Together

# Agenda

- Oracle Solaris Container
- Oracle Solaris Cluster
- Oracle Solaris Cluster und Failover Container
- Zusammenfassung
- Fragen und Antworten

# Oracle Solaris Zonen

Containers

**ORACLE®**

SOLARIS

**SPARC & x86**

# Oracle Solaris Container

- Container sind eine Ablaufumgebung für Anwendungen jeder Art mit folgenden Eigenschaften
  - Sicherheit
  - Kontrolle der verwendeten Ressourcen
  - Flexibilität
  - Volle Performance
  - Virtuelles Betriebssystem
  - Ohne Virtualisierungsschicht (kein Hypervisor)
- Container sind keine virtuellen Maschinen!

# Services in Containern – Warum?

- Eingeschränkte Rechte für höhere Sicherheit
  - Sicherheitskonfiguration und RBAC individuell pro Container
  - Kein direkter Zugriff auf Kernel oder “fremden” Hauptspeicher
  - Keine Manipulation am Netzwerk\* oder Kernel-Modulen
- Durchsetzung harter Sicherheitsgrenzen aus der Global Zone
  - Konfigurierbare Rechte, Sparse Root Zones, IP Instances, IP Filter, etc.
- Resource Control und Management
  - CPU, Memory, Disk, Netzwerk, Devices, etc.
- Sichere Beobachtbarkeit
  - BART\*\*, Solaris Auditing, etc.
- \*: Nur bei shared IP Konfigurationen
- \*\*: siehe auch <http://www.c0t0d0s0.org/archives/4069-Less-known-Solaris-features-BART.html>

# Warum ?

- Einfacheres Change-Management – nur eine Umgebung zu verwalten
- Kein Performance Overhead
- Hoher Grad der Standardisierung notwendig
- Möglichkeit der „Management Delegation“



# „Spezielle“ Container

- Branded Zones
  - Container für “andere” Betriebssysteme
    - Solaris 8 & 9 (nur SPARC)
    - Solaris 10 (für Solaris 11)
    - Cluster – für Container Cluster
- “Capped Container” als Lizenzgrenze für Oracle
  - Was ist ein Capped Container?
- Container als Cluster-Knoten
  - Failover Zones
  - Flying Container
  - .....

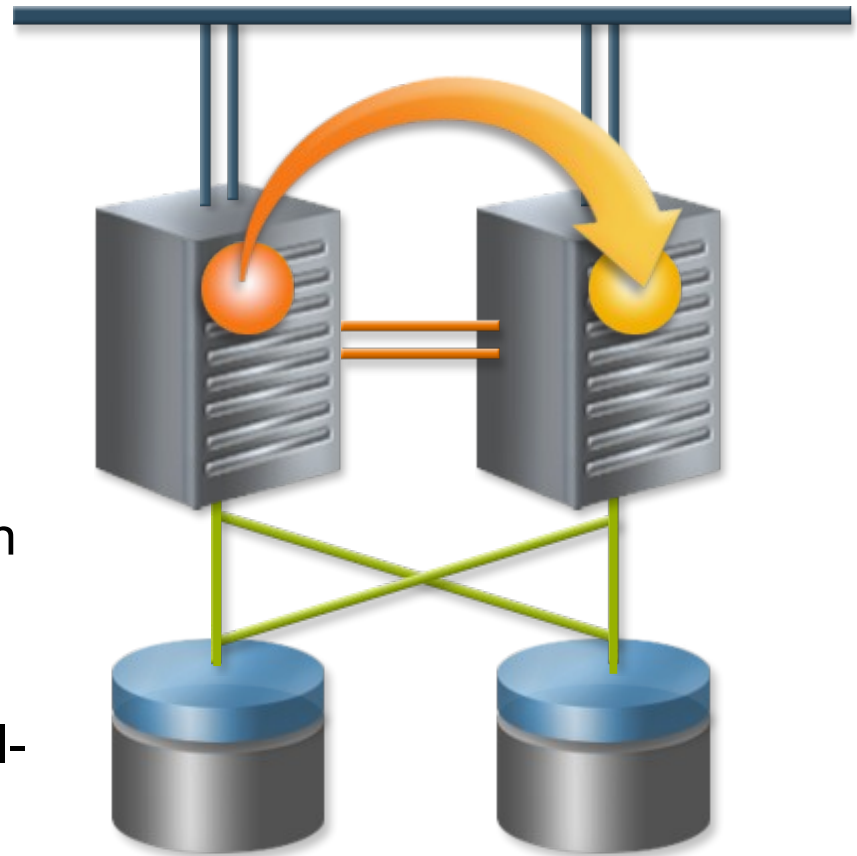
# Agenda

- Oracle Solaris Container
- **Oracle Solaris Cluster**
- Oracle Solaris Cluster und Failover Container
- Zusammenfassung
- Fragen und Antworten

# Oracle Solaris Cluster

## Keeps Data Centers Running 24x7

- Höchste Verfügbarkeit durch Hardware-Redundanz und ausgefeilte Software-Algorithmen
- Überwacht das Cluster
  - Server
  - Storage
  - Netz
  - Anwendungen
- Automatisches Recovery des Clusters und der Anwendungen in Fehlersituationen
- Ermöglicht (fast) “zero-downtime” Wartung durch “rolling” und “dual-partition” Upgrades



# Oracle Solaris Cluster

- Integration in den Solaris Betriebssystemkern
- enge Koppelung der Clusterknoten
- sicheres Quorum- und Membership-Konzept
- umfassendes Framework für die Anwendungsintegration
- große Anzahl vorgefertigter Agenten für Anwendungen
- Integration in Disaster Recovery Infrastrukturen mit der Oracle Solaris Cluster Geographic Edition
- Aktuelle Version: 3.3 5/11

# Agenda

- Oracle Solaris Container
- Oracle Solaris Cluster
- Oracle Solaris Cluster und Failover Container
- Zusammenfassung
- Fragen und Antworten

# Einige grundlegende Annahmen

## Für Fortgeschrittene

- Der Zonenrootpfad enthält nur das „Betriebssystem“
  - Keine Anwendungen
    - Darüber kann man streiten
  - Keine Daten
- Ein Zpool pro Zone
  - Zonenpfad liegt unter zpool/zonenroot!!!
    - Notwendig für Live Upgrade
  - Eingehängt nach Belieben
- Zonen werden ohne manuellen Eingriff erstellt
  - zfs clone
  - Skript

# Fliegende Container – oder: Failover Zonen

- Failover Zone / HA Container Agent

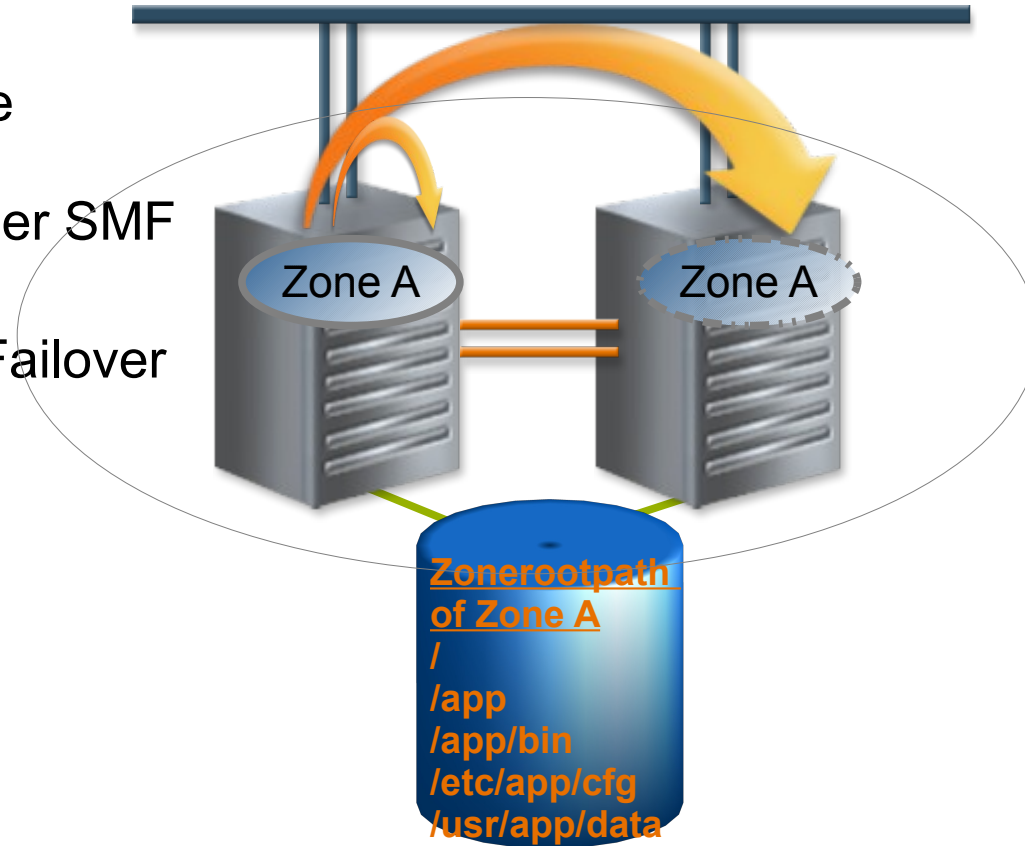
- Zone ist eine Ressource
- Zone wird überwacht, nicht die Anwendung
  - Integration von Skripten oder SMF Diensten möglich
- Fehler triggert Neustart oder Failover der gesamten Zone

- Pro

- Verwalten nur einer Zone
- Management Delegation

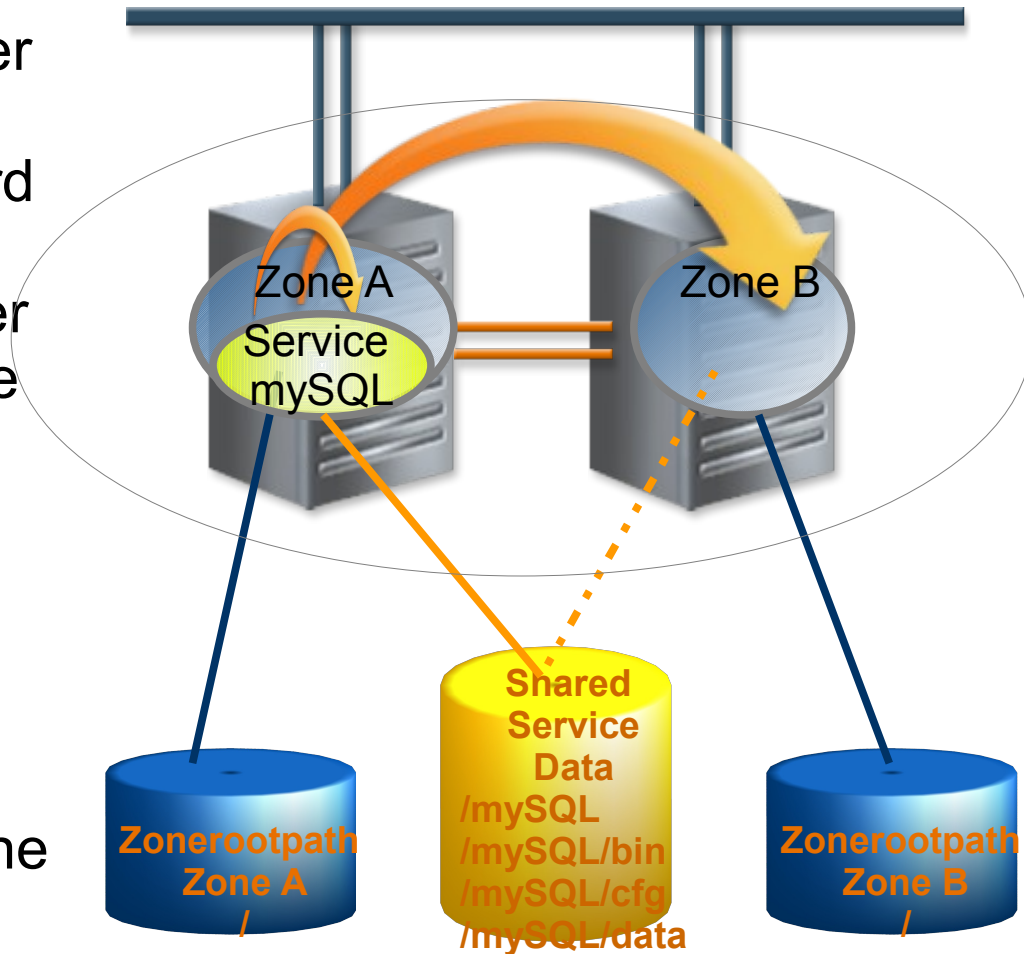
- Contra

- 1 Zone ist ein SPOF
- Aufwändiger zu patchen



# Flying Services – Zone Nodes

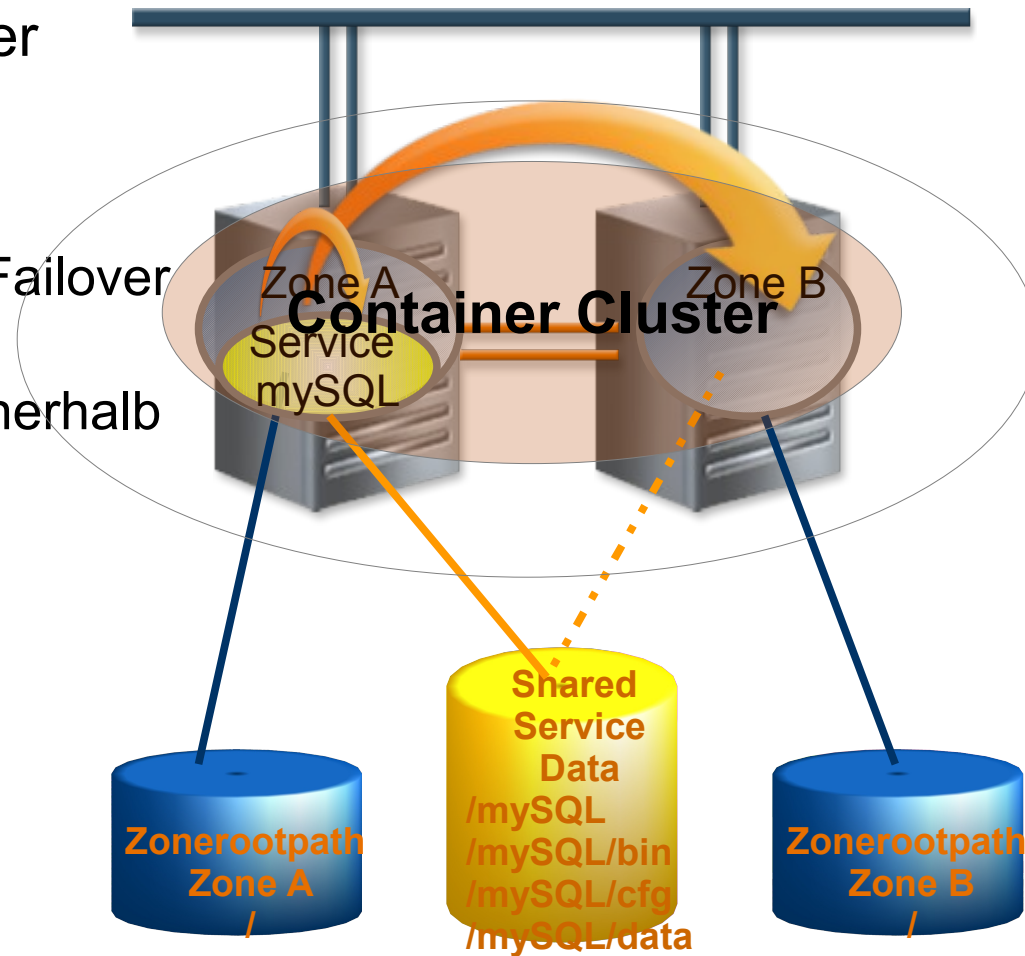
- Zone Node
  - Zone ist ein virtueller Cluster Knoten
  - Anwendung in der Zone wird überwacht
  - Fehler bewirkt Neustart oder Failover in die andere Zone
- Pro
  - Einfache Patch Prozedur
  - Standard Cluster Agenten
  - Weniger Abhängigkeiten
- Contra
  - 2 Container zu verwalten
  - Cluster „sichtbar“ in der Zone





# Flying Services – Oracle Solaris Container Cluster aka Zone Cluster

- Zone Cluster
  - Zone ist ein virtueller Cluster Knoten
  - Anwendung in der Zone wird überwacht
  - Fehler triggert Neustart oder Failover der gesamten Zone
  - Volle Cluster Funktionalität innerhalb des Zonen Clusters
- Pro
  - Management delegation
  - „Easy to use“ cluster
- Contra
  - 2 Container zu verwalten
  - Etwas komplexer bei der Konfiguration



# Agenda

- Oracle Solaris Container
- Oracle Solaris Cluster
- Oracle Solaris Cluster und Flying Container
- Zusammenfassung
- Fragen und Antworten



# Failover Zonen und der HA Container Agent

- Voraussetzung
  - Zonenrootpfad auf shared Storage
  - Sparse root oder full root?
    - Beides geht
    - Full root hat weniger Einschränkungen beim Patchen
  - autoboot=false (Boot nur(!) durch das Cluster)
- HA Container ist der Agent für Failover Zonen
  - GDS basiert
    - sczbt
  - Zwei „Hilfs“-agenten
    - sczsh
    - sczsmf

# HA Container Agent

## sczbt, sczsh und sczsmf

- sczbt – Zone boot
  - Startet, überwacht und stoppt eine Failover Zone
- sczsh – Zone Shell
  - Hängt ab von sczbt
  - Startet, überwacht und stoppt eine Ressource in einer Failover Zone mit Hilfe von Skripten
- sczsmf – Zone SMF
  - Hängt ab von sczbt
  - Startet, überwacht und stoppt eine SMF Ressource in einer Failover Zone

# Starten und Stoppen der Zone durch HA Container

- Start
  - zoneadm ... boot
  - Ausführen des Start-Skripts für sczsh
  - svcadm enable für sczsmf
- Stopp sczbt
  - zoneadm ..... shutdown -y -g0 -i0
  - Wenn 60% von Stop\_timeout verbraucht sind: zoneadm ... halt
- Stopp sczsh
  - Nutzt das Stopp-Skript zum Stoppen des Dienstes
- Stopp sczsmf
  - svcadm ... disable
  - Nach Ablauf von 60% von Stop\_timeout, kill -SIGTERM, dann kill - SIGKILL

# Überwachung der Zone durch HA Container

## sczbt – Zone Boot Variante

- „The corresponding zsched process for the zone is running.“
- „Every logical hostname that is managed by a SUNW.LogicalHostname resource is operational.“
- „The specified milestone for the native zone brand type is either online or degraded.“
- „If the milestone is not online or degraded, or if the specified run level is not active, the fault monitor restarts the zone. If this fault persists, the fault monitor fails over the resource group that contains resource for the zone boot component.“

# Stoppen einer Anwendung

- Typischerweise durch ein Stopskript
- Default, falls nicht verfügbar: kill -15
- Achtung
  - Es muss sichergestellt sein, dass
    - alle Anwendungskomponenten beendet sind
- Achtung
  - Wenn eine Anwendung auf einem Knoten nicht vollständig gestoppt ist, kann es bei einem Neustart auf einem anderen Knoten zu Datenkorruption kommen
- Achtung auf Property „failover\_mode“
  - Wenn Stoppen fehlschlägt, kann es zu einem Node Panic kommen (in Abhängigkeit von failover\_mode)

# Migration einer Anwendung

- Grundsätzlicher Ablauf
  - Stoppen des Containers
  - Stoppen der anderen Ressourcen, Exportieren der Zpools/Metasetes
  - Import der Zpools/Metasetes auf anderem Knoten
  - Booten der Zone
- Kein detach/attach
  - detach/attach zu zeitaufwändig
  - Was, wenn attach fehlschläge?



# Flying Container mit Solaris Cluster

## Ein Beispiel

- Notwendiger SC Agent: HA Container Agent
- Notwendiges Paket: SUNWsczone
- Paketverzeichnis: /opt/SUNWsczone/sczbt/
- Kopieren der Konfigurationsdatei für guug\_zone
- Registrierung des gds Resource Typs notwendig
  - `clrs register SUNW.gds`

# Flying Container (2)

- Architektur

- Eine Ressource-Gruppe – *Fzone\_rg* - mit

- einer sczbt (d.h. SUNWgds) Ressource - *Fzone\_rs*

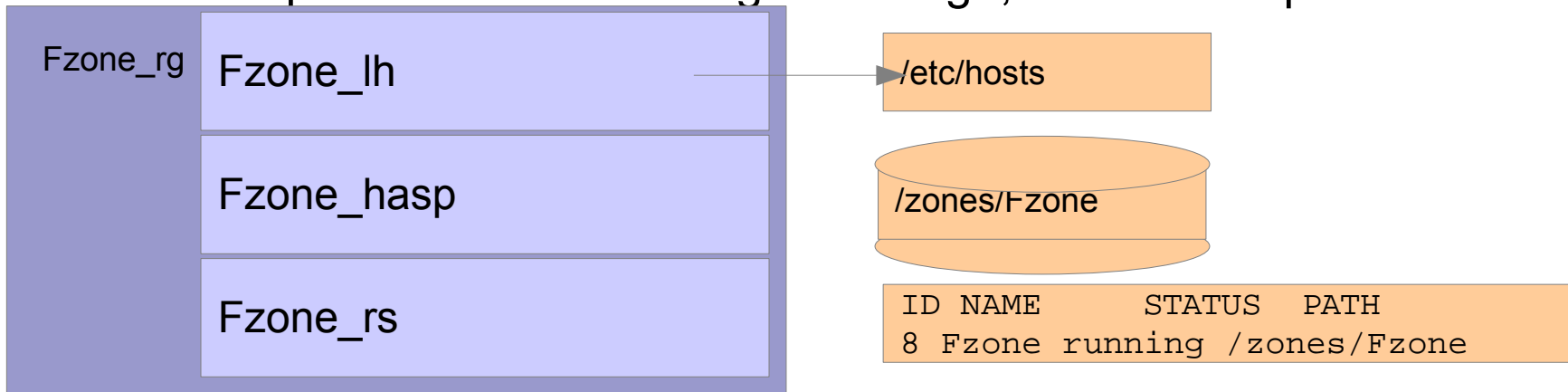
- verwaltet die Zone, d.h. bootet, stoppt, verschiebt, überwacht

- einer HALogicalHostname Ressource - *Fzone\_lh*

- verwaltet die IP-Adresse der Zone, die auch von Anwendungen verwendet werden kann

- einer HAStoragePlus Ressource - *Fzone\_hasp*

- importiert den notwendigen Storage, z.B. einen Zpool



# Flying Container (3)

- Anlegen der Ressource Gruppe und der Ressourcen
  - Vorher Zone stoppen

```
[CL1-b:root] zoneadm -z Fzone_zone halt
[CL1-b:root] clrg create Fzone_rg
[CL1-b:root] clrslh create -g Fzone_rg -h Fzone_lh Fzone_lh
[CL1-b:root] clrslh create -g Fzone_rg -t SUNW.HAStoragePlus \
-p Zpools=Fzone Fzone_hasp
[CL1-b:root] clrg online -M guug_rg
```

```
[CL1-b:root] clrslh status -g Fzone_rg
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
-----	-----	-----	-----
Fzone_lh	CL1-b	Online	Online - LogicalHostname online.
	CL1-a	Offline	Offline - LogicalHostname offline.
Fzone_hasp	CL1-b	Online	Online
	CL1-a	Offline	Offline

## Flying Container (4)

- Notwendiger Status des Containers auf allen(!) Knoten:  
installed
  - ACHTUNG: Bitte keine Dateien unter /etc/zones editieren oder kopieren
  - Detach und Attach Mechanismen nutzen
- Ergebnis: Status des Containers auf allen(!) Knoten:  
installed

## Flying Container (5)

- Achtung: RG immer noch ohne Zone!!!
- Anlegen der HA Container Ressource durch
  - Eintragen der notwendigen Informationen in die Konfigurationsdatei
  - Ausführen des Registrierungsskripts
  - HA Container Agent benötigt ein Verzeichnis zum Speichern von Parametern, typischerweise unter <zonepath>/pfiles

```
[CL1-b:root] cd /opt/SUNWsczone/sczbt/util
[CL1-b:root] vi sczbt_Fzone_config
...
RS=Fzone_rs
RG=Fzone_rg
PARAMETERDIR=/zones/Fzone
SC_NETWORK=true
SC_LH=Fzone_lh
FAILOVER=true
HAS_RS=Fzone_hasp
Zonename=Fzone
```

# Flying Container (6)

- Registrierung HA Container Ressource und Start

```
[CL1-b:root] ./sczbt_register -f ./sczbt_Fzone_config  
sourcing ./sczbt_Fzone_config
```

```
Registration of resource Fzone_rs succeeded.
```

```
Validation of resource Fzone_rs succeeded.
```

```
[CL1-b:root] clrs status Fzone_rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
-----	-----	-----	-----
Fzone_rs	CL1-b	Offline	Offline
	CL1-a	Offline	Offline

```
[CL1-b:root] clrs enable Fzone_rs
```

```
[CL1-b:root] clrs status Fzone_rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
-----	-----	-----	-----
Fzone_rs	CL1-b	Online	Online
	CL1-a	Offline	Offline

```
[CL1-b:root] zoneadm -z Fzone list -v
```

ID	NAME	STATUS	PATH	BRAND	IP
8	Fzone	running	/zones/Fzone	native	shared

# Flying Container (7)

- Manuell initierter Schwenk des Containers

```
[CL1-b:root] clrg switch -n CL1-a Fzone_rg
```

```
....
```

```
[CL1-a:root] clrs status -g Fzone_rg
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
-----	-----	-----	-----
Fzone_lh	CL1-a	Online	Online - LogicalHostname online.
	CL1-b	Offline	Offline - LogicalHostname offline.
Fzone_hasp	CL1-a	Online	Online
	CL1-b	Offline	Offline
Fzone_zone_rs	CL1-a	Online	Online
	CL1-b	Offline	Offline

```
[CL1-a:root] zoneadm -z Fzone list -v
```

ID	NAME	STATUS	PATH	BRAND	IP
17	Fzone	running	/zones/Fzone	native	shared

# Agenda

- Oracle Solaris Container
- Oracle Solaris Cluster
- Oracle Solaris Cluster und Failover Container
- **Zusammenfassung**
- Fragen und Antworten



# Zusammenfassung

- Oracle Solaris Cluster ist das HA Produkt für Anwendungen
- Zehntausende von Installationen
- Höchstverfügbare Infrastrukturen mit bis zu 99,999% Verfügbarkeit in SLAs
- Viele Agenten als Produkt mit vollem Support erhältlich
- Einfache Integration von Failover Containern mit Hilfe des HA Container Agenten
- Vielfältige Konfigurationsmöglichkeiten

# Mehr Informationen

- „Oracle Solaris Cluster Essentials“, 2010, Tim Read, Prentice Hall
- „Oracle Solaris Cluster Concepts Guide“, Teil der Produkt-Dokumentation
  - [http://download.oracle.com/docs/cd/E18728\\_01/html/821-1254/](http://download.oracle.com/docs/cd/E18728_01/html/821-1254/)
- [blogs.sun.com](http://blogs.sun.com)

# Agenda

- Oracle Solaris Container
- Oracle Solaris Cluster
- Oracle Solaris Cluster und Failover Container
- Zusammenfassung
- Fragen und Antworten

**ORACLE®**