ORACLE VM

ORACLE VIRTUALIZATION

ORACLE

Wie kommt die Intelligenz in mein Oracle VM Template?

Manuel Hoßfeld
Leitender Systemberater, BU Database Technologies

Oracle Deutschland B.V. & Co. KG

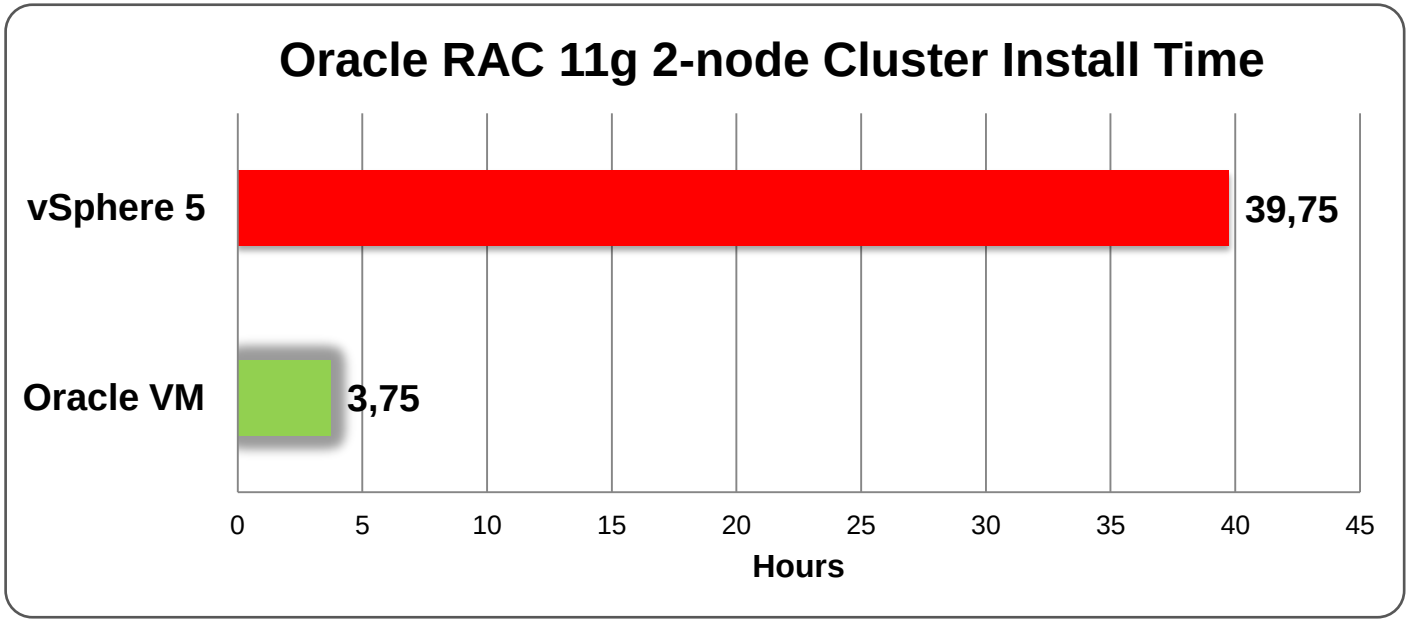
Agenda

- Oracle VM Templates...
  - Warum und Wozu?
  - Eigenschaften und Begrifflichkeiten (Template vs. Assembly...)
- Intelligenz in VM Templates
  - Zielsetzung
  - Implementierung und konkrete Anwendung

Oracle VM Templates

“Die magischen Wesen” (?)

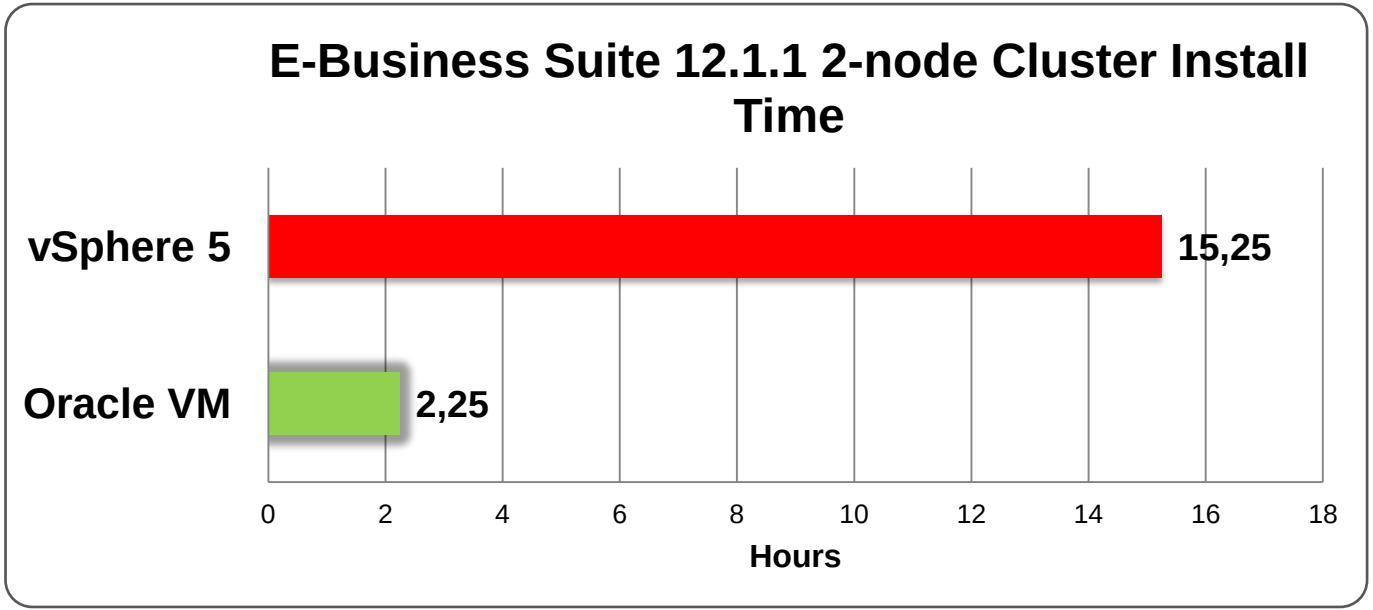
Schnelles Deployment mit OVM Templates – Beispiel 1



Oracle VM Template vs. Traditional Install of Oracle RAC on vSphere 5

Evaluator Group Lab Validation: "Oracle VM – Quantifying The Value of Application-Driven Virtualization"

Schnelles Deployment mit OVM Templates – Beispiel 2



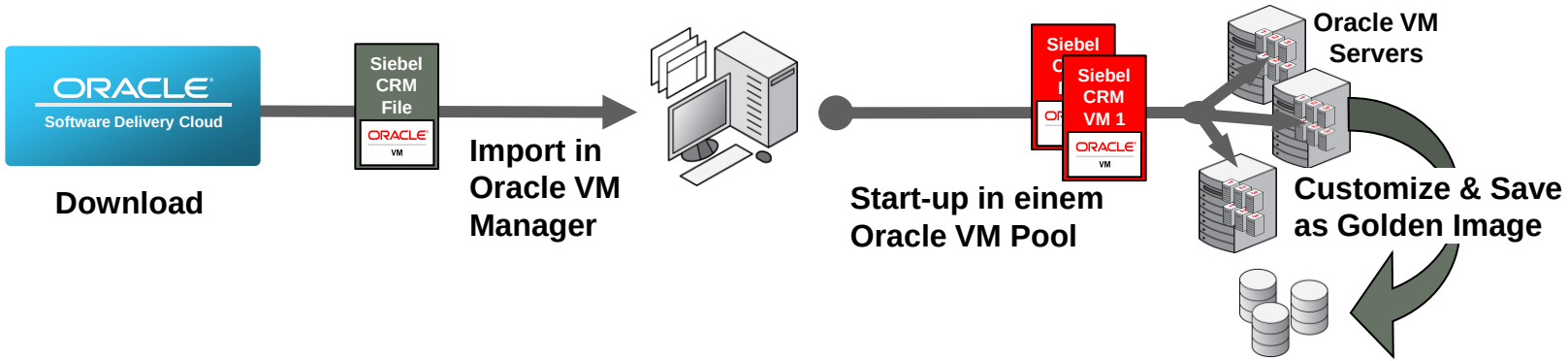
Oracle VM Template vs. Traditional Install of E-Business Suite 12.1.1 on vSphere 5

Evaluator Group Lab Validation: "Oracle VM – Quantifying The Value of Application-Driven Virtualization"

Oracle VM Templates mit fertigen Anwendungen

Schnelles Deployment; 100+ Templates verfügbar

- Pre-built, pre-configured, production-ready VMs
- Apps, Databases, Middleware, OS; Siebel CRM, Oracle RAC, More…
- Nahezu kein Wissen über Installation von OS und Applikation erforderlich
- Bsp.: Automatisiertes Ausrollen eines produktionsreifen 8-Knoten RAC Clusters in wenigen Minuten über eine einfache Konfigurationsdatei



Oracle VM Templates

Eigenschaften & Begrifflichkeiten

Oracle VM Templates & Assemblies

Einleitung – Ziele und Eigenschaften

- Ziel:
  Schnelles, wiederholbares Ausrollen “vorgefertigter” VMs
- Erforderlich:
  Automatisches, individualisiertes Konfigurieren beim ersten Starten der VM
  - Netzwerk-Konfiguration (IP, Name, Gateway…)
  - User-Konfiguration (Passwörter für root, oracle etc.)
  - Ggf. Anwendungs-Konfiguration
- Außerdem:
  - “Rückmeldung” einer VM “nach außen” z.B. über aktuelle IP usw.

Oracle VM Templates & Assemblies

Einleitung – Historie & Nomenklatur

- „*Früher*“: Templates als **.tgz-Dateien**
  - Enthalten virtuelle Platten + vm.cfg (=>Konfigurationsdatei)
  - I.d.R. nur eine VM pro Template
  - Direkt als Basis neuer VMs nutzbar
  - Keine Metadaten über die VMs (abgesehen von vm.cfg) enthalten
  - Meist individuelle, kaum normierte „First-Boot Mechanik“
  - „Rückkanal“ nicht vorhanden (außer selbst implementiert)

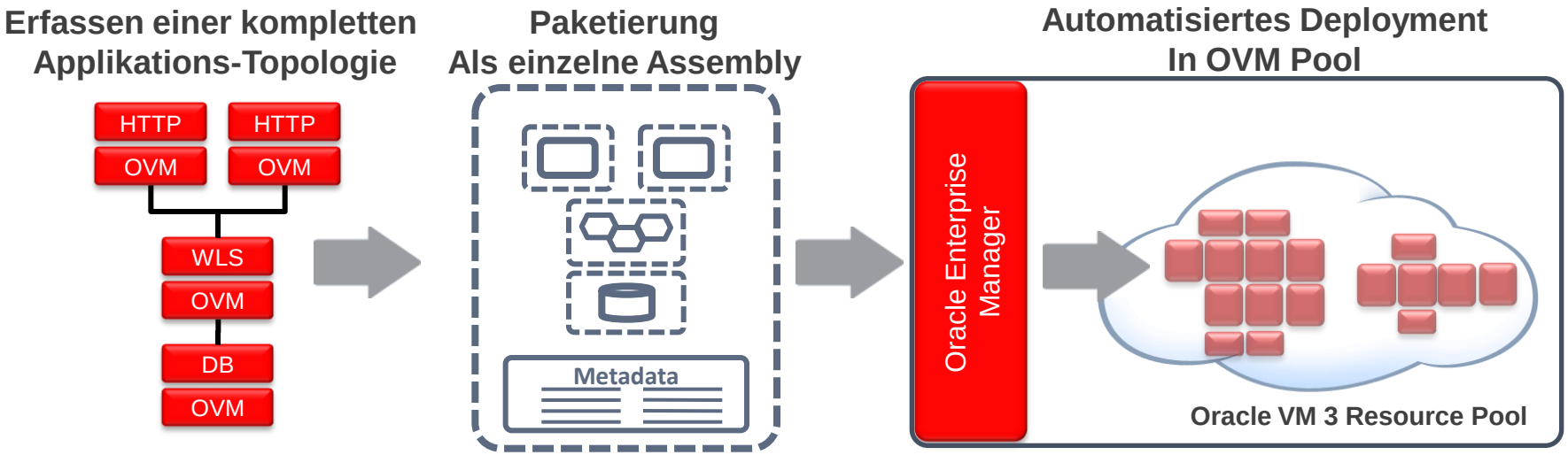
Oracle VM Templates & Assemblies

Einleitung – Historie & Nomenklatur

- „*Jetzt*“: Templates als Assembly im OVF (Open Virtualization Format) – Archiv mit Endung .ova
  - Grundsätzlich eigentlich: Assembly als „Verbundapplikation“ mit mehreren VMs (instanziiert aus Templates) -> **OVAB** = Oracle Virtual Assembly Builder
  - Im Kontext OVM 3.x: Assembly als „Verpackung“ einzelner Templates
- Assemblies direkt nicht als Basis neuer VMs im OVM Mgr. nutzbar
  - Ablauf: Import von Assembly -> Erzeugen von Templates daraus -> Erzeugen von VMs

Oracle Virtual Assembly Builder

Standardisierte, konfigurierbare “Verbundanwendungen”



- Paketieren komplexer, mehrschichtiger, multi-VM Anwendungen in einzelne, portable “Assemblies”
- Ausrollen von Assemblies in OVM Pool mit autom. Konfiguration und “late-binding” Parametern
- Hinweis: „Restricted use“ Lizenz von OVAB ist in Oracle VM Support Subscription enthalten (ansonsten eigentl. Bestandteil von Oracle Weblogic EE)

Oracle VM Templates

...und wo kommt jetzt die Intelligenz her?

Das Oracle VM Template
…”dumm geboren”?

- Grundsätzlich:
  - Ein Template wird als Klon einer bestehenden VM (bzw. deren virtuellen Platten & Konfiguration) erzeugt
  - Unterschiede zum Original:
    - Neue ID
    - Neue NIC(s) (=>MAC-Adressen)
  - Alles andere ist identisch!
  - Daher => VM aus Template verhält sich auch identisch:
    - Keine neue/andere IP
    - Keine neue Konfiguration von OS und Applikationen

Intelligenz kommt ins Template
Die Oracle VM Guest API

- **API für bidirektionale Kommunikation** zwischen „Gast“ (=VM) und Oracle VM Manager (via deren Server / deren XEN Hypervisor)
- Technische Voraussetzungen:
  - Auf OVM Servern und im OVM Manager:
    Keine (Fähigkeit grundsätzlich vorhanden)
  - In VMs: Installation von sog. „**Oracle VM Guest Additions**“
- Grundsätzliche Funktionsweise:
  - Senden und Empfangen von Messages über Key/Value – Paare
  - Vorgefertigte Mechanik (Skripte) für „First-Boot“-Konfiguration über entspr. Messages enthalten

Oracle VM Guest API

Installiert über “OVM Guest Additions”

- Guest Additions in neueren VMs / Templates von Oracle (a.k.a. „Assemblies“, also OVF-basiert) bereits enthalten
- ...können auch nachträglich über Linux Pakete *ovmd* und *ovm-template-config* installiert werden (z.B. von public-yum)
- Technische Bestandteile:
  - ovmd: kleiner Hintergrundprozess in der VM, macht das eigentliche Message-Handling
  - ovm-template-config: Satz von Skripten für „First-Boot“-Konfiguration
    - Verschiedene „Module“ mit vordefinierten Messages z.B. Für Netzwerk-Konfiguration

Oracle VM Guest API

Setzen und Lesen von Messages

- Oracle VM Manager (GUI)
- *OVM CLI, Bsp.:*

```
OVM> sendVmMessage vm name=ol6u4vm1 key=com.oracle.linux.hostname message=ol6u4.test.com
```

- *ovm_message* – Kommando
  (Bestandteil der separaten „Oracle VM Utilities“), Bsp.:

```
# ./ovm_vmmessage -u admin -p password -h managerhost -v MeineVM -q key1
    Oracle VM VM Message utility 0.5.2.
    Connected.
    VM : 'MeineVM' has status :  Running.
    Querying for key 'key1.
    Query successful.
    Query for Key : 'key1' returned value 'Hugo'.
    Key set 5 minutes ago.
```

- *ovmd* – d.h. Aufruf des Daemons innerhalb einer VM, Bsp.:

```
# ovmd -p MeinKey=IrgendeinWert
```

Oracle VM Guest API

Konkrete Anwendungsfälle

- Jede VM aus „neueren“ Templates (d.h. Mit bereits enthaltenen Guest Additions) meldet dem VM Manager ihre IP-Adresse
- VM Guest API wird (implizit) benutzt für...
  - Konfiguration von aus Templates erzeugten VMs (z.B. bei Cloud / IaaS) in **EM 12c Cloud Control**
    - Achtung: ...NUR bei Assembly-basiertem Workflow (nicht bei „Standard“-Templates)
    - Einfachste Bezugsquelle: „**Self-Update**“ lädt Assemblies in SW-Library
  - Skriptbasiertes (Massen-)Deployment von vorgefertigten Datenbank (RAC od. Single Instance) VMs über *deploycluster.py* (Bezug über MOS oder eDelivery)

Oracle VM 3.x, EM 12c, OVAB

Weitere Informationen:

- **Oracle VM im Oracle Technology Network (OTN):**
  http://www.oracle.com/technetwork/server-storage/vm/overview/index.html
- **Oracle VM Utilities Guide:**
  “2.5 Using Oracle VM Virtual Machine Messaging”
  http://docs.oracle.com/cd/E35328_01/E35333/html/vmutl-using-vmmessage.html
- **Interessante Oracle Blogs zum Thema Virtualisierung**
  - **http://blogs.oracle.com/wim**
  - **http://blogs.oracle.com/virtualization**
  - **http://blogs.oracle.com/olivi_de**
- **Oracle Virtual Assembly Builder (OVAB)**
  http://www.oracle.com/technetwork/middleware/ovab
- **Deutsche DBA Community (für EM12c / Cloud Themen)**
  - **http://blogs.oracle.com/dbacommunity_deutsch**

Join the conversation

@ORCL_Virtualize

facebook.com/OracleVirtualization

youtube.com/OracleVirtualization

ORACLE


ORACLE®