

Deployment patterns for Fusion Middleware

a best practice session by
Simon Haslam & Jacco H. Landlust

`id jacco`

- Oracle Since 2000, Linux since 2001
- Independent Red Stack Administrator
- Oracle ACE since 2006
- Infrastructure: Architecture, High Availability, Troubleshooting & Performance
- <http://oraclemva.wordpress.com>



`id simon`

- Oracle since 1996 (UNIX since 1989)
- Founded Veriton in 1996
- Oracle ACE Director since 2009
- Architecture, Design, Installation
- <http://simonhaslam.co.uk>



Disclaimer

- Best practices of our (Limited) experience
- We are not working for Oracle
- We do not pretend this list is complete
- We are not 'native' American speakers

Agenda

**Best Practices for Designing a Fusion
Middleware | Ig Infrastructure**

Normal Design Process

1. Some product is bought (or build in-house) that runs on FMW 11g
2. Buy some kit
3. a DBA is sent to OU to get training
4. The week after training DBA has to build and maintain a production platform with help from one or more developers
5. Start running production (and into trouble...)

Gartner

- 8 - 12 % TCO in Design + Develop + Test
- 88 - 92 % TCO in Maintaining

Middleware is not Trivial

- It's (mission) critical
- Complicated: lots of moving parts
- Get involved in projects from the start

Domain

Machine A

AdminServer

Machine B

Managed Server

Managed Server

Cluster

Managed Server

Node Manager

Node Manager

Real Agenda

1. Logical Design
2. Physical Implementation
3. Maintaining, the Full Life Cycle

Logical Design

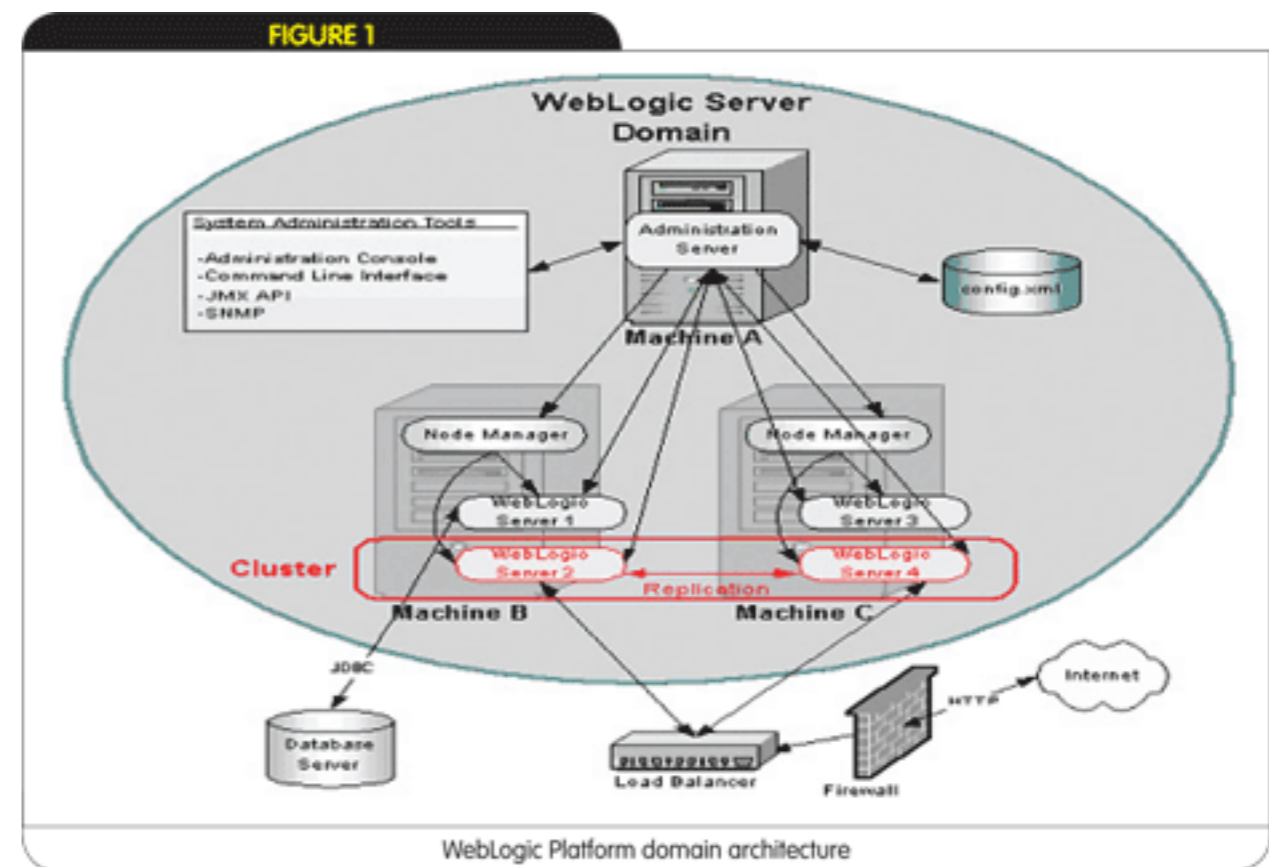
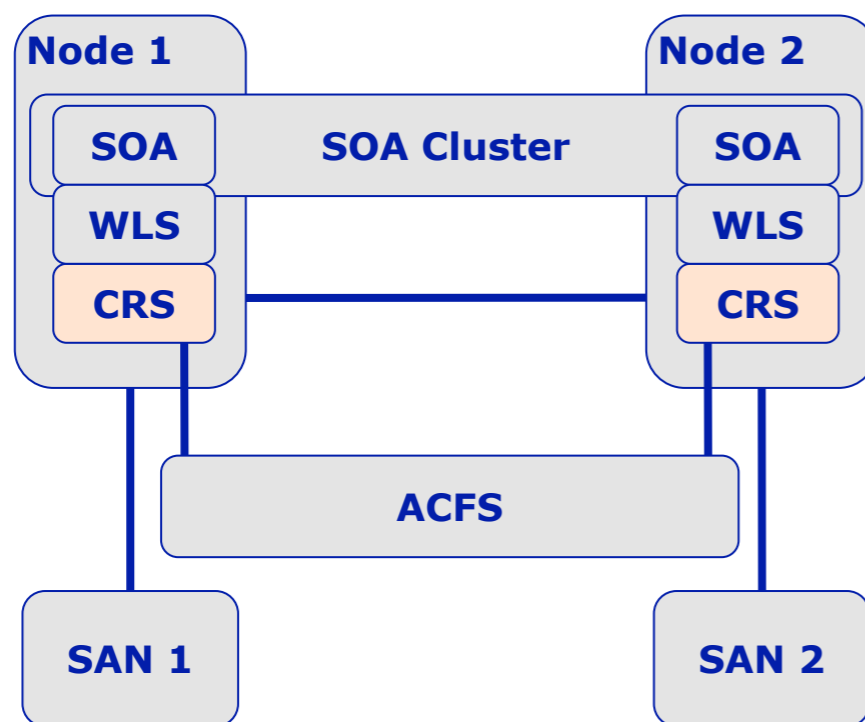
Decide upon Definitions

- Before you design the infrastructure
- Brief your whole team (including management)
- Clear naming convention

Document your Train of Thought

- Tedious but usefull for future use
- Don't wait till after the project / implementation
- Mark facts as facts and opinions as opinion
- Formulas and rules

We Love Pictures



If it's too complicated to draw on a diagram, it's most likely wrong

Segregation between Environments

- DTAP & SOD
- Separation of servers & network

Design for Security

- SOD
- SSL whenever possible
- Also in DEV
- Firewall
- Connection Filters

Don't mix Layered Products in One Domain

- oracle_common scripts
- dependencies on upgrade / patching

Nodemanager: Domain vs Node

- Node specific: settings are domain wide
- Domain specific configuration
- Separation of OS users
- SSL trust-stores

Separate System Components from Java Components

- Automatic network separation
- Different purpose

Never run Apps in AdminServer

- Mixture of management tooling & customer applications
- Difficult troubleshooting

Cluster Everything

- From the start (one node cluster)
- Find unexpected behaviour because of cluster as soon as possible
- Extending clusters is easy

Separate Directories

- Use personal accounts for administrators
- Separate application users from infrastructure users

Physical Implementation

x86

- We've only seen x86/x64 customers running Linux
- Buy the fastest kit you can get

Scale-up vs Scale-out

- Larger JVM give more GC issues
- Our customers do scale-out
- SAME

Virtualization

- Not for vmotion-like functionality
- Cloning (if you put the effort into it)
- Partition hardware

Operating System

- Don't adopt a new OS because of WLS
- The best OS is the one you're an expert at

Shared Storage

- NFS
- OCFS2
- GFS2
- ACFS
- rsync

Shared Storage: take notice of

- HTTP lockfiles
- Large logfiles
- UCM tmpdirs

Use VIP's whenever Possible

- Faster failover upon node failure
- Easier to rebuild on new node
- Logical separation of application and machine

Get a LoadBalancer

- Performance
- Stability
- SSL offloading

oraInventory in MW_HOME

- oraInventory only used by installers & oPatch
- self-contained package
- easier backups

Separate Binaries from Configuration

- Replace binaries more easy
- Local disk for binaries
- Backup configuration

Only use Java-based Nodemanager

- Easier to setup
- Lots of bugs exist for script-based nodemanager
- SSL is a pain on script based nodemanager (if not impossible)

Separate AS from MS

- On different boxes
- Easier upgrade path
- No interference of management tooling
- (if you have the licenses)

Setup AdminChannel (and port)

- Separate admin traffic from application
- Dedicated listen thread
- Has to run over SSL

JDBC

- min-connections = max-connections
- Statement cache size < open_cursors
- FAN (GridLink) when on RAC
See session #20942 13:15 on Wednesday

Centralize Configuration

- No configfiles that need to be changed in JAR/EAR files
- No environment dependent ear-files.
- Easier to maintain

Logging

- Centralize logging
- Logrotation on size + time
- Easier troubleshooting

Standardize

- Throughout all environments
- Easier troubleshooting (at 2 AM)

Automate

- Standardizes installation
- No differences between prod and test

Test Test Test

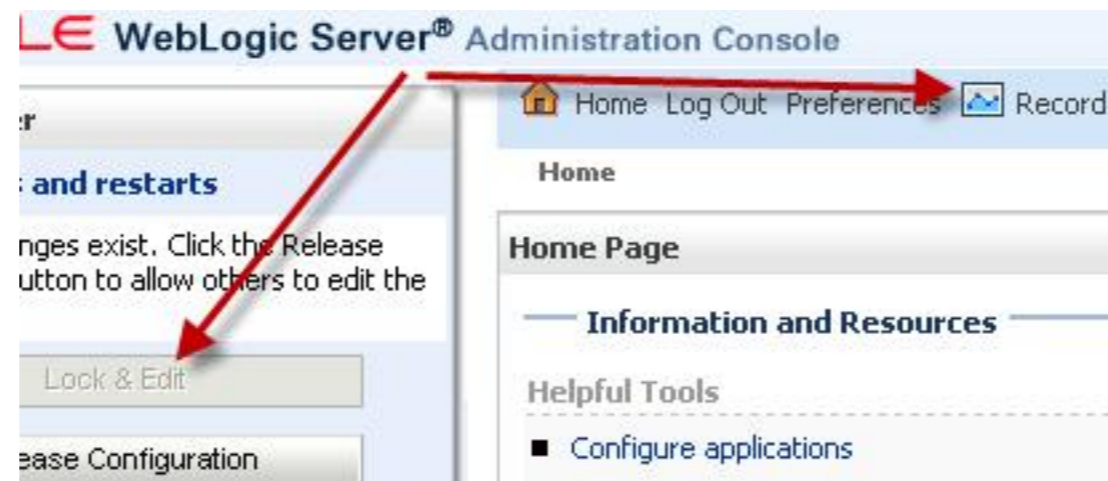
- Find bugs before go-live
- Benchmark
- Stress test until break
- Predictable performance

Maintaining

the full life cycle

Track Changes

- Audit provider
- Record changes with WLST



Record Incidents

- adrci:
 - Build-in fault diagnostibility infrastructure
 - Health monitor reports
 - Package incidents & problem information

learn WLST

- mBeanbrowser
- Automating tasks

Manage deployments

- Learn Ant
- CloningClient.jar
- No environment specific ear-files
- Plan.xml

Out of Place Upgrades

Grid Control

- Stack monitoring
- SLA checks
- Corrective action library

MOS

Configurations

- Not OCM per-se
- Faster SR handling

Summary

first Think, then Act

Standardize & Automate

Document & Test