



Moving Business-Critical Java Applications to Oracle Cloud

Nikitas Xenakis, Co-op

Simon Haslam, eProseed



Agenda

- **Introduction**
- **Technology Drivers, Context**
- **Original Architecture, Challenges**
- **Technical Considerations**
- **Database & Middleware Target Architecture**
- **Implementation Approach**
- **Lessons Learnt and Next Steps**
- **Q&A**

About Me



Nikitas Xenakis

Platform Specialist Architect, The Co-op

- 18+ years as Enterprise DBA (v7-12cR2/19c)
- CAB/Beta Member: Oracle Database, Oracle RAC, Data Integration (GoldenGate)
- Global Leaders' Transaction Processing



@Nikitas_Xenakis



<https://www.linkedin.com/in/nikitasxenakis>



ABOUT ME



Simon Haslam

- Platform / Infrastructure Architect
- Focus includes HA, DR, security, automation



Relevant to this session

- WebLogic / FMW installations since 2000s
- First research/webcast on JCS in 2016
- Designed & built SOA CS integration platform for global use since Oct 2017
- On team migrating eProseed Lux data centres to OCI

Doing the right thing since 1844

Because we are not just
about profit



The Co-op in 2019



Leading UK Convenience Retailer

- ✓ Annual Revenue: \$14.5B
- ✓ 2800+ Owned Stores
- ✓ Retail, Wholesale, Franchise, Ecommerce
- ✓ 14 Distribution Centres
- ✓ Logistics Network servicing 10,000+ Outlets



Co-op HQ, Manchester UK - One of the most sustainable large buildings in the world



Modern Business Demands

- ✓ **New Markets/Channels**: Wholesale, Franchising, Ecommerce
- ✓ **Fit for the future**, continuously improve
- ✓ **Fuel for Growth**, efficiencies re-invested to the business
- ✓ **Technology** as **enabler** and **transformer** for business growth

Technology Drivers

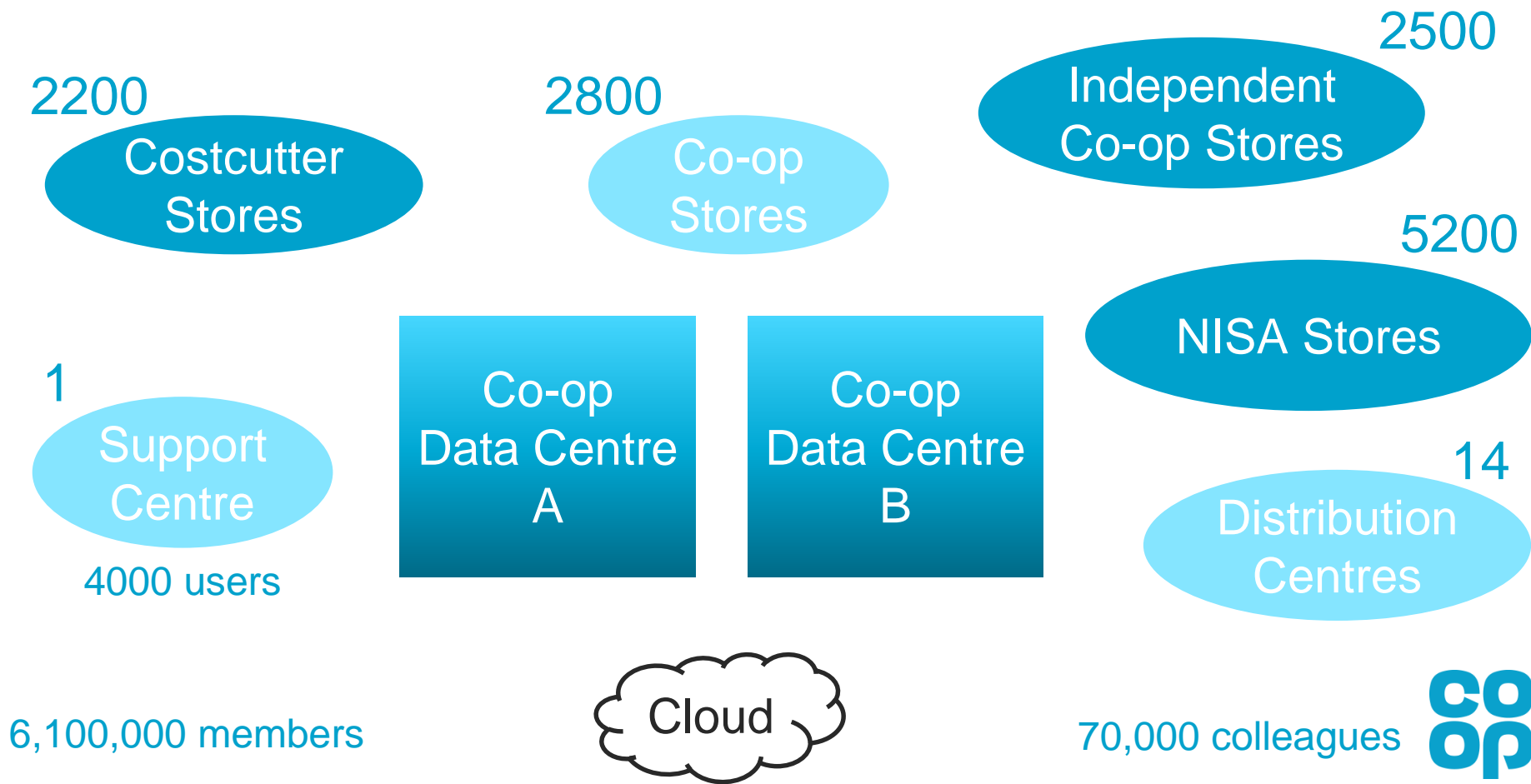
- ✓ **Simplification & Standardisation** of Database & Middleware platforms
- ✓ **Zero lost productive hours** from preventable issues, **protect service**
- ✓ Continuous Delivery, Continuous Integration (**CI/CD**)
- ✓ Increase **availability**, **scalability**, **agility**, **security** – downtime, poor performance is extremely costly
- ✓ **Data Centre Exit** Strategy

The Challenges right now...

Develop and maintain a HA/MAA On-Premises and Cloud Landscape while at the same time ...

- ✓ **Cloud First** for new Solutions/Deployments
- ✓ Digital Transformation, **Agile** & **DevOps**
- ✓ **Data Centre** Minimizing footprint and **Exit**

Business & IT Context



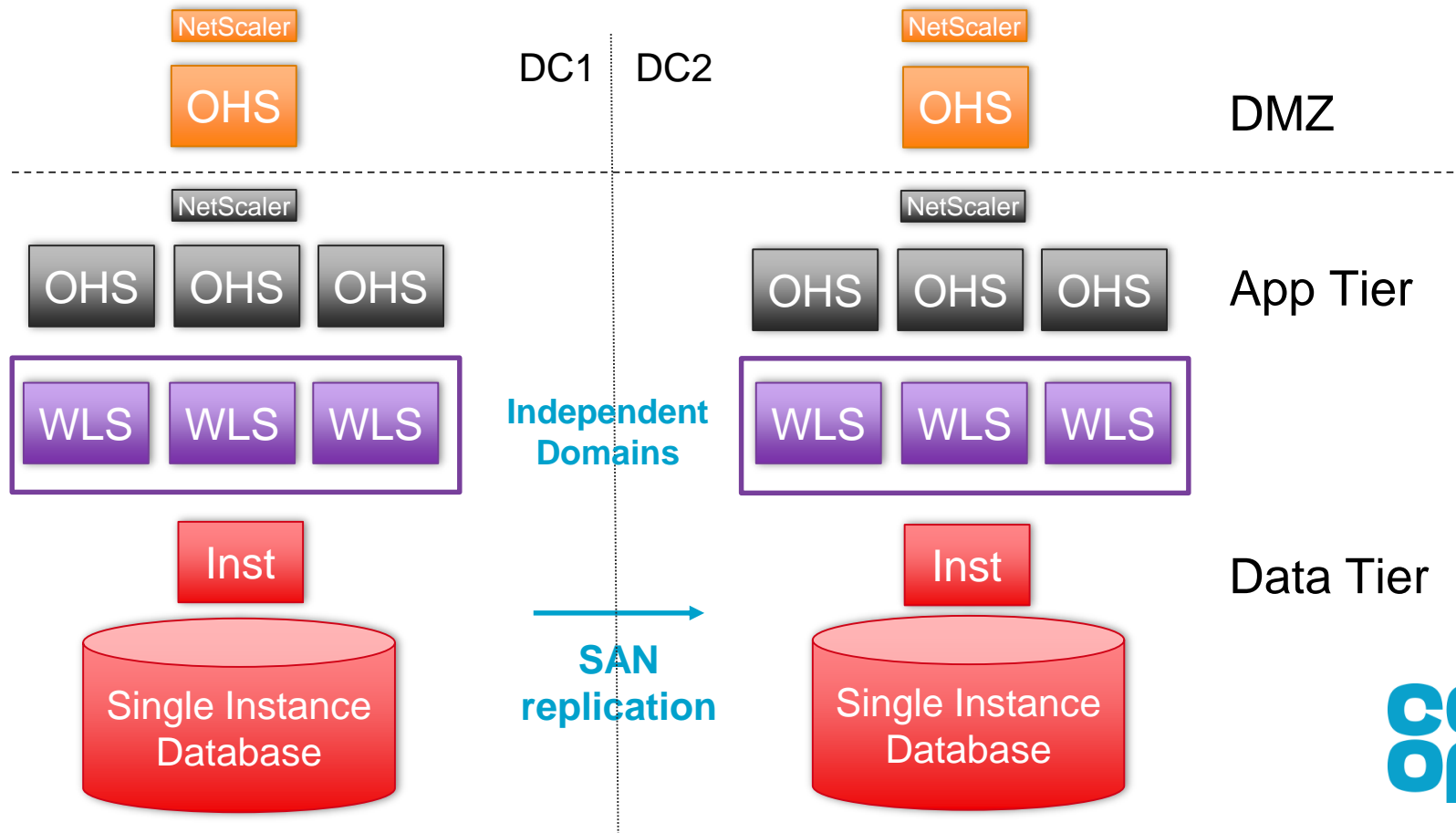
Our Use Case



Our Use Case

- **Business-critical** Ordering App used in stores
- **Java** based Application, **externally facing** (DMZ) and currently running on-premises in Co-op's Data Centres
- **Limited HA** and **DR** to cover loss of one data centre
- Oracle software versions:
 - **Fusion Middleware - Oracle WebLogic & Oracle Webtier (HTTP Server) 12.1.3**
 - **Oracle Database 11.2.0.4**

As-Is Architecture

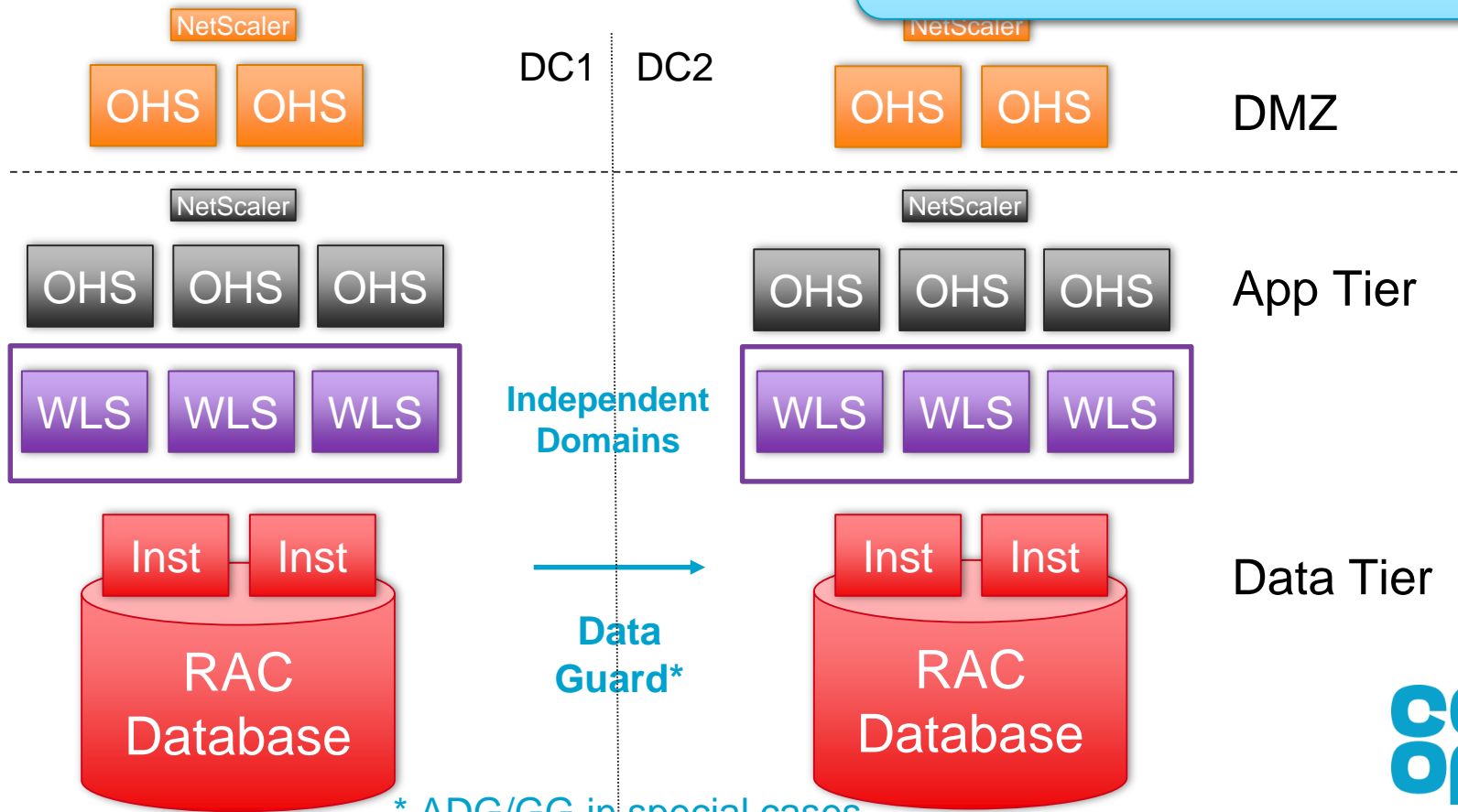


Challenges & Drivers

- **Data Centre Exit** strategy – WOS is medium term lifespan app so needs to be migrated to cloud as part of DC exit.
- Current versions (WLS 12.1.3) of Oracle products are soon **out of support** (Premier ended Dec 2017, **Extended** to **Dec 2019**) so will impact Oracle support cost & potentially ability to maintain service levels.
- **Services include OHS in DMZ**, open to internet, so needs support & patches.
- Mostly reliable but have issues when OS patching DMZ in particular: limited testing capability for DMZ/OHS so every **Windows patch application runs risk** of outage.
- **Limited HA in DMZ web tier layer** and **database layer** so single component failure **risks non-trivial outage**.

On-Premises MAA R

Everything fully redundant
plus independent DR



* ADG/GG in special cases

Java App Architecture Options on Oracle Cloud



PROVISIONING OPTIONS FOR WEBLOGIC TODAY

Traditional Install

Traditional on-prem or IaaS – full install with OUI, WLST etc

Java Cloud Service

First implementation, limited choices, legacy

Marketplace Image

new

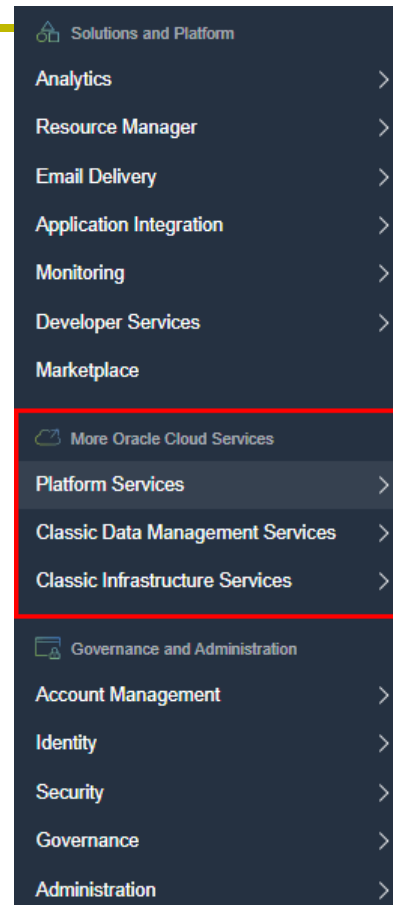
OCI method, combination of images & configuration

Docker/Kubernetes

Modern, layered approach

JAVA CLOUD SERVICE ONTO OCI

- Still available but in our opinion seems legacy
 - Uses Platform Services Manager
 - Console changes & PM hints suggest retirement



JAVA CLOUD SERVICE ONTO OCI

- Restricted configuration
 - Domain name derived from Instance Name
 - Choose the AD (1 only), cluster can't span, if use OTD then same VCN/subnet
 - Needs a DBCS instance (can be ATP)
- Only provisions into one compartment (ManagedCompartment) out on admin seg

Now best avoided

Create Oracle Java Cloud Service Instance

Provide basic instance information

* Instance Name

Description

Notification

License Type ☒ My organization already owns Oracle middleware software licenses. Bring my existing middleware software license to the Oracle Java Cloud Service. ☐ Subscribe to a new Oracle Java Cloud Service software license and the Oracle Java Cloud Service.

* Service Level

* Software Release

* Software Edition

Database Configuration

* Database Instance Name

Backup and Recovery Configuration

* Backup Destination

WebLogic Clusters

Compute Shape

Server Count

* Domain Partitions

Enable Access to Administration Consoles ☐

Deploy Sample Application ☒

WebLogic Access

Enable Authentication Using Identity Cloud Service ☐

* SSH Public Key

* Local Administrative User Name



WEBLOGIC ON OCI MARKETPLACE

- Non-legacy products arriving on OCI Marketplace
 - Template / stack model
- Creates domain with some customisation:
 - Domain name
 - Compute prefixes
 - # nodes
 - NM/MS ports/admin channels
 - Sample app
- OCI-specific
 - AD (single one)
 - Compartment
 - VCN
 - Needs a key vault (virtual one becoming available)

The screenshot shows the 'Create Stack' wizard in the Oracle Cloud console. The wizard has three steps: 1. Stack Information, 2. Configure Variables (current step), and 3. Review. The 'Configure Variables' step is titled 'WebLogic Server Instance' and contains several configuration fields with descriptions and validation messages.

ORACLE Cloud

Create Stack

Configure the variables for the infrastructure resources that this stack will create when you run the apply job for this execution plan.

1 Stack Information
2 **Configure Variables**
3 Review

WebLogic Server Instance

RESOURCE NAME PREFIX

The names of all compute and network resources will begin with this prefix.
This variable is required.

WEBLOGIC SERVER SHAPE
Select an option
The shape for all WebLogic Server compute instances.
This variable is required.

SSH PUBLIC KEY

Use the corresponding private key to access the WebLogic Server compute instances.
This variable is required.

WEBLOGIC SERVER AVAILABILITY DOMAIN
Select an option
The name of the availability domain in which to create the WebLogic Server compute instances.
This variable is required.

WEBLOGIC SERVER NODE COUNT
1
The initial number of WebLogic Server compute instances, and also the number of managed servers in the cluster.

WEBLOGIC SERVER ADMIN USER NAME

The name of the administrator in the WebLogic Server domain.
This variable is required.

WEBLOGIC SERVER ADMIN PASSWORD

The encrypted password for the administrator in the WebLogic Server domain. Use a WebLogic Administrator password that starts with a letter, is between 8 and 30 characters long, contains at least one number, and, optionally, any number of the special characters (\$ # _). For example, Ach1z0Rd.
This variable is required.

Back Next Cancel

AN ASIDE: DOCKER & KUBERNETES PRIMER

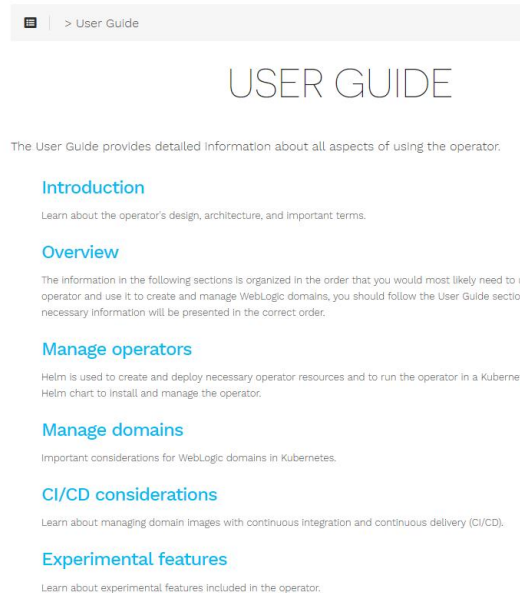
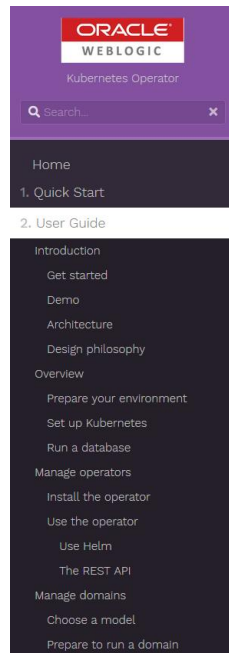
DOCKER & KUBERNETES

- Docker is the de-facto container solution allowing layered images of software
 - Developer-driven
 - A “scalpel-sharp, generic installer”
 - Images are applied to a single host, e.g. your PC
- Docker registries are like app stores (Docker Hub, OCI Registry)
- Kubernetes is a very popular Docker image orchestration engine
 - Allows you to define services based on images, then does all the lifecycle management you need for each service



MORE KUBERNETES CONCEPTS

- Kubernetes (K8s) doesn't know about deployment technology
 - operators, e.g. weblogic-operator controls lifecycle for WLS cluster (WebLogic & FMW Infra already, others in progress)
- Registry manages the images
- K8s **manages** nodes, pods & secrets
- Typically you **centralise logging**
- **Scaling** out of the box
- Load balancing / ingress control
- As K8s manages cluster it's doing some functionality of WLS Admin Server



MANAGING KUBERNETES CLUSTERS

- You can build your own Kubernetes clusters, e.g. on-prem or on IaaS
- Someone else can set up & run the K8s cluster for you: managed Kubernetes
 - Already several mature vendors/products
 - Pick your deployment target (yes, we are still at Oracle OpenWorld!)
 - Licensing rules still apply
- Oracle Container Engine for Kubernetes (OKE)
 - Licensing rules still apply
- Added value such as Wercker, now called Pipelines, for CI/CD

COMPARISON OF TRADITIONAL WLS PROVISIONING

Traditional

- ✓ We (all WLS admins) know how to do it
- ✓ Re-use existing investment in WLST etc
- ✗ Takes a long time
- ✗ Automation takes a lot of work unless you use something like Puppet or Ansible
- ✗ Deployment is a separate exercise
- ✗ Encourages pets



Java Cloud Service

- ✓ Very natural for WLS admins
- ✓ Up and running very quickly
- ✓ Includes a web tier
- ✗ Deployment is a separate exercise
- ✗ Very restricted topologies
- ✗ Rules as to what you can do
- ✗ Have full JFR/OPSS/repo whether you want it or not
- ✗ Patching is DIY, even JDK

NOW best avoided

COMPARISON OF CLOUD PROVISIONING

WebLogic OCI Marketplace

- ✓ Up and running very quickly
- ✓ Strategic
- ✓ Some topology options (no DB needed)
- ✓ Comfortable for all WLS admins
- ✗ Patching is DIY, even JDK
- ✗ App deployment is separate
- ✗ Very new

WebLogic on Kubernetes

- ✓ Modern, layered approach, fits into the big picture
- ✓ Strategic
- ✓ Multi-cloud & on-prem portability
- ✓ Encourages cattle
- ✓ Includes app deployment for DevOps / full CI-CD
- ✗ New tools, skillsets
- ✗ Learning curve for trad. WLS admins
- ✗ Fairly new

MORE KUBERNETES & ORACLE CLOUD...

LOAD BALANCING OPTIONS

Supported by the WebLogic operator:

- Traefik
 - Voyager
 - Apache
- ⇒ All have knowledge of cluster & internal k8s networking (dynamic IPs)
- ⇒ No dominant choice yet, especially when get to FMW
- OCI LBaaS (nginx)

Notes

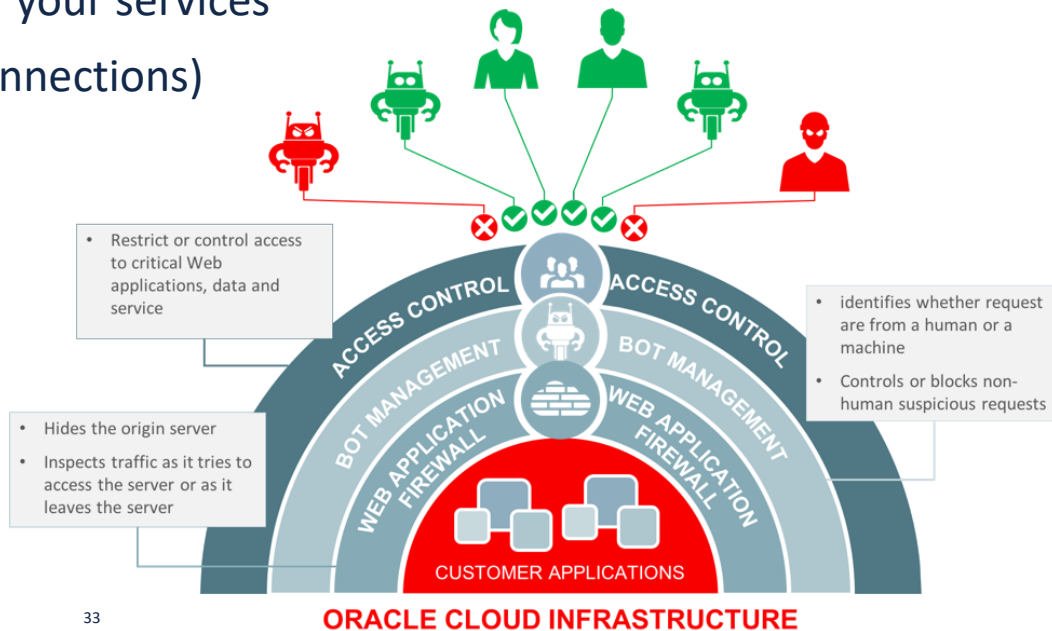
None of these LB have WLS' "pro-active" failure detection in the WebLogic plugin/OHS/OTD (maybe Apache could get it?). That health detection functionality will *approximately* move into the WebLogic operator over time?

If you're running REST services in WLS maybe new OCI API Gateway could be useful?

If you're using WAF then that is the true front-end for users.

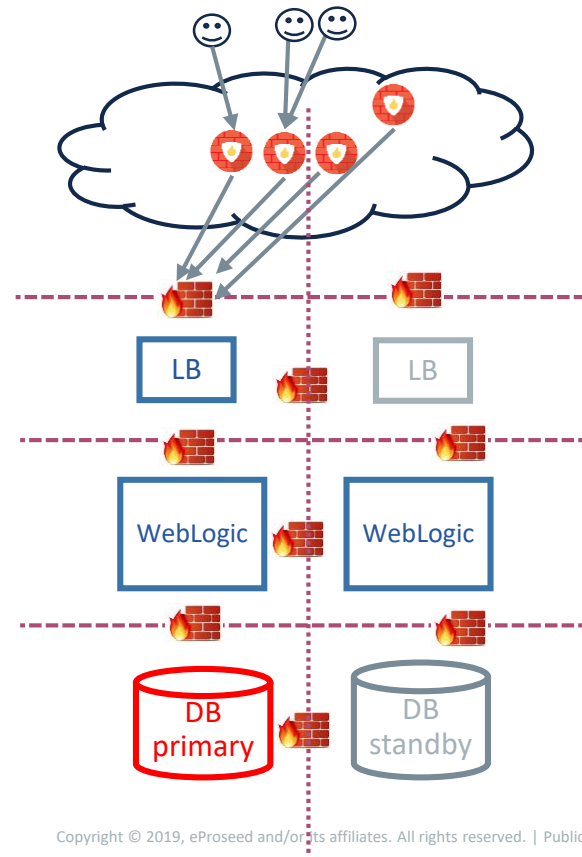
WEB APPLICATION FIREWALL

- Oracle now provides WAF in OCI
- DDoS & optional L7 DDoS protection
- Global proxy layer that sits in front of your services
- Very inexpensive (billed by data & connections)



WEB APPLICATION FIREWALL

- WAF is a proxy & traffic filter but not a load balancer
 - WAF has lots of policies it can apply from sophisticated ones (like botnet protection) to simple (IP white-listing)
- All public traffic only goes through WAF, with OCI systems then IP white-listed to WAF addresses
- Multiple proxies, around the world
 - It appears to be routed via proxy in your “nearest” OCI
- Your public DNS points to a WAF CNAME, which then resolves to WAF proxies
- Multiple ways to script deployment
 - Terraform provider supports main WAF features (policy creation, TLS certs, etc)



Database Architecture Options



Maximum Availability Architecture (MAA) in Oracle Cloud Infrastructure (OCI)

HA/MAA Architecture Considerations

- **RPO** – How much data can you afford to loose ?
- **RTO** – Downtime? How quickly should you be back up ?
- **Performance** – Performance after the event ?
- **Perceived Application Outage** – What does this mean to the end-user and operation (**Transparent ? or Not ?**)

MAA in Oracle Cloud Infrastructure (OCI)

Why ?

- **Best In-Class** for Mission-Critical (Platinum/Gold) workloads
- Oracle Database Enterprise Edition and all Options (Inc. RAC)
Certified
- Not just IaaS full PaaS, DBCS available
- Virtual, Bare Metal and Exadata (dedicated) offerings
- Multitenant inherently at no additional cost



MAA in Oracle Cloud Infrastructure (OCI)

Why ?

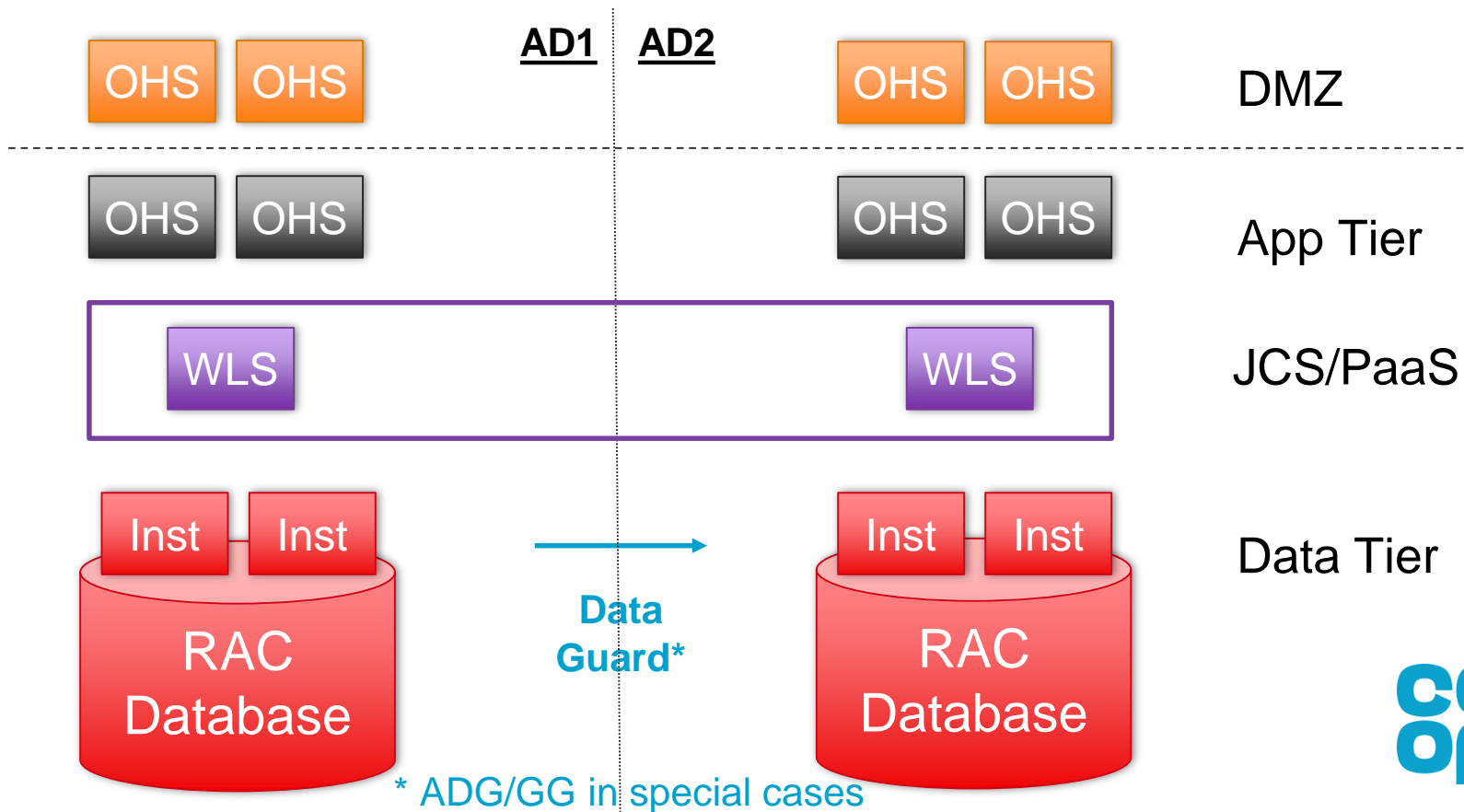
- BYOL and various commercial models (Universal Credits) available
- Cost-effective on licensing (1 Core = 0.5 CPU, matches On-Prem)
- Closest match to On-Prem MAA platinum patterns
- Several **Differentiators** and **Innovations** on **OCI** incl. **PaaS**, **DBCS**, **Exadata (EXACS)**, **ATP**, **ADW**

Why Not ?

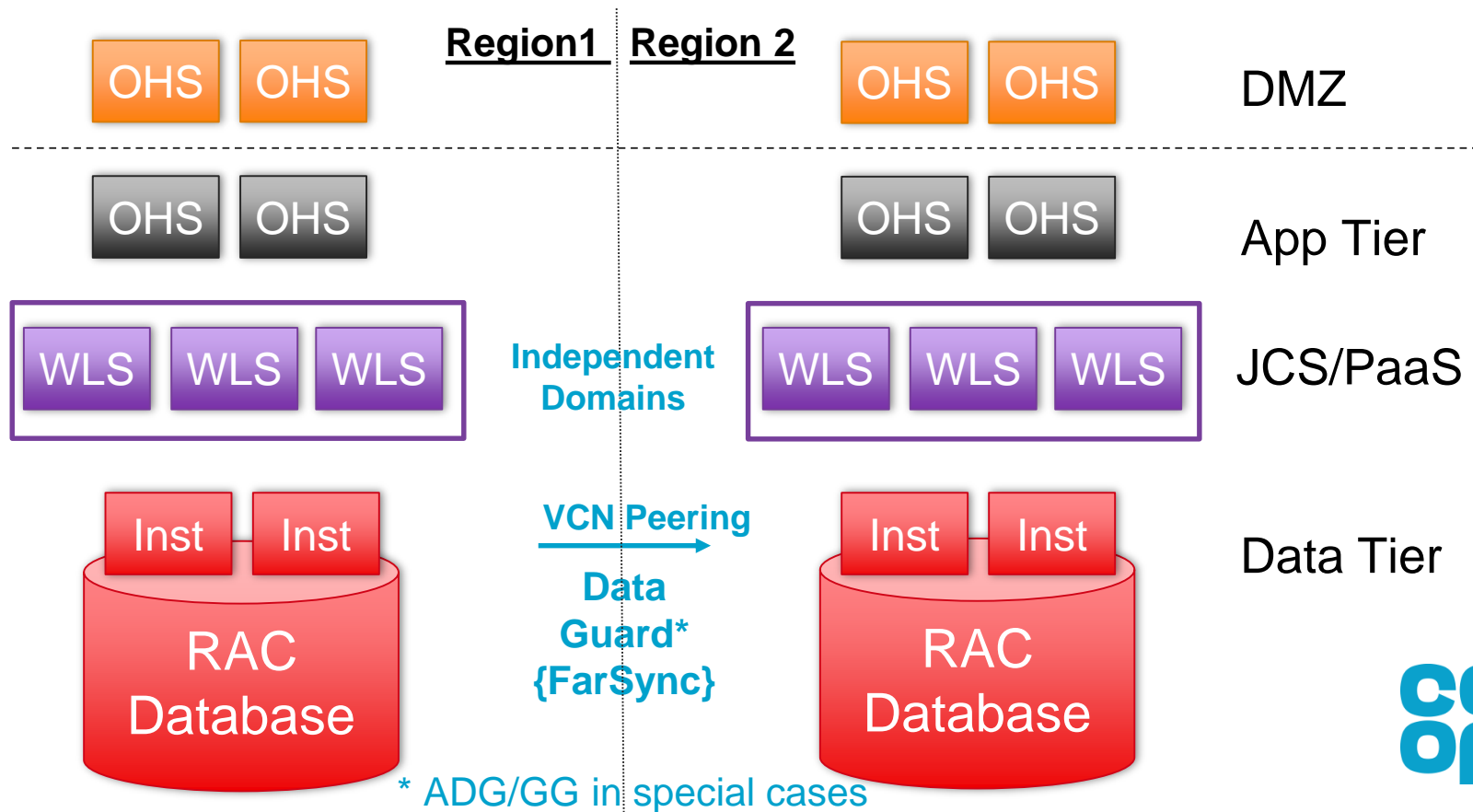
- Oracle catching-up on Cloud but very quick



MAA in OCI – HA General Approach



MAA in OCI – HA/DR General Approach



MAA in OCI Design Patterns

	Database		Middleware
1	Single instance (1PDB in CDB)	Data Guard {&FSFO}	WLS 1 or 3 nodes 1 domains/site
2 Preferred	RAC (1 PDB in CDB)	Data Guard	WLS/JCS 3 node & Grid Link
3	RAC (1 PDB in CDB)	Active Data Guard	WLS 3 node {AC/TAC}
4 Preferred	RAC (1 PDB in CDB)	Active Data Guard {& GoldenGate}	(non-Oracle)/(T)AC

MAA in OCI Design Patterns


 New

	Database		Middleware
1	Single instance (1PDB in CDB)	Data Guard {&FSFO}	WLS 1 or 3 nodes 1 domains/site
2 Preferred	RAC (1 PDB in CDB)	Data Guard	WLS/OKE 3 node & Grid Link
3	RAC (1 PDB in CDB)	Active Data Guard	WLS 3 node {AC/TAC}
4 Preferred	RAC (1 PDB in CDB)	Active Data Guard {& GoldenGate}	(non-Oracle)/(T)AC



Autonomous Database

Why Autonomous Database ?

Autonomous benefits to Co-op – **Considerations**

- **Quickest path to Exadata**, quick provisioning (5 mins)
- Regular Database Patching applied
- Security patches automatically applied
- **Compliance** with **Security**, **Risk**, **Audit**
- Fit for purpose for DevOps and CI/CD pipeline
 - **High degree of automation, lower TCO**

Why Autonomous Database ?

Autonomous benefits to Co-op – **Challenges!**

- **Serverless** option is shared but **Dedicated** is available for Max resource protection
- Currently **ATP** available in 1 Availability Domain (AD)
- (Active) Data Guard not available currently
- DR capability cross-region is currently limited

Why Autonomous Database ?

Specifics to our Use Case (Inbound/Outbound files)

- Data Integration requires thought
- Sqldr not available currently for uploading data
- Can use data pump, external tables, database links, SQL*Developer
- Can also use ODI, Goldengate



Why Autonomous Database ?

Specifics to our Use Case (Inbound/Outbound files)

- Instead of reading and writing files to the OS with ATP, you read and write files to the object store.
- Create your own private bucket on the object store and then using your object store credentials you can read and write files to that bucket from ATP via the DBMS_CLOUD package instead of UTL_FILE.

Data load examples : <https://docs.oracle.com/en/cloud/paas/atp-cloud/atpug/load-data-cloud.html#GUID-07900054-CB65-490A-AF3C-39EF45505802>



DBCS or Autonomous ?

DBCS or Autonomous for our Use Case ?

Specifics to our Use Case – **Batch processing**

- **DBCS** for our use case is low risk and **requires less change**
- **ATP** has a **cost of change** associated around batch processing deployment, configuration

Implementation Approach

Next Generation of (Cloud) Platforms

Modern Platforms should be architected and engineered based on Infrastructure & Platform as Code (IaC, PaC) with the following in mind:

- ✓ Simple
- ✓ Repeatable
- ✓ Scalable, Elastic
- ✓ Secure
- ✓ Versionable

KEY CONSIDERATIONS FOR THIS USE CASE

- Stateless
 - No JMS
 - No local files
 - Session state is for authentication/authorisation (internal mechanism)
- Plain Java webapps
 - No ADF / JRF / OPSS etc
 - JSP/JSF/JDBC
- Traditional database architecture
 - Files in and out of the system go via tables, PL/SQL, SQL*Loader
 - Flat file data to/from other systems

JCS ruled out as legacy

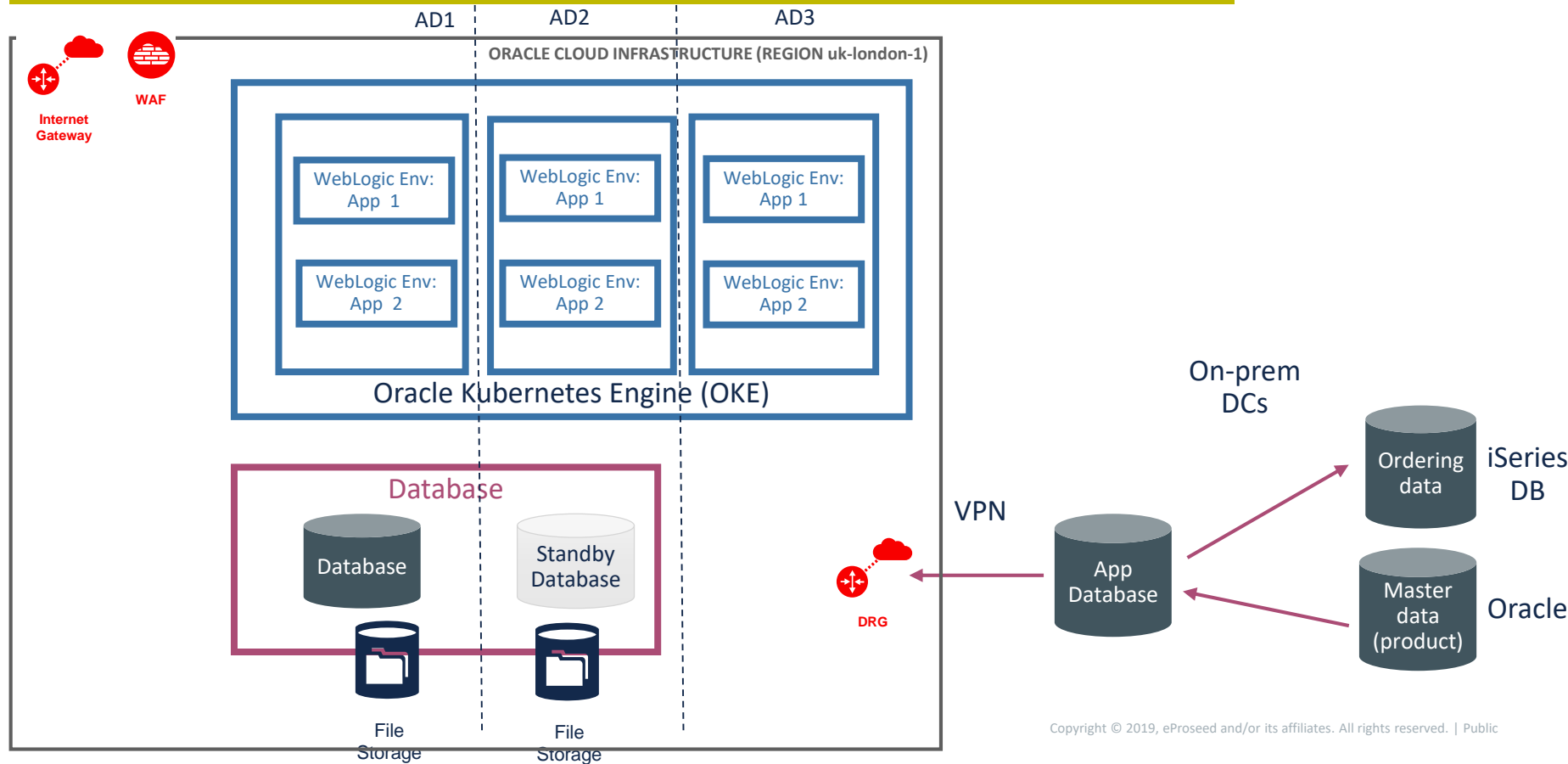
Marketplace is an option but we have more apps to migrate so want to use same compute resources

Containers are strategic deployment for WLS & FMW for Oracle so we wanted to align with that

OCI Database options considered for database, with OCI File and Object Storage considered for files

Lift and shift => don't yet want to change (external) developer workflow or make code changes to database... one step at a time 😊

ARCHITECTURE



PHASED APPROACH

- Phase 1
 - WLS 12.1 to 12.2 upgrade
 - DB 11.2 to 19c upgrade
 - Persistent volume for domain & logging
 - Traditional app deployment from dev team
 - Production proving
- Phase 2
 - Exporter for centralised logging & metrics
 - Pipeline for image production, including smoke test
 - Bake app into image



Upgrade

Refine

INFRASTRUCTURE AS CODE

- Terraform for OCI resources:
 - Compartment
 - Network (VNCs, security lists, IG, DRG)
 - DBCS
 - WAF (not everything possible in TF yet)
 - OKE cluster
- Kubernetes
 - Admin client setup (OCI CLI, weblogic-operator)
 - Covered in plans
- Database configuration
 - SQL & datapump initially. PDB later.

ORACLE PROVIDED TOOLS

- WebLogic Operator
- (optional, as you get more mature)
 - WDT – introspection of existing domains
 - WebLogic Image Patching tool

SUMMARY

SUMMARY

- Docker/Kubernetes offers modern developer-friendly tooling (on-prem or cloud)
 - Benefits to Ops even if not fully adopted by development
 - There is quite a lot to learn, though OKE makes the K8s cluster set up almost trivial
 - WebLogic in OCI Marketplace might be a good option for shorter-term apps
- OCI provides many services that can be easily used, e.g. LBaaS & WAF
- Oracle database has different approach to segregation/containerisation
 - Autonomous database is the future but DBCS is good stepping stone
 - Follow full MAA best practices for OCI
- BYOL makes OCI cost effective for migrations

Co-op HQ

One of the most sustainable large buildings in the world

2013

Thank You!

Q&A



https://en.wikipedia.org/wiki/One_Angel_Square

