

# #PTK

PASS THE KNOWLEDGE

— INDEPENDENT —  
**UK OUG**  
UK ORACLE USER GROUP

[WWW.UKOUG.ORG](http://WWW.UKOUG.ORG)

WINTER 2019/20 | ISSUE 72

**FORMERLY ORACLE SCENE**

AN INDEPENDENT  
PUBLICATION NOT  
AFFILIATED WITH  
THE ORACLE  
CORPORATION

**TECH  
EDITION**  
→  
TURN OVER FOR  
BUSINESS  
APPS

+

## OPENWORLD 2019

5 important tech take-aways  
from San Francisco

+

## MASTERING THE CLOUD

From on-premise  
to Oracle Fusion:  
the lessons learned  
at a UK university



+

## PREPARING A PLATFORM ON DEMAND

How to boost  
Oracle FMW  
with Ansible

# Free



# and easy!

Starting out with Oracle's Autonomous Database –  
and how to get the most from it without paying a penny

# BUSINESS APPLICATIONS *Xchange*

15 - 16 June 2020, Oval Cricket Ground

**E-BUSINESS SUITE**

**PEOPLESOFT**

**JD EDWARDS**

**HYPERION**

**APPS TECH**

Forget Apps19,  
we've reinvented the  
Applications Conference.  
Say hello to BAX.

Join the Business Apps  
community next June for the  
latest updates and  
insights from leading Oracle  
experts. Find out more at  
[ukoug.org/bax](http://ukoug.org/bax)

**CLOUD ERP**

**CLOUD EPM**

**CLOUD HCM**

**CLOUD INNOVATION**

## #PTK Tech Edition

### IN THIS ISSUE

**11**

#### Highlights from OpenWorld 2019

In case you couldn't make it  
to Oracle's flagship event...

**14**

**COVER STORY**

#### First steps with Autonomous

How to get started with Oracle's  
self-driving cloud database – plus  
making the most of 'Always Free'

**18**

#### Mastering the cloud

...at Staffordshire University:  
Part 2 of their journey to Fusion

**22**

#### Delivering a platform on demand

How to boost Oracle Fusion  
Middleware with Ansible

**26**

#### 'But that worked yesterday!'

The finer points of self-testing  
using utPLSQL



**14**



**FLIP  
OVER**

**FOR BUSINESS  
APPS!**

### PLUS

**05**

#### Me & the UKOUG

Meet board Member  
Advocate Neil Chandler

**06**

#### News Etc & Events

RIP Mark Hurd, Cloud  
Free Tier, plus our 2020  
gatherings for your diary

**08**

#### Speaker of the Year

Who were the top presenters  
at UKOUG events in 2019?

Flip over to the Business Apps  
side of this issue to find out  
how the new breed of IT  
leaders are using technology to  
drive maximum value in their  
organisations (page 16) – plus  
even more on what matters  
in the world of business  
and technology.

### #PTKOnline

View this latest issue  
online and access the  
archive of Oracle Scene  
editions here:  
[www.ukoug.org/ptk](http://www.ukoug.org/ptk)

# Welcome to the new issue of #PTK Pass The Knowledge



One of the most striking announcements at Oracle OpenWorld this year was the launch of its Always Free Cloud Tier. Larry Ellison himself has promised that this will remain completely gratis, forever, which is a genuinely bold move that may well help Oracle to grow its foothold in the cloud market and be of real value to a wide range of users.

However, perhaps more significant, was the buzz around Autonomous

Database itself, with its promise of taking a lot of the grunt work out of provisioning and running databases in the cloud, thus freeing up time and resources for more rewarding tasks.

This is yet another way in which advances in digital technology – from Oracle and elsewhere – are allowing tech experts to increase their value to the organisations in which they work: instead of spending time doing boring, repetitive tasks, Autonomous should allow them to focus on activities that help their businesses to grow and flourish – such as identifying and provisioning innovative new technology, establishing better data management and analysis or creating improved processes.

Turn to our cover story (**'First steps with Oracle Database Autonomous Transaction Processing', page 14**) to find out more about Autonomous, what it can do – and advice on how to get started in nine easy steps.

Elsewhere in this issue, we have a round-up of some key tech **highlights from OpenWorld 2019 (page 11)**, plus a useful look at lessons learned from a migration to **Oracle Cloud at a leading UK university (page 18)**.

I hope all these articles will, as ever, help you to maximise the value of the Oracle technology you deploy.

I would also encourage you to flip over and take a look at the Business Apps side of the magazine – especially if increasing the value of IT to your organisation is high on your agenda. Our feature on the role of the modern IT leader (**'The right stuff', page 16**) should help you to do just that.

Finally, I'd like to take this opportunity to say how proud I am to be taking over as Editor of #PTK, having worked closely on this magazine's relaunch during 2019. I'd like to thank my predecessor, Andy Nellis, for his smart thinking and clear vision during his time in the role. And even greater thanks goes to every UKOUG member who has contributed to the magazine or shared feedback so far – it's all valuable and will help us to make the 2020 editions even better for the whole community. If you have any comments, ideas or criticisms, please drop me a line at [editor@ukoug.org](mailto:editor@ukoug.org).

With best wishes for a successful and prosperous new year,

**James Lawrence**  
Editor

#### About the Editor

James Lawrence is a professional multi-media journalist and editor who has been covering business and technology for more than a decade.

#### #PTK: PASS THE KNOWLEDGE EDITORIAL TEAM

##### Editor

James Lawrence  
[james.lawrence@topdogcomms.co.uk](mailto:james.lawrence@topdogcomms.co.uk)

##### Art Director

Peter Allen  
[peter@peterallendesign.co.uk](mailto:peter@peterallendesign.co.uk)

##### Consulting Editor

Martin Widlake

##### UKOUG contact

Tania Huntington  
[tania.huntington@ukoug.org](mailto:tania.huntington@ukoug.org)

##### Advertising

Kerry Stuart  
[kerry.stuart@ukoug.org](mailto:kerry.stuart@ukoug.org)

##### Membership

Jordan Osborne  
[jordan.osborne@ukoug.org](mailto:jordan.osborne@ukoug.org)

##### UKOUG governance

A full listing of Board members, along with details of how the user group is governed, can be found at: [ukoug.org/meettheteam](http://ukoug.org/meettheteam)

##### UKOUG office

19-23 High St,  
Kingston upon Thames,  
London KT1 1LL  
Tel: +44 (0)20 8545 9670  
[info@ukoug.org](mailto:info@ukoug.org)  
[www.ukoug.org](http://www.ukoug.org)

##### Produced by

##### Top Dog Communications Ltd

Tel: +44 (0)7913 045917  
[james.lawrence@topdogcomms.co.uk](mailto:james.lawrence@topdogcomms.co.uk)  
[www.topdogcomms.co.uk](http://www.topdogcomms.co.uk)

##### Designed by Peter Allen Design

[peter@peterallendesign.co.uk](mailto:peter@peterallendesign.co.uk)  
[www.peterallendesign.co.uk](http://www.peterallendesign.co.uk)

#PTK: Pass The Knowledge  
© UK Oracle User Group Ltd  
The views stated in #PTK are the views of the author and not those of the UK Oracle User Group Ltd. We do not make any warranty for the accuracy of any published information and the UK Oracle User Group will assume no responsibility or liability regarding the use of such information. All articles are published on the understanding that copyright remains with the individual authors. The UK Oracle User Group reserves the right, however, to reproduce an article, in whole or in part, in any other user group publication. The reproduction of this publication by any third party, in whole or in part, is strictly prohibited without the express written consent of the UK Oracle User Group.  
Oracle is a registered trademark of Oracle Corporation and/or its affiliates, used under licence. This publication is an independent publication, not affiliated or otherwise associated with Oracle Corporation. The opinions, statements, positions and views stated herein are those of the author(s) or publisher and are not intended to be the opinions, statements, positions, or views of Oracle Corporation.



**Neil Chandler** is a Member Advocate on the UKOUG board. Here, he explains how he's helping the group to adapt and move forward



**Right from the first event I attended, 25 years ago,** I could see the advantage of the UKOUG community – and especially the networking aspect. When you're interacting with your peers and learning from them, you're listening to people talk about problems that you haven't faced yet but might face in the future. So you get to understand where they went wrong and how avoid it, and it lets you implement best practices based upon their experiences. The other thing is that you get to interact not just with your peers but also with Oracle product managers and the executives. So you learn what's new, what's coming down the line, what's going to help you improve your businesses and increase your return on investment for the products.

**Being a member gives you a voice.** Any individual company, unless they're huge, tends to be ignored, but when you've got a user group the way we have, the collective voice tends to get listened to. And Oracle does listen to both the positives and the negatives that we transmit to them from the membership. It's a really great interaction that you might not otherwise get, especially if you're from one of the relatively smaller companies.

**Being involved broadens your horizons... and it's important to have fun at the same time**

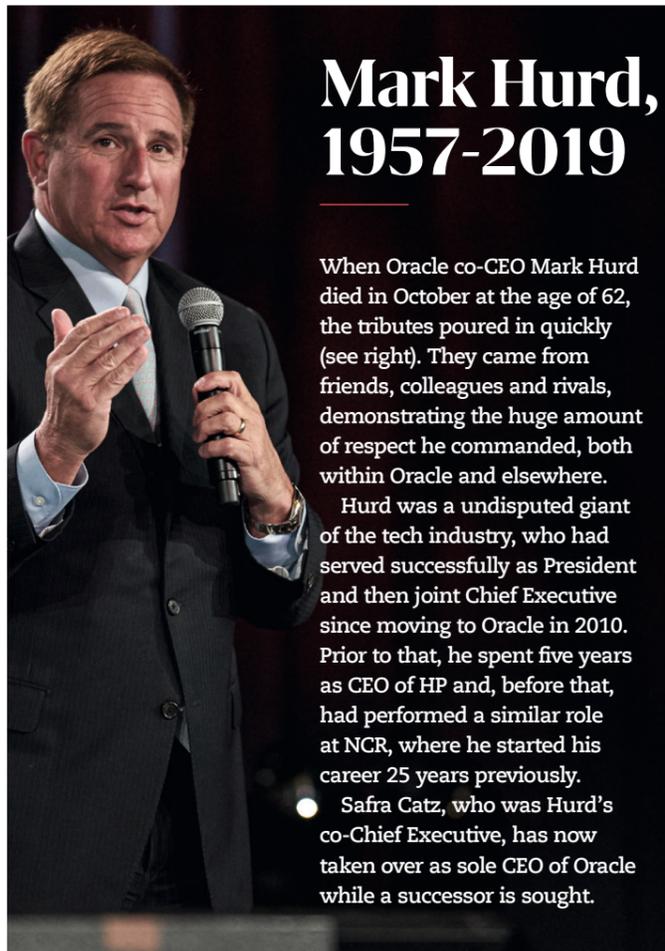
**Joining the board seemed like a natural progression for me.** I've gone from attendee to volunteer – to special interest group (SIG) chair, to conference speaker, then to conference lead and now to Member Advocate on the board. I decided to do it because I could see that the world was changing and we also needed to bring change to the UKOUG. It seemed like the right time to take up the baton and try to bring a fresh approach to the board to see if we can move the group forward.

**I've been on the board since March 2018 and we've changed a lot in that time.** One thing I'm really proud of is that we've consolidated our SIGs into summits to provide an improved member experience. It can be hard for our members to get time out of the office, so when they do manage to, you need to give them as good an experience as you possibly can – and I think the summits do that. We've split our large joint conference into a much more focused set of offerings in relation to each area in the UKOUG: JD Edwards, Applications and Tech. We're also looking

to increase the amount of online content – more webinars, for example. And we're re-engaging with the volunteer communities by trying to give them some additional focus. Plus we're also reacting to changes at Oracle itself, because as the company transforms, grows and moves its focus, we've got to move along with it.

**You'll notice at our events that the people having the most fun are the volunteers.** By volunteering, you get so much more out of it than you put in. Being involved with the organisation exposes you to a wide range of products and technologies that you might not come across in your day job, and that broadens your horizons. But it's also important that we have fun at the same time. Learning's great, but learning while you're enjoying yourself is even better!

► *Neil Chandler is an independent Oracle consultant who has been working in IT for three decades. Turn to page 5 in the Business Apps section to read a point of view from our other Member Advocate, Linda Barker.*



## Mark Hurd, 1957-2019

When Oracle co-CEO Mark Hurd died in October at the age of 62, the tributes poured in quickly (see right). They came from friends, colleagues and rivals, demonstrating the huge amount of respect he commanded, both within Oracle and elsewhere.

Hurd was a undisputed giant of the tech industry, who had served successfully as President and then joint Chief Executive since moving to Oracle in 2010. Prior to that, he spent five years as CEO of HP and, before that, had performed a similar role at NCR, where he started his career 25 years previously.

- Safra Catz, who was Hurd's co-Chief Executive, has now taken over as sole CEO of Oracle while a successor is sought.

### 'WE LOST AN ICONIC LEADER IN HIGH-TECH'

**Larry Ellison, founder and CTO, Oracle:**  
"Oracle has lost a brilliant and beloved leader who personally touched the lives of so many of us... All of us will miss Mark's keen mind and rare ability to analyse, simplify and solve problems quickly. Some of us will miss his friendship and mentorship. I will miss his kindness and sense of humour."

**Marc Benioff, CEO, Salesforce:**  
"I'm so sad to hear of the passing of Mark Hurd. He was always very kind to me and I always enjoyed seeing him at the Warriors [basketball matches]."

**John Chambers, former CEO of Cisco:**  
"We lost an iconic

leader in high-tech... Having known Mark Hurd as a peer and a competitor, I have tremendous respect for him and the differences he has made in the industry. Mark will be missed."

**Bill McDermott, until recently CEO of SAP:**  
"He was a self-made success in the industry and presided over mega accomplishments. While we competed vigorously in the market, we enjoyed professional respect."

**Martin Widlake, President, UKOUG:**  
"Mark was a driving force behind Oracle's cloud strategy and a great friend to Larry Ellison. I'm sure he will miss Mark on both a personal and professional level."

## ORACLE LAUNCH CLOUD FREE TIER

One of the most talked-about announcements at Oracle OpenWorld this year was the launch of the Cloud Free Tier. Most commentators believe the company has made this radical step in order to give its relatively small foothold in the cloud market a boost in the enterprise sphere, and also to hook in users from a young age who many not have previously considered Oracle as an option.

While other cloud providers, such as Amazon, Microsoft and Google, also offer free cloud trials and

limited levels of continued free usage, Oracle claims that its offering goes way beyond its competitors'.

Perhaps the most attractive element of the 'Always Free' offering is the option to have two Oracle Autonomous Databases, with tools like Oracle Application Express (APEX) and Oracle SQL Developer, as well as two OCI compute VMs, block storage devices, object storage and back-up storage.

Whether Oracle's tactic will work remains to be seen, but in the meantime, there's nothing to stop everyone



following the advice Larry Ellison gave at the end of his presentation – risk-free. "Open your laptop and give it a try," he said. You can also turn to pages 11 and 14 in this issue to see some expert analysis and advice on how to get started with Autonomous.

► **Details of the full offering from Oracle are here:** [www.oracle.com/cloud/free](http://www.oracle.com/cloud/free)

## TURN OVER TO BUSINESS APPS...

If you want to find out more about the skills you need to ensure tech adds maximum value to your organisation, flip over to the Business Apps half of this issue, where we take a close look at the evolving role of the IT leader (page 16). You can also see the results of the UKOUG Partner of the Year Awards (page 10), and find out more about some of the biggest issues currently facing business and technology.

We have an exciting line-up of events coming in 2020: they're a perfect opportunity to network with Oracle experts, hear from users facing the same challenges as you – and also a chance to have some fun!

### FIND OUT MORE

For all the latest information on UKOUG events, and to book your place, visit our website: [www.ukoug.org/events2020](http://www.ukoug.org/events2020)



<p><b>27 FEBRUARY</b></p> <p><b>Public Sector Forum</b> (Birmingham)</p> <p><i>More forums coming soon – explore our online calendar for more dates</i></p>	<p><b>12-13 MARCH</b></p> <p><b>OUG Ireland</b> (Dublin)</p>	<p><b>26 MARCH</b></p> <p><b>Higher Education Forum</b> (London)</p> <p><i>More forums coming soon – explore our online calendar for more dates</i></p>	<p><b>MAY (DATE TBC)</b></p> <p><b>OUG Scotland</b> (Glasgow)</p>
<p><b>3 JUNE</b></p> <p><b>Technology Summit</b> (Birmingham)</p>	<p><b>15-16 JUNE</b></p> <p><b>Business Applications Exchange</b> (The Oval, London)</p>	<p><b>23 SEPTEMBER</b></p> <p><b>Licence Management</b> (Location TBC)</p>	<p><b>29 SEPTEMBER</b></p> <p><b>Technology Summit</b> (Leeds)</p>
<p><b>8 OCTOBER</b></p> <p><b>Partner of the Year Awards</b> (London)</p>	<p><b>12 NOVEMBER</b></p> <p><b>Apps Summit</b> (Location TBC)</p>	<p><b>29 NOVEMBER - 2 DECEMBER</b></p> <p><b>Techfest20</b> (Location TBC)</p>	<p><b>UKOUG WEBINARS</b></p> <p>We're also planning more Webinars in 2020, on a wide range of topics relevant to both our Tech and Apps communities. To stay up-to-date on what's coming, go to: <a href="http://www.ukoug.org/events2020">www.ukoug.org/events2020</a></p>



Some of the 2018 winners soaking up the glory and taking to the stage

# And the winners are...

**W**e're proud to announce the winners of the UKOUG Tech Speaker of the Year Awards 2019. As ever, the standard was high, and those who have landed a trophy should be proud. These awards, sponsored by Oracle Groundbreakers, are important to us as we



know how much hard work goes into putting together an engaging presentation – and without our speakers, our events would not be possible.

So, congratulations to all the winners – and thank you, from everyone at the UKOUG, to every single person who spoke at one of our events in 2019.

The trophies are being given out on Monday 2 December at the Welcome session of Techfest19. The Speaker of the Year Awards for our Apps community will be announced at our exciting Business Applications Event in June 2020. For more information on this, and to book, go to: [www.ukoug.org/bax](http://www.ukoug.org/bax).

## The best speakers

The following awards are for those who received the highest feedback ratings at last year's Tech18 conference.

### Best New Oracle Speaker

**Winner:** Roy Swonger

**Subject:** Understanding the New Oracle Database Release & Patching Model

### Best New Overseas Speaker

**Winner:** Amarnath Kothapalli

**Subject:** Introduction to Machine Learning with Oracle DV

### Best New UK Speaker

**Winner:** Amy Grange

**Subject:** Robotic Process Automation

### Best Oracle Speaker

**Winner:** Tammy Bednar

**Subject:** How the Oracle Database Appliance Built-in Security Features can Help to Accelerate the Adoption of Regulations and Compliance

### Best Overseas Speaker

**Winner:** Roel Hartman

**Subject:** Benefits of Developing APEX Applications on the Exadata Express Cloud Service

### Best UK Speaker

**Winner:** Robin Moffat

**Subject:** Embrace the Anarchy (REDUX): Apache Kafka's Role in Modern Data Architectures

## The best judging scores

These awards are for the speakers with the highest scores given to their abstracts by the conference paper judges and conference committee during the Tech18 planning process.

### Highest Oracle Judging Score

**Winner:** Shakeeb Rahman

**Subject:** The Most Amazing Universal Theme Yet

### Highest Overseas Judging Score

**Winner:** Sten Vesterli

**Subject:** Blazingly Fast Application Development – With Visual Builder Cloud Service

### Highest UK Judging Score

**Winner:** Antony Heljula

**Subject:** Autonomous Machine Learning – Data Science for Everyone

## The Oracle Scene Award

This award is for the Oracle Scene magazine article that members spent most time reading in total in our online edition. (In case you're new to the UKOUG, Oracle Scene is the name we used to give to #PTK.)

### Most Read Article in Oracle Scene 2018

**Winner:** Simon Greenwood

**Subject:** Stick = Oracle Forms, Twist = APEX

## Think you can do it too?

Want to speak at a future UKOUG event and share your Oracle knowledge? Then submit your ideas for your presentation here: [www.ukoug.org/contentsubmission](http://www.ukoug.org/contentsubmission)

If you're selected for next year's Techfest, you'll be automatically entered for our Speaker of the Year Awards.

To find out more about the upcoming UKOUG schedule for 2020, take a look at our online events page: [www.ukoug.org/events2020](http://www.ukoug.org/events2020).

# The 5 best things about being a UKOUG member

We polled our members to find out what they love most about the UKOUG. Here are the five most common reasons they gave for joining the group:

Here are the best ways to make the most of your membership:

**To network with other users and Oracle experts**



**To find solutions to Oracle-related problems**



**To develop and raise their professional profile**



**To keep up-to-date with Oracle information – from an independent voice**



**To connect with Oracle partners**



If you're not a member and would like to join – and benefit from all of the above and more – or if you've already joined and want to get more from your membership, visit our website which is full of information, ideas and knowledge-sharing from Oracle users like you:  
[www.ukoug.org](http://www.ukoug.org)

— INDEPENDENT —  
**UK OUG**  
 UK ORACLE USER GROUP

# Highlights from Oracle OpenWorld 2019



As ever, this year's Oracle fest in San Francisco was packed with a huge array of announcements. If you weren't able to be there, here's a handy round-up of some of the key technology highlights

By Biju Thomas

## 1 ALWAYS FREE GEN 2 CLOUD

This is a welcome announcement and I think there will be a lot of developers and administrators making use of this tier. If you do not have an Oracle Cloud account, you still need to sign up for the 30-day Cloud Trial and create your account.

When creating new instances in OCI, you can indicate that you want to use the Always Free tier. The Always Free resources are available as long as you want, with no time constraints – subject only to the capacity limits noted. When your 30-day trial period for the expanded set of services ends, you can continue using Always Free services with no interruption.

Here's a list of the Always Free services:

### DATABASES

- ▶ Autonomous Transaction Processing or Autonomous Data Warehouse
- ▶ 2 databases in total, each with 1 OCPU and 20GB storage
- ▶ Autonomous Databases include APEX for web-based app development, SQL Developer Web for database

development and SQL Notebooks for Machine Learning

### COMPUTE

- ▶ 2 virtual machines with 1/8 OCPU and 1GB memory each

### STORAGE

- ▶ 2 block volumes, 100GB total
- ▶ 10GB object storage
- ▶ 10GB archive storage

### LOAD BALANCER

- ▶ 1 instance, 10Mbps bandwidth

### MONITORING

- ▶ 500 million ingestion datapoints, 1 billion retrieval datapoints

### NOTIFICATIONS

- ▶ 1 million delivery options per month, 1,000 emails sent per month

### OUTBOUND DATA TRANSFER

- ▶ 10TB per month

There are no SLAs associated with the Always Free tier – but what else would you expect? You can find out more information about Always Free here: [www.oracle.com/cloud/free/](http://www.oracle.com/cloud/free/) And turn to page 14 in this issue of #PTK to find out how to get started. ▶

## 2 DATABASE SUPPORT AND ROADMAP

Oracle database non-container architecture has been deprecated since the Oracle 12.2 version. From Oracle 20c onwards, only the container architecture will be supported. The non-container databases are not supported in 20c.

Since a new version is coming out every year, Oracle's release roadmap introduced two types of annual releases. Interim releases (20c, 21c) are supported for two (or three) years – with only premium support and no extended support. Long-term releases (19c, 22c) have four years of premium support and two or more years of extended support.

If you're running the 11.x or 12.x release of databases, consider upgrading to 19c. This is the long-term support release of the 12.x family and has premium support until March 2023 and extended support (for additional cost) until March 2026. Unless 20c and 21c has a feature that you must absolutely have, there is no reason to upgrade to 20c or 21c because those releases are short-lived. So, if you upgrade to 19c now, you have until 2022/2023 to upgrade to 22c.

E-Business Suite announced its support on Oracle Database 19c, with container architecture.

With non-container databases going away, another welcome announcement is the ability to create three PDBs in a container database without paying for additional multi-tenant licences. For more information on this, see: <https://docs.oracle.com/en/database/oracle/oracle-database/19/dblic/Licensing-Information.html#GUID-0F9EB85D-4610-4EDF-89C2-4916A0E7AC87>

Three PDBs are included in all editions of Oracle Database 19c and higher. This announcement would make many organisations consider moving to the CDB architecture. If you create four or more PDBs within one container database, you will need a multi-tenant licence. CDB has many administration benefits. For more on this, see: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/introduction-to-oracle-database.html#GUID-AB84D6C9-4BBE-4D36-992F-2BB85739329F>



## 3 AUTONOMOUS LINUX AND O.S. MANAGEMENT SERVICE

Hot on the heels of the Autonomous Database comes the autonomous operating system. Oracle Autonomous Linux does the following automagically:

- ▶ Automatic provisioning
- ▶ Automatic scaling
- ▶ Automatic tuning
- ▶ Automatic online patching and updating
- ▶ Automatic security monitoring and remediation

To find out more on this, take a look at: [www.oracle.com/cloud/compute/autonomous-linux.html](http://www.oracle.com/cloud/compute/autonomous-linux.html)

Autonomous Linux uses the Ksplice technology to apply patches while the

system is running.

It is 100% compatible with IBM RedHat, thus customers can migrate their RHEL workloads to Oracle Autonomous Linux without any modification. It is included free with Oracle Cloud.

OS Management Service, via the Oracle Cloud Infrastructure console or APIs, and resource governance policies, enables you to automate capabilities that will execute common management tasks for Linux systems, including patch and package management, security and compliance reporting, and configuration management.

## 4 PARTNERSHIP WITH VMWARE AND MICROSOFT

There were a few key announcements about the Oracle OCI & Microsoft Azure partnership. More data centres will have this connectivity in 2020, and it was great to know that the latency between apps running on Azure VM and Database on OCI is around 4ms.

I saw a demo of how to create the interconnect between Azure and OCI. It looks straightforward and easy – from either side.

Oracle Apps (EBS, JDE, etc) will be certified when run on Azure in a cross-cloud configuration.

The partnership roadmap includes:

- ▶ Connecting to Oracle DB via OCI Interconnect
- ▶ Integration with Active Directory
- ▶ Distributed logging using managed ELK on Azure
- ▶ Caching using Coherence cluster

The partnership with VMWare was a surprise, but a very welcome announcement. It will help customers easily migrate VMWare vSphere workloads to Oracle's Generation 2 Cloud Infrastructure (expect

it to go live in Q4 of 2019).

With this, Oracle becomes a partner in the VMware Cloud Provider Program and the Oracle Cloud VMware Solution will be sold by Oracle and its partners.

The solution will be based on VMware Cloud Foundation and will deliver a full stack software-defined data centre (SDDC) including VMware vSphere, NSX, and vSAN. Oracle will also provide technical support for its software running in VMware environments in customer on-premise data centres and Oracle-certified cloud environments. You can find out more here: [www.oracle.com/corporate/press-release/oow19-oracle-and-vmware-091619.html](http://www.oracle.com/corporate/press-release/oow19-oracle-and-vmware-091619.html)

And there was one further announcement related to partnerships: Oracle Marketplace will soon have an option to pay for third-party product licences that are sold on Marketplace, using Oracle Universal Credits. This will consolidate billing for the customer, which includes Oracle services plus third-party services.

## 5 AN EMPHASIS ON MACHINE LEARNING AND ANALYTICS

Oracle Autonomous Database and Oracle Autonomous Linux use advanced machine learning (ML) and autonomous capabilities, which Oracle claims will deliver cost savings, security and availability, and free up critical IT resources to tackle more strategic initiatives. The system delivers automated patching, updates and tuning without human intervention. ML is key in all Oracle innovations.

Oracle is also developing several products where end-users can benefit from machine learning algorithms without knowing too much about the ML internals or data complications. The Oracle Machine Learning notebook and Oracle Analytics Cloud are poised to be great analytics tools for the business.

The AutoML product, which is coming soon, is very exciting. AutoML helps to automatically

build and compare machine learning models.

Here are a few highlights from the presentation given by Charlie Berger, Senior Director of Product Management, Machine Learning, AI and Cognitive Analytics at Oracle:

### AUTO MODEL SELECTION

- ▶ Identify in-database algorithm that achieves highest model quality
- ▶ Find best model faster than with exhaustive search

### AUTO TUNE HYPERPARAMETERS

- ▶ Significantly improve model accuracy
- ▶ Avoid manual or exhaustive search techniques

### AUTO FEATURE SELECTION

- ▶ Reduce the number of features by identifying most predictive
- ▶ Improve performance and accuracy

You can download the presentation from here: [tinyurl.com/y3ygmxml](http://tinyurl.com/y3ygmxml)

## AND A FEW MORE HIGHLIGHTS...

Here's a round-up of some other items that caught my interest (many are not available now, but in the roadmap for Q4 of 2019 and Q1 of 2020)...

▶ **Dedicated VM Hosts:** Single tenant, never share hardware with another customer. Pay only for the dedicated VM host, no additional charge for the VMs running on it.

▶ **Cloud Shell:** Simple browser-based terminal access, available from the OCI console.

▶ **Elastic Instances:** Flexible compute instances giving true elasticity. Customer will be able to pick the number of cores and memory. No downtime required when scaling cores or memory.

▶ **Instance Resize:** Change instance shape (4 core VM to 8 core

VM), instance family (Intel to AMD) and change fault domain without losing any configuration.

▶ **Oracle Maximum Security Zone** where security is not a choice, but always on. Resources launched in this zone will be on dedicated infrastructure

with the highest levels of data encryption and network security.

▶ **Zero Downtime Migration:** Simple database migration to Oracle Cloud leveraging Oracle MAA best practices. This tool will also have the option to migrate from OCI-Classic to OCI-Gen2.

▶ **OCI Application Performance Monitoring:** Allowing OCI and on-premises customers to monitor enterprise applications and diagnose application problems.

▶ **Availability of Microsoft SQL Server on OCI:** MS SQL licence provided by OCI.

▶ **Oracle Data Safe:** A unified control centre for Oracle Databases which helps you understand the sensitivity of your data, evaluate risks to data, mask sensitive data, implement and monitor security controls, assess user security, monitor user activity, and address data security compliance requirements.

▶ **Oracle MySQL service:** Self-managed MySQL Enterprise Edition on Oracle Cloud Infrastructure.

▶ **Exadata X8M:** The industry's first database machine with integrated persistent memory and RoCE, delivers 2.5X performance increase and 10X lower latency. ❌



### ABOUT THE AUTHOR

Biju Thomas is Principal Solutions Architect at OneNeck. He is an Oracle ACE Director and has more than 20 years of Oracle development and database management experience.

# First steps with Oracle Database Autonomous Transaction Processing

Here's how to get started with Oracle's self-driving cloud database – and yes, there really is an 'Always Free' option

By Rita Nuñez



## 10-SECOND SUMMARY

- ▶ Oracle's Autonomous Database takes a huge amount of effort out of getting started as a lot of the work is done for you automatically – it can literally take just five minutes.
- ▶ There is an Always Free version – it's available for students, developers, and enterprise prototypes for an unlimited time.
- ▶ The database is optimised for online transactional processing (OLTP) applications such as e-commerce, order entry, retail sales, human resources and financial transaction systems.



### What is Oracle Database Autonomous Transaction Processing?

It's a kind of database cloud service that Oracle provides. It only runs in the cloud, so is not a product – you can't download it.

### Why all the fuss?

This Database is 'autonomous' – which means it's 'self-driving'. That means you don't have to install the database software, you don't have to configure the instance, you don't have to update it, you don't have to patch it, you don't have to tune the instance, you don't have to manage the storage, and you don't have to think about availability.

It's self-securing, which means Oracle protects the database from external and internal attacks. It's self-protecting, which means Oracle protects against hardware and software failures, site failures and data corruption. When you create an Oracle Autonomous Database, Oracle creates a High Availability Database, so it's an RAC Database with Active Data Guard. Oracle also automatically back up the database every day. Oracle provides the Oracle Autonomous Database service that does all these tasks for you.

### Which database release is used?

This database is an Oracle Database 18c (at the moment, but soon will be 19c) and is an Enterprise Edition. The Autonomous Transaction Processing is optimised for online transactional processing (OLTP) applications such as e-commerce, order entry, retail sales, human resources and financial transaction systems, among others.

### How easy is it to set up a database using this?

With this service, you can create your database in less than five minutes and very easily. Each Oracle Database Autonomous Transaction Processing includes APEX and SQL Developer Web, and there are drivers to connect from many developers' languages: Java, Python, Node.js, .Net, Visual Studio. Also, you can connect from Oracle Tools: SQL Developer, SQLPLUS, SQL Loader, SQLcl and Data Pump. The connection from Client is encrypted so you must download a Client Credential (Wallet).

### Is it really true that there's a free version?

Larry Ellison announced the Oracle Cloud Free Tier at Oracle OpenWorld in September. So, it's possible to get Oracle Autonomous Database for free and test it. Go to [oracle.com/free](https://oracle.com/free) to get set up your Oracle Always Free account. This includes two Oracle Autonomous Databases, two Oracle Cloud Infrastructure Compute VMs; Block, Object, and Archive Storage; Load Balancer and Data Egress; Monitoring and Notifications and more. These services will remain free for students, developers, and enterprise prototypes for an unlimited time. Also, you have 30 days' free trial with US\$300 in credits to try other services.

For more details, go to my blog and see how to create an Oracle Always Free account: <http://bit.ly/AlwaysFreeAccount>. And turn over now to find out how to get started... ▶

# 9 easy steps to Autonomous Transaction Processing

Here's how to get going for the first time in just a few simple steps...

## 1. GET AN ORACLE CLOUD ACCOUNT

To start, you need a free account or choose to pay a subscription in Oracle Cloud. I recommend you start testing this service with a free account. You can do this at: [cloud.oracle.com/free](https://cloud.oracle.com/free).



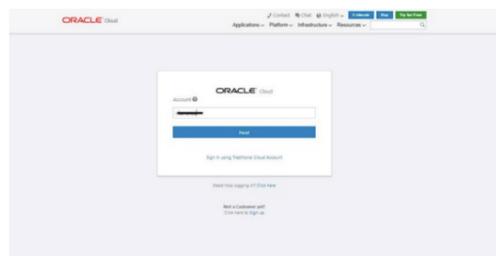
## 2. NOW LOG IN WITH YOUR ORACLE CLOUD ACCOUNT

Go to [cloud.oracle.com](https://cloud.oracle.com).

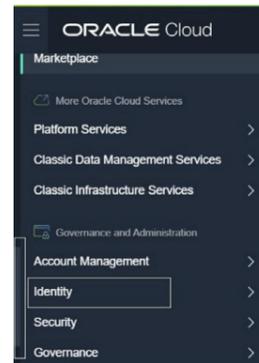
Note that your Oracle Cloud account will not be the same as the Oracle account that you can have for OTN and MOS.

To log in, you need to enter your account name, your username and your password.

An account name is different to a username – the username is your email, but the account name is generic. That means that once you have created your account, you can create more users to log in to the same account.



## 3. CREATE A COMPARTMENT



A compartment is a logical division and the objective is to group your Oracle services.

Scroll the left bar to find **Identity > Compartments** and create a DEMO compartment.

Compartments

Name	Status	OCID	Authorized	Subcompartments	Created
ritanunez (root)	Active	...ckm3ba	Yes	5	-

Create Compartment [help](#) [cancel](#)

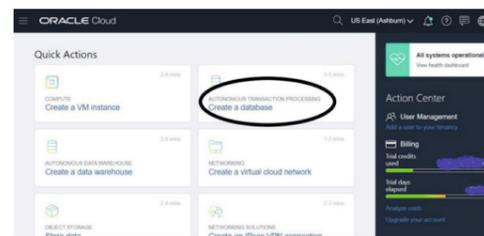
NAME: DEMO

DESCRIPTION:

PARENT COMPARTMENT: ritanunez (root)

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources. [Learn more about tagging](#)

## 4. SELECT AUTONOMOUS TRANSACTION PROCESSING - CREATE AUTONOMOUS DATABASE



Choose **DEMO Compartment**. Enter the display name that you would like, as well as the database name. In this example both are 'ATPTest'.

Provide basic information for the Autonomous Database

Choose a compartment: DEMO

Display name: ATPTest

Database name: ATPTest

## 5. IN YOUR FIRST DATABASE, YOU CAN LEAVE ALL THE DEFAULTS

'Serverless' means that you do not require exclusive hardware. At this point you can dimension the CPU and Storage – 1 CPU and 1TB of storage is the minimum. Auto scaling allows it to automatically assign up to three times the number of CPUs if the workload increases.

Choose a deployment type

Serverless (Selected): Run Autonomous Database without provisioning infrastructure.

Dedicated Infrastructure: Run Autonomous Database on dedicated Exadata infrastructure.

Configure the database

CPU core count: 1

Storage (TB): 1

Auto scaling:  (Selected)

NOTE: If you want to create an Always Free Autonomous Database, select **Always Free** and in this case, resources are limited to 1 OCPU and 0,02TB (20G) of storage. Auto Scaling is un-enabled.

Configure the database

Always Free:  (Selected)

CPU Cores: 1

Storage (TB): 0.02

Auto scaling:

## 6. ENTER THE PASSWORD FOR THE ADMIN USER OF YOUR DATABASE

Create administrator credentials

Username: READ-ONLY

ADMIN

Password: [Redacted]

Confirm password: [Redacted]

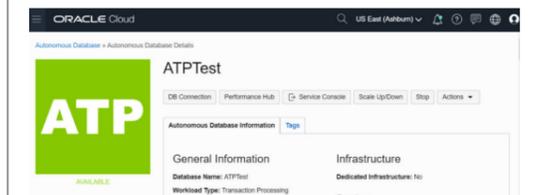
## 7. CHOOSE A LICENCE TYPE

**Bring Your Own Licence** means you already have an Oracle Database licence and you can use it in the cloud. If you don't have the licence, select that you need this service with licence included. For Oracle Always Free Databases the only option enabled is 'license included'.

After this selection, click the **Create Autonomous Database** button.

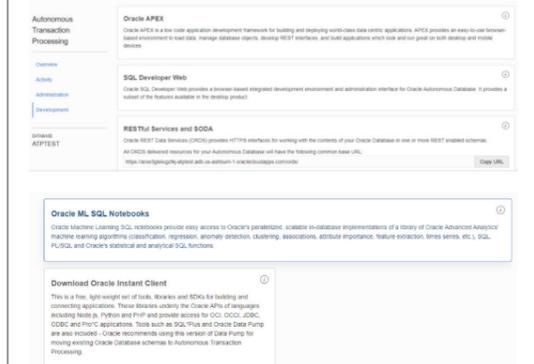
## 8. ORACLE IS PROVISIONING YOUR DATABASE

In a few minutes, ATP provisioning will turn to the 'available' green colour. Now your database is ready!



## 9. START DEVELOPMENT USING APEX OR SQL DEVELOPER WEB

Click **Service Console** and you will see the links. Also, from this screen you have the option to copy URL, use REST, use Oracle ML SQL Notebooks and download Oracle Instant Client.



To learn more, take a look at the documentation for Autonomous Transaction Processing: <https://docs.oracle.com/en/cloud/paas/atp-cloud>

### ABOUT THE AUTHOR

Rita Nuñez is an Oracle ACE Director, an Oracle Database and Oracle Cloud Specialist and an instructor for Oracle University. She is CEO and Founder of Tecnix Solutions.





**B**ased in Stoke-on Trent, with operations in Stafford, Shrewsbury and more recently London, Staffordshire University is known as the ‘connected university’, with its emphasis on putting students first. With a strong underlying set of values – typified by the #ProudToBeStaffs hashtag – the university has improved its performance in all of the major UK higher education league tables over the last few years, achieved TEF Gold (the highest award for teaching excellence) this year and was recognised in the international league tables for the first time in 2018. As a civic university, the institution has strong connections within the city and region, as well as nationally and globally, working with employers, SMEs and other significant partners, and has played an important role in the development of degree apprenticeships in the public sector in England in the last three years.

How appropriate, then, that the finance team at the university are connecting with the digital cloud world and moving to Oracle Fusion, which they did with the help of a team from Namos Solutions. The finance team are keen to get on board with the modern best practices and data-driven intelligence built into the Oracle cloud applications.

In my last instalment (see Issue 69), the project was approaching the User Acceptance Test (UAT) phase and the team were working to close out Systems Integration Testing. The main core functionality was working well, and the University Finance team had been given access to the Systems Integration Test (SIT) environment to encourage them to get familiar with the new screens and to have the first ‘touch and feel’ of the system.

SIT testing was still ongoing, however, as some of the integrations were proving hard to crack and it was becoming apparent that some of them would not be ready for the start of UAT. In particular, the Recurring Payments interface was proving to be a real challenge and the process had to be completely redesigned. The recurring payments process is where credit card transactions are presented to a payment platform and an attempt made to take the funds from the credit cards. The existing process had a design flaw that meant that the majority of transactions being sent to the payment platform were erroring and had no chance of being processed as they did not contain the complete set of information.

We wanted to improve this situation and to be able to only send transactions which would be processed successfully by the payment platform, but the code required to produce the correct set of transactions was more complex than expected. That, together with the fact that the resources required to design the new process were deployed resolving other AR issues, meant that we had to move some of the project milestone dates. The start of UAT and also the go-live date had to be moved back to accommodate the delay on the integrations.

I don’t think I need to explain the difficult conversations ▶

# Mastering the cloud

## at Staffordshire

### University: Part II

In the second instalment of this journey to Oracle Fusion, we take a look at the lessons learned from getting connected

By Richard Lumsden



#### 10-SECOND SUMMARY

▶ Keen to leverage the data-driven intelligence built into Oracle cloud applications, the university finance team used Namos Solutions to help them move to Oracle Fusion.

▶ Bumps along the road included the transition from a development environment to a production environment, automating tax

calculations on POs and matching security cyphers from the Oracle Supplier Network to the Justransform production environment – but these were resolved.

▶ Lessons included the importance of fully understanding the business processes, double-checking the configuration – and aligning with all third parties.

that resulted from this but, as a team, we worked through the facts and data and came up with a plan with new dates that everyone was comfortable with.

### TESTING, TESTING...

Once everything was in place for UAT, the testing ran pretty smoothly. Defects were raised and fixed in a timely fashion and in the words of one of the team “we tested it to death”. We got to the point where the university team were ready to go, and we headed into cutover. Acknowledgement should go to the university finance team in that not only do they know their data well, they also know their processes. They were instrumental in building the cutover plan and advising realistic dates for each of the tasks.

For Oracle Cloud ERP implementations, the approved strategy is that the build of the production environment is a build from scratch using the configuration templates created over the course of the project. From an audit perspective this is not ideal and leads to some interesting conversations with your customer along the lines of, “So we’ve tested the UAT environment and are happy with that – but now you’ve built a completely new one for production – how do we know they are the same?”

At this stage Oracle do not allow you to copy a development environment into the production environment. There are tools to do configuration comparisons, but they do involve a significant amount of technical effort to set up and this is not typically built into the project plan. There must be a better way to avoid this concern, but I have not found one yet.

So, the production environment was built and the first major cutover task was to load the customers and get the customer interface from the SITS system up and running to de-risk the go-live weekend. In the first instalment, I raised the issue of the person type customer load template which has 18 separate tabs. Hats off to the university team as this duly arrived on time and was successfully loaded into the system.

This really was a Herculean effort from a number of parties, and it is testament to the work on previous migrations during the project that this process ran so smoothly. On future university projects we have revised our approach to this customer load and now recommend using an interface for this as the data preparation required is much simpler.

So, when you want to arrange a copy of one environment (POD) to another – a P2T or Production to Test or a T2T Test to Test – you need to provide at least three weeks’ notice. While this is good news for Oracle in terms of resource planning, it’s a nightmare for the project team to specify an exact date when the environment is ready for copying.

Also, there is no method for copying a development environment onto the production environment. This



## Preparation for the production environment build is absolutely key and must be rigorously checked

forces you to plan in a production environment build with all the inherent risks that brings with regard to the potential for human error and timing issues in terms of which module gets configured first. Let’s not forget that the customer has signed off the UAT environment and now you have built a completely new one which is not the one that was tested.

Preparation for this production environment build is absolutely key, and the build must be rigorously checked to confirm that it is an exact copy of UAT. Endless rounds of smoke testing are required in order to be as certain as you can that everything is in good shape.

Another nagging issue that was bubbling away was that, during SIT, we had identified a potential problem with dunning letters – whereby multiple dunning letters, rather than the single letter that was expected, were being produced for customers who had multiple sites. A severity one Service Request was duly raised with Oracle, and this was identified as a bug and a potential showstopper for go-live. The SR was escalated using the standard Oracle process and a planned delivery date was provided which was shortly before go-live.

This was good news in that we would get the fix before go-live, but it was not a perfect situation. It would leave us with a support environment that would be out of step with the production environment at go-live and we would therefore not necessarily be able to recreate production issues in the development environment in order to be able to fix them.

To mitigate against this, we worked with Oracle to find a solution and found that we could have the production

environment release done at the same time as the support environment release. The view was that, as the date for the fix was just before go-live and prior to the first time that the dunning process would be run in production, we would be able to perform testing in the support environment to confirm that the fix had worked prior to having to use the new code in anger in production.

There was, of course, a slight risk that the new code would impact the production processes, but we were given assurance that the code fix was local to dunning letters and would not impact anything else.

### PREPARING FOR GO-LIVE

In order to have the production and development environment releases synchronised, we had to move to the weekly release cadence, and this would allow us to get our fix ahead of the scheduled quarterly release. This involved several phone calls and business case justifications, but everything was nicely lined up for go-live.

Oracle kept us informed of progress but the release date for our fix kept slipping... And finally, it slipped past the go-live date for the project. Thankfully, the University team were flexible enough to be able not to run the dunning process until it got resolved.

The fix eventually arrived, however, was tested successfully and the university are now using it as part of their business-as-usual, but we did make a go/no-go decision based on the original delivery date for the fix. In my view, Oracle need to be aware of these types of situations and the potential impacts of not hitting their delivery dates. In other circumstances we would have had to abort the go-live at a very late stage with all the painful consequences of that for all concerned.

So, the production environment was built and data migration is in full flow. Next in line were the open purchase orders. The load went in 100% – great news... But the tax was not calculating correctly. A check on the configuration revealed that the tax had not been set to auto-calculate. There was no way to do a bulk reset.

A brief SR conversation with Oracle ended up in a statement that a data fix would not be provided as the Oracle functionality had worked correctly. The only way around this was to set the auto-calculate flag and to ‘touch’ each purchase order in the system manually to force an update. This meant assembling a team of consultants to work through a weekend to update the 1,400 purchase orders.

An eager crew was duly put together, provided with a list and asked to go into each purchase order to trigger the tax update. By the end of the weekend, the purchase orders were done and the university team confirmed that everything looked good. Crisis averted – but it does highlight the need to check the configuration fully prior to loading any data.

### ALL SYSTEMS GO!

Finally, it’s cutover day. All the endpoints are pointing in the right direction, suppliers are loaded and the first purchase orders are being sent out. But they don’t arrive! When checking the logs, there is a security cipher issue where the ciphers on the Oracle Supplier Network (OSuN) don’t match those on the Justransform production environment so the purchase orders can’t get through.

Luckily, the UAT Justransform environment is still able to use the ‘out of date’ security ciphers which are being used by OSuN, and we are able to hop the purchase orders through Justransform UAT and through to Justransform production, so order is restored. This does mean that the Justransform UAT environment is temporarily in a production state and can’t be brought down for maintenance until the issue is resolved. Oracle have now updated the security ciphers on OSuN and so this issue has been fixed, but it did mean a late night to get the routing secured and working perfectly.

The university is now live on Financials and have mastered that part of the cloud. So, what next? The university is continuing to push the boundaries with its progressive digital cloud strategy and recently opened the Digital Institute London.

It was a pleasure working with such a knowledgeable team. They are a credit to their organisation and are capable of surmounting any challenge that is put in front of them. The wonders of PBCS await! ☒

### THE LESSONS LEARNED:

- ▶ The pursuit of a quality solution is paramount.
- ▶ Don’t compromise by just doing a straight lift-and-shift exercise.
- ▶ Make sure that you understand the business processes being implemented. It may take a bit longer, but your customer will appreciate the process improvement in the long run.
- ▶ Check your configuration, re-check it and then check it again. The extra time taken will benefit everyone in the long run.
- ▶ Make sure that the third parties you are working with are aligned in terms of having their UAT environment configured the same as production. This will avoid the unexpected go-live issues we came across in terms of security ciphers.

### ABOUT THE AUTHOR

Richard Lumsden is Portfolio Director at Namos Solutions, an Oracle Platinum Partner and Cloud Excellence Implementer (CEI) specialising in the implementation and support of ERP, EPM and HCM solutions.



# Delivering a platform on demand

Boosting Oracle Fusion Middleware with Ansible: how one project was handled using this powerful automation tool – plus some lessons learned

By Michael Mikhailidi

## 10-SECOND SUMMARY

- ▶ RedHat Ansible is a powerful automation tool. It was designed to control operating systems and configurations using Python, YAML and Jinja2 templates.
- ▶ There are plenty of CI/CD systems that provide Ansible connectors and allow you to execute Ansible playbooks.
- ▶ By using Ansible it is possible to: install, configure and start the whole set of Oracle Fusion Middleware products in hours; make sure that all the products are patched and hardened; ensure that all servers and domains have consistent configuration and setup.

Almost as soon as I joined Attain LLC, and started my career in the Federal Services division, I was assigned to two projects with one of our largest clients. Many large institutions have multiple contractor companies, with well-defined terms and boundaries. If you multiply that by very formalised communications and security-driven environments, you can imagine the challenges we had during the project implementation. Additionally, there were the challenges of environment-configuration in production, public-facing applications and outdated architecture design. All those factors meant we had to meet the following challenges:

- ▶ Minimise production-ready platform delivery time
- ▶ Eliminate any discrepancies in the configurations from host to product
- ▶ Have a fully documented environment
- ▶ Track all the configuration changes
- ▶ Follow DevOps principles as much as possible
- ▶ Use open source or existing products.

Attain partners with major software vendors and the project team chose Ansible as an automation tool for the Oracle Fusion Middleware platform delivery. It started as a new generation environment, based on the latest OFM 12.2.1.3 products. Initially, I wasn't sure if Ansible was the right fit for Oracle Fusion Middleware, but even the half-way delivery was so promising that we started to expand coverage to the existing production product set as well.

Now, when our Operations and Support team are adopting our techniques and scripts to monitor and control the other environments, I can share my new experience and show you why Ansible is a good fit for most of the Oracle Fusion Middleware-related activities. But first, I'll introduce Ansible and define some terminology.

## MEET ANSIBLE

Ansible started as an open source project and even if you see RedHat all over the place, the core part of the framework is still open source. We aren't using Ansible Tower in our project, so I'll keep it out of scope. (For more information on Ansible, go to their website, [ansible.com](https://www.ansible.com).)

### To start using Ansible all you need is:

- ▶ An Ansible controller – practically any 64-bit Linux and installed Ansible software, configuration, installed modules, and plugins
- ▶ Setup your SSH keys on targets; Ansible uses key-based authentication to access remote hosts.

- ▶ An inventory file to let your controller know how to find your target hosts
- ▶ Basic knowledge of YAML (Yet Another Markup Language)

That's right: you don't need to install agents or have anything but Python 2.5+ on targets.

## ANSIBLE INVENTORY AND YAML

Ansible supports multiple inventory formats from none (you always have access to *localhost* target) to dynamic inventories where a list of targets created by some program code. Inventory documentation gives you all the examples in two structured formats: INI and YAML. Let's take a look at the sample inventory (see Figure 1).

```
soahost1.vb.mikhail.com
osbhost1.vb.mikhail.com

[webtier]
ohshost1.vb.mikhail.com listen_port=6443 frontend=apps.mikhail.com
ohshost2.vb.mikhail.com listen_port=7443 frontend=apps.mikhail.com

[portal]
wcpghost[1:6].vb.mikhail.com ms_port=8889 secured=yes
```

Figure 1 Ansible inventory in INI format

### It describes 10 targets, where:

- ▶ Single entry servers are soahost1 and osbhost1
- ▶ Machines ohshost1 and ohshost2 under [webtier] section are members of the group *webtier*.
- ▶ The next group *portal* contains six machines through the range. You can access those hosts individually as *wcpghost1*, *wcpghost2* ... *wcpghost6*.
- ▶ Ansible implicitly adds group *all* with all the targets in the inventory.

Now, while you have targets and group of targets defined, you can manage targets at once, using ad hoc commands or playbooks. So far, you don't see anything unusual: technically any orchestration tool provides similar ability – maybe not as easy, but comparable. However, it also allows you to define variables for groups and targets. On our sample hosts in the *portal* group have two variables each: *ms\_port* and *secured*. Variables in this group have the same value for each target, but they are different for the *webtier* group – you may already notice that *listen\_port* has different values.

This is where Ansible gets really powerful: you can create an OHS configuration playbook and refer to the variable *listen\_port*. At runtime, Ansible compiles the actual execution environment and uses the right value for the target.

Inventory in INI format is well suited for the simple environments and a few predefined variables. You can still define groups and subgroups, host and group variables, ▶

but it would be hard to read and trace the dependencies.

Now let's look at how you can do the same with YAML (see Figure 2). Yet Another Markup Language is an attempt to get the best from multiple worlds. It allows you to define structured machine-readable text documents with the minimal overhead for the language syntax. As a result, you can read a YAML document as comfortably as any plain text file.

```
---
all:
  hosts:
    soahost1.vb.mmikhail.com:
    oshost1.vb.mmikhail.com:
  children:
    webtier:
      hosts:
        ohshost1.vb.mmikhail.com:
        listen_port: 6443
        frontend: apps.mmikhail.com
        ohshost2.vb.mmikhail.com:
        listen_port: 7443
        frontend: apps.mmikhail.com
    portal:
      hosts:
        wcpohost[1:6].vb.mmikhail.com:
        ms_port: 8889
        secured: yes
  ...
```

YAML document starts with three dashes  
Now we explicitly define all group. In YAML the line ended with ':' defines a new element (object)  
New element **hosts** contains target definitions  
Sub element defines **target** entry, should be ended by ':'  
Even if target has no variables.  
Element **children** contains subgroups for the group all.  
Group name definition  
Sub element **hosts** for the group targets  
Target name  
target variables can be simple type (String, Boo, Number)  
Or complex – object, list, or any combination.  
Another group portal.  
And group targets  
Same range definition.  
Variables for all hosts in the range.  
Formal YAML file ends with three dots. Could be omitted

Figure 2 Ansible inventory in YAML format

### MODULES, ROLES AND PLAYBOOKS

Ansible offers you four different types of building blocks to build your automation. Documentation and community name them: modules, plugins, roles and playbooks.

In general, modules and plugins are local Python modules, stored on the Ansible controller and available for the Ansible utilities. They are atomic units, with the Ansible API, and perform some useful tasks or manipulations. The main difference between plugin and modules is the execution place: plugin is executed as a part of the Ansible execution process when modules are bricks for roles and playbooks. As an example, a plugin could be a function that escapes `\n` characters with `CHR(13)CHR(10)` to make SQL-compliant strings, while module can be a code that runs prepared SQL on targets systems. You can use modules as ad hoc commands or as a task in plays.

Out of the box, Ansible provides an abundant set of modules and plugins. Even after a year of development, we have not created a custom module or plugin. The main reason why we don't create them are *roles*.

If you can identify and describe an activity, which can be reused by others, you should consider an Ansible role. Examples of roles we have created over the time are: Install and update JDK binaries, RCU wrapper to create and drop repositories for products, TLS enforcement for WebLogic domains and many others. Form a language standpoint, role is a YAML-based set of tasks, and you can use it the same way as you use imported functions in other languages. From the development and design perspective, a role is a predefined folder structure with the module calls (tasks), internal role variables and default variables



(quite similar to function parameters with the default values). Additionally, a role may contain files and templates required for successful execution.

With roles, modules, and plugins, you can create a *playbook*. An Ansible playbook is a YAML document with one or more *plays*. The play is the section in the playbook which prescribes Ansible to execute and list tasks in order against named targets. The play could have variable definitions, connect instructions, and roles to include.

Figure 3 illustrates the play structure:

```
---
- name: Start web servers
  hosts: webtier
  remote_user: oracle
  vars:
    timeout: 5
  tasks:
    - name: Start OHS instances
      include_roles:
        name: ohs-control
      vars:
        state: started
    - name: Wait for servers to start
      wait_for:
        host: "{{ inventory_hostname }}"
        port: "{{ listen_port }}"
      timeout: "{{ timeout }}"
```

YAML file start line  
Name of the play (optional, but useful). Task definition starts with '-', in YAML it's a list item definition.  
Play starts with the target(s) definition  
Specify user name for the SSH connection  
Define common variables for the play  
In our case timeout for the task  
Element tasks is a list of execution steps  
Name for the task in the task list  
Module allows you to call a role as any other modules  
Specify a role to include  
Task variable definitions. Similar to a function arguments  
Specify what role should do with the OHS  
Next task in a list  
Module wait\_for waits while host:port would be open.  
inventory\_hostname is predefined variable, always point to the current target.  
There is no variable listen\_port in play, the value comes from the inventory  
Play has variable with the same name defined, so this one could be omitted

Figure 3 Sample play definition

Ansible always executes tasks on targets in exactly the same order as they are defined in the play. However, Ansible doesn't maintain the execution order for targets, so for the group *webtier*, both targets execute tasks simultaneously.

It's also worth mentioning that targets in play are independent, so if a task fails, Ansible drops all the subsequent tasks for this target while continuing the others.

### MAIN ANSIBLE UTILITIES

I'll skip the Ansible installation part as there is plenty of information out there on how to install packages for your system. Plus, Oracle administrators seldom have privileged access to the OS. So let's make sure that you have a controller with the working Ansible installation. This sample command and output (see Figure 4) are from my virtual machine with Fedora 28 and Ansible 2.7. Your output could differ in the detail, but the meaning will be the same.

```
[mmikhailidi@fed28 ~]$ ansible --version
ansible 2.7.10
  config file = /home/mmikhailidi/ansible.cfg
  configured module search path = [u'/home/mmikhailidi/ansible/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.15 (default, Oct 15 2018, 15:24:06) [GCC 8.1.1 20180712 (Red Hat 8.1.1-5)]
```

Figure 4 Validate Ansible version

Now you know the first Ansible command: **ansible**. It allows you to run modules and even roles from the command line. Let's see how to run a simple module with argument (see Figure 5): *command*. Switch `-m` specifies the module name and with the attribute `-a` you can pass parameters.

If your terminal supports colours, Ansible tools highlight

the output by default. Green means everything is green; yellow tells you that operation has made a change on the remote system, and red always means "Oops". All the other colours are reserved for system output and debug information.

```
[mmikhailidi@fed28 ~]$ ansible localhost -m command -a id
localhost | CHANGED | rc=0 >>
uid=1000(mmikhailidi) gid=1000(mmikhailidi) groups=1000(mmikhailidi)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[mmikhailidi@fed28 ~]$
```

Figure 5 Execute command module with argument

### Let's take a glance at other command line utilities:

► **ansible-playbook** – that command you use to execute playbooks. Many of the command arguments match with the **ansible** command, but it also provides extra options for the playbook management: syntax validation, test executions, execution path validation, debug and step-by-step execution.

► **ansible-inventory** gives you information about the compiled Ansible inventory. You can see a list or tree-like structure with the inventory variables. This tool is a real help, especially if you have a large inventory with the environment details.

► **ansible-vault** covers encrypted data. You can encrypt variables, inline passwords, or some sensitive text data. As a result, you have a text file encrypted by the AES256 algorithm. To access encrypted data, you should provide a password at the execution or create your private password file and configure Ansible to use it.

► **ansible-galaxy** is a role manipulation utility. It works with the public roles catalog (see [galaxy.ansible.com](http://galaxy.ansible.com)) and allows you to import roles from the catalog, or create a new role template on your Ansible controller. If you have written a useful role or you have created the great plugin, **ansible-galaxy** allows you to share your code with the community.

► **ansible-doc** gives you module details description with all arguments and usage examples.

### ANSIBLE AND ORACLE FUSION MIDDLEWARE

Ansible was born to manage OS configuration, so even before we started OFMW delivery and product installations, we realised that Ansible saves us a great deal of precious time on environment validation. Utilities **ansible** and **ansible-playbook** gather detailed information about targets. The main controls we have tested are: installed/missed packages on the host; available block devices, filesystems and free space; what shared volumes are mounted and mount point permissions, firewall rules between machines in the different network zones and load balancer configuration.

Slowly, the project achievements filtered through to the other teams. Now our support and operations team considers Ansible as a tool for daily routines in existing environments. One of the biggest challenges is 'the main ►

switch’ – the environment-wide start/stop procedure. In-product and cross-product dependencies, cache activations and intermittent service calls all push the manual start and validation beyond two hours – with Ansible, we hope to fit the whole process into one hour.

We had to perform some environment-related playbooks ad roles. The biggest one is related to security limitations. We didn’t get all the required Python packages to support Ansible standard certificate modules. I have created a new role on top of ubiquitous OpenSSL. As of now, it covers all needs in the certificate and keystores management. The latest implementation smart enough to request a new certificate for each target collects all of them from the remote machines and sends a rich-formatted email with the attached archive.

Even before we started a simple search through the Ansible Galaxy, our worst suspicions were confirmed: Ansible with Python/YAML and Oracle Fusion Middleware with Java/XML are from different planets. The public repository holds roughly 21,000 roles, where only about 400 are related to Oracle, and only 18 mention WebLogic. Most of the roles in the public repository are simple Java installers or wrappers for the WebLogic scripting language.

Quick code checking in the public GitHub repos confirmed this: without deep customisation, the primary technique for the Oracle technology stack is the orchestrated calls of *template* and *shell* modules. Of course, we use other modules and plugins, but only as a target preparation to perform shell script run. Even with such a limitation, Ansible is much more productive than any of the scripting techniques I have used before, and there are many reasons why.

As a language YAML is quite clean and even if you can’t read it as naturally as promised, it’s still easier to read than any other shell scripts. Modular by nature, Ansible decomposes large automation scenarios into well-isolated roles and modules. We can create, modify and test the role code changed without touching the whole repository. As a result, the automation scenario could be quite sophisticated. For illustration: the SOA installation playbook contains 16 plays with 24 tasks and roles. However, it’s only the tip of the iceberg; the all task estimation reaches 1,700. The code excerpt on Figure 6 illustrates the role usage in the SOA Suite delivery playbook and, for contrast, the piece of the role implementation steps. Now, to create another domain, all you need to know is what variables should be appropriately set.

Speaking of variables: Ansible has a unique approach to the variables and visibility scopes. The documentation names 16 levels of variable precedence. So if your code refers to the variable with name *fmw\_version*, it could be a global variable, group or target variable in the inventory or a declaration in role defaults. Finally, your play can declare *fmw\_version* variable in the playbook. At runtime, Ansible computes the final variable value using all the sources in

the precedence order. For example, we can create new certificate requests for all targets at once. The very same role creates requests with proper attributes such as full and short hostnames, DNS alternatives and load balancers individually for each target.

```
##### Playbook with the role #####
##### Configure SOA Infrastructure #####
- name: Configure SOA Infra properties
  remote_user: oracle
  hosts: soa-admin-group
  tags:
    - domain
    - frontend
    - config
  vars_files:
    - soa-config-vars.yml
  roles:
    - role: ora-soa-config
      vars:
        state: active
##### Core role operations #####
## Configure SOA cluster frontend urls
- name: Configure SOA URLs
  block:
    - template:
      src: soa-{{ fmw_version }}-urls.py.j2
      dest: "{{ tdir_path }}/soa-{{ fmw_version }}-urls.py"
    - name: Update SOA cluster URLs
      shell:
        cmd: |
          cd {{ tdir_path }}
          export WLST_PROPERTIES="-Dweblogic.security.SSL.enableJSSE=true -
          Dweblogic.security.TrustKeyStore=CustomTrust -
          Dweblogic.security.CustomTrustKeyStoreType=JKS -
          Dweblogic.security.CustomTrustKeyStoreFileName=/u01/app/oracle/keystores/trust.jks
          Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2 -
          Dweblogic.MaxMessageSize=300000000"
          {{ fmw_home }}/oracle_common/bin/wlst.sh soa-{{ fmw_version }}-
          urls.py
      register: url_out
    - debug: var= url_out.stdout
```

Figure 6 Role use and role implementation example

While shell and template are the stars in our plays, it’s easy to find supporting crew. Ansible modules were created with the configuration automation in mind. You can manage files and folders (*file* and *tempfile* modules), copy files and content to and from remote hosts (*copy* module), manipulate lines in the text files (*lineinfile*) and many other useful operations. Behind the simple facade, modules offer you a decent set of extra features.

Let’s take a closer look at the *copy* module. Ansible computes file hashes and never copies a file if the hash is the same. It may save you minutes and hours for patch and binaries. Most of the time, when I update something, I want to have a backup copy; now you don’t need to handle backups, just set the *backup* attribute to ‘yes’ and the *copy* module creates a backup with the timestamp in a name.

Built-in Jinja2 templates take a great deal of the burden from our shoulders. With Ansible templates you don’t need a mind-twisted shell code full of *regex*, *grep*, and *awk* spaghettis. All of it is already at your service and much more powerful. Our first approaches to Ansible were not far from the standard shell scripts plus WebLogic scripts, where the shell script facilitates the WebLogic script main logic execution. It has an explanation: the Python language is way more advanced than any version of the Bourne Shell interpreter. Now we have two equally capable tools, and scales tipped toward Ansible/Jinja2. Our latest WebLogic script templates don’t have any flow control, conditions or loops. At most, the exception is handling the calls when you shut down the server and it’s already down or duplicate role assignments. We are moving all the

control logic to the Ansible level. If the classic WebLogic script loops over the parameter list, our latest implementations are a list of commands to execute. In the ideal decomposition state, each list item should have a separate connection, execution and output, but the WLST connection overhead takes too much time.

## LESSONS LEARNED

### 1. Inventory is the key

We started this project with separate inventory and roles set for each product domain. Quite soon, we realised it’s barely possible to maintain multiple role copies and separate, rudimentary inventories. Moreover, I have lost a great deal of Ansible power. You can’t execute commands against all targets, you should maintain multiple copies, and you need to change your configuration or specify inventory every time you run Ansible playbooks. Now we switched to single inventory with the motto: “Inventory is your environment documentation.”

Everything you know about the environment should be in the inventory. For example, we keep our knowledge as inventory variables, and most of them have no direct relation to the hosts. Just a few examples: environment name and lifecycle status, and current data centre code are global variables in our inventory. We use them to generate managed server names, email subjects and many other things. All the WebLogic domain information is stored in group variables, so every server in the group knows how to find the admin server and who is the part of the WebLogic cluster. Of course, the inventory grows big and hard to maintain, but fortunately Ansible allows you to break inventory documents into smaller parts, so basic host and group details are in one file and all product information is in separate documents. At execution time, Ansible compiles all of them into a single data structure.

### 2. Use tags to control book execution

I haven’t said a word about the tags in the main text. However, you find them extremely useful when your playbook grows in size. As in social media, you can tag almost anything in Ansible playbooks and roles (see Figure 6). Later you can include or exclude plays and tasks using the **ansible-playbook** attributes. Imagine the few last configuration steps failed during the SOA installation due to some network connectivity issues. With tagged plays, you don’t need to re-execute all the previous steps to deliver the SOA domain. You specify the tags to include and rerun the playbook, to complete the missed steps.

### 3. Declare role dependencies

You can define a role with a dependency on the other roles. For example, when I need to install software on the target, I call the product installation role. However, you can’t just install the product, because you need to install and prepare Fusion Middleware home first, and before you do so, you should have Java installed and fortified. With

role dependencies, we don’t prescribe all those steps in the books. Playbook has one single play with two tasks: ‘install product SOA 12c’ and ‘patch product SOA 12c’. The installation role follows the dependency chain down to the Java installation role. A single entry in the product book covers all the necessary installation, configuration and patching steps. Of course, you should be mindful going down this road. I’m sure you don’t want to have WebCenter Portal installed just because you run the start command against the wrong server group. There is another thing you should be careful of with dependencies: role durability.

### 4. If it ain’t broken, don’t fix it

Ansible roles declare dependencies, Ansible runs books in parallel on all targets, and you never know which server starts an execution first; if a play fails on one server, Ansible drops all the subsequent tasks. It gives you so many unknowns that the only proper way to create a role is to make it bulletproof. Let’s continue the example of the software installation. When you run a product installation role, you can’t predict the actual state of the target. It may or may not have Oracle JDK installed and configured. It could have Fusion Middleware installed but not patched. Whatever the current state is, your role should handle it. The most straightforward installer in our repository is Oracle JDK installation and performs all these checks before changing the state of the host:

- ▶ Checks if the software is installed (JDK home folder and binary file Java are present)
- ▶ Checks if the software is operable (Runs **java -version** from the target JDK folder) and it matches with the requested version
- ▶ If JDK requested as default, role validates if the Java home link points to the right JDK
- ▶ Validates if certificates and security configuration are identical to the role defaults.

After a year of project evolution and Ansible expansion to the other groups, I can say (with cautious optimism) that we have made a step towards DevOps transformation and implementation. We have significantly improved platform delivery – but how fantastic it would be if the entire environment could be delivered not in months, but hours. ☑

## ABOUT THE AUTHOR

Michael Mikhailidi is a Senior Leader Specialist, Federal Services at Attain LLC. He has worked with Oracle technology products for decades. He is also on the Oracle OEM customer advisory board.



“But that worked yesterday!”

...or why you should test with utPLSQL. Here, we take a detailed look at how to get started

By Samuel Nitsche

**10-SECOND SUMMARY**

- ▶ Automated self-tests are a fundamental part of application and web development nowadays, to the point where it's hard to imagine modern software development without them. Yet they are still surprisingly uncommon in the database.
- ▶ This article shows how easy it is to start developing self-tests in Oracle with utPLSQL version 3 (<https://utplsql.org>).
- ▶ We also take a look at the many benefits these tests can bring to your project and your software.

The lament in the headline might be very familiar to most developers and even more painfully to quite a few users. “What happened between yesterday and today to cause my application to break? And why did it have to be reported by my user?”

After some contemplation, we remember a small change we made yesterday, shortly before leaving for home. The customer asked for a new column in the View `v_starwars_characters` which we added on the fly. Nothing special, but very urgent, and it seemed so simple: just a new field with information about the movies in which the *Star Wars* characters appear (see the table below).

ID	NAME	EPISODES
1	Darth Vader	3,4,5,6
2	Luke Skywalker	4,5,6,7,8
3	Rey	7,8

This is very simple `create or replace view`. Of course

we tested afterwards via `select` if the new column shows the expected results. But if we now try to change the name of a character, we get the following database error:

```
ORA-01733: virtual column not allowed here
```

Maybe you already guessed it, maybe you experienced this special behaviour by the Oracle database yourself: whenever you replace a view, the belonging instead-of trigger is lost. A small, simple change, with big and annoying consequences.

How could we have prevented this error? “Better testing” is an obvious answer and it won't take long until we get to checklists and standardised test protocols to be sure that every important aspect of our application is tested, after every change.

That's the point where automated self-tests become interesting, because they are exactly that: a number of narrowly defined, standardised checks of a certain functionality — for example, whether updating a specific column of a view is possible.

We can create such a test with straight standard PL/SQL like this:

```
declare
  l_name varchar2(2000);
begin
  update v_starwars_characters set name =
    'Anakin Skywalker' where id = 1;
  select name into l_name
    from v_starwars_characters where id = 1;

  if ( l_name <> 'Anakin Skywalker' ) then
    raise_application_error(-20000, 'Update did
not work!');
  end if;
end;
```

We could now save this block as a script and run it after every update. However, this would become pretty confusing and tedious with a growing number of tests, which is why it's helpful to use utPLSQL. The free, actively maintained open source framework can take a lot of work out of our hands.

**INSTALLING utPLSQL**

To use utPLSQL it has to be installed into the database first. To do this you can download the latest release version from GitHub (<https://github.com/utPLSQL/utPLSQL/releases/latest>). You can find some prepared scripts in the folder “source” to smoothly install utPLSQL:

- ▶ `install_headless.sql` to do a default-installation into schema “ut3” with public synonyms (run as sys-user)
- ▶ `install.sql` to install it into a different schema (no sys-user needed). In that case you need to grant access to the utPLSQL functions by running one of the following scripts:
  - ▶ `create_synonyms_and_grants_for_public.sql` for all users
  - ▶ `create_user_grants.sql` and `create_user_synonyms.sql` for a specific user

A detailed installation guideline can be found at: <http://utplsql.org/utPLSQL/latest/userguide/install.html> or in folder `docs/userguide/install.html` of the downloaded ZIP archive.

**A FIRST TEST WITH utPLSQL**

While most PL/SQL developers are used to the procedural model and rely heavily on packages, utPLSQL uses the object-oriented possibilities of the Oracle database and provides a “fluent” API. It might look odd at first, but you will get used to it quickly and probably come to love it.

utPLSQL provides some very helpful public methods after its installation, first of all the so-called “expectations” (see Figure 1).

```
ut.expect(actualValue)
  .to_equal(expectedValue)
  .to_be_greater_than(value)
  .to_be_between(min, max)
  .to_be_like('%partialString%')
  .not_to_be_null()
  .not_to_be_less_than(value)
...
```

Figure 1: utPLSQL Expectations. For PL/SQL, the object-oriented, “fluent” syntax might be a bit unfamiliar

The second key part of the framework is the “annotations” with which we can turn regular PL/SQL packages into test-suites. This is a utPLSQL test-suite header with annotations `%suite` und `%test` – these are getting parsed and interpreted by the framework, but only in the package-headers:

```
create or replace package ut_v_starwars_characters as
  -- %suite(View: V_STARWARS_CHARACTERS)

  -- %test(Update character-name via view)
  procedure update_name;
end;
```

Here, `%suite` is the only annotation that is absolutely necessary. It tells utPLSQL that the package is in fact a test-suite. With `%test` we mark the subsequent procedure as a test, which we can now implement inside the package body as usual:

```
create or replace package body ut_v_starwars_characters as
  procedure update_name
  as
    l_actual_name v_starwars_characters.name%-
type;
  begin
    -- Arrange: Setup test-data
    insert into star_wars_characters (id, name)
      values (-1, 'Test-Char');

    -- Act: Do the actual update
    update v_starwars_characters set name =
      'Darth utPLSQL' where id = -1;

    -- Assert: Check the output
    select name into l_actual_name from
      v_starwars_characters where id = -1;
    ut.expect(l_actual_name).to_equal('Darth
utPLSQL');
  end;
end;
```

In a similar way to the standard PL/SQL example, we execute an update on the view. Afterwards we select the column we just updated and compare the result with our expected value via utPLSQL-expectation.

One problem we face in a test scenario is that we don't know exactly which data might exist in our current database state. Therefore, we initially make sure we have data to update. In our example, we use a little trick to avoid collisions with eventually existing data: most primary keys are INTEGERS, but if the related Sequence or Identity starts with 1, we can just use negative IDs for our test data.

Now we can run the test with the `ut.run`-method. Here's the result of the unit-test:

```
set serveroutput on
call ut.run('ut_v_starwars_characters');
View: V_STARWARS_CHARACTERS
Update character-name via view [.002 sec]
(FAILED - 1)

Failures:
```

```
1) update_name
   ORA-01732: Data manipulation operation not
legal on this view
   ORA-06512: in «SITHDB.UT_V_STARWARS_
CHARACTERS», line 10
   ORA-06512: in line 6
Finished in .002346 seconds
1 tests, 0 failed, 1 errored, 0 disabled, 0
warning(s)
```

As expected the test fails and utPLSQL even provides the complete stack trace.

It's also noteworthy that we don't get a “failed” but an “errored” test in the final test run summary. The test caused an ORA-exception, which the framework caught and documented for us. If we had more tests, those would nonetheless be run.

Now we can address the essential problem and re-create the lost instead-of trigger (for demonstration without any functionality first):

```
create or replace trigger save_v_starwars_
characters
  instead of update on v_starwars_characters
  for each row
  begin
    null;
  end;
```

If we now re-run our test-suite, we get a slightly different result:

```
call ut.run();
View: V_STARWARS_CHARACTERS
Update character-name via view [.409 sec]
(FAILED - 1)

Failures:
1) update_name
   Actual: <Test-Char> (varchar2) was
expected to equal: <Darth utPLSQL' (varchar2)
   at «SITHDB.UT_V_STARWARS_CHARACTERS.
UPDATE_NAME», line 14 ut.expect(l_actual_name).
to_equal('Darth utPLSQL');

Finished in .41099 seconds
1 tests, 1 failed, 0 errored, 0 disabled, 0
warning(s)
```

The test is still failing, but this time it's our expectation that caused the failure and not an ORA-exception. The failure report also tells us very specifically what went wrong and in which line the expectation failed.

Now we implement the trigger completely and run our test-suite again:

```
create or replace trigger save_v_starwars_
characters
  instead of update on v_starwars_characters
  for each row
  begin
    update star_wars_characters
      set name = :new.name
      where id = :new.id;
  end;

call ut.run();
View: V_STARWARS_CHARACTERS
Update character-name via view [.004 sec]
```

```
Finished in .005951 seconds
1 tests, 0 failed, 0 errored, 0 disabled, 0
warning(s)
```

Our view works as expected – and the next time we lose the trigger, we will notice it.

## BENEFITS OF AUTOMATED TESTS

As you’re reading this article, the probability is very high that you don’t need to be convinced about the advantages of automated tests. But I still want to list some of the benefits automated tests provide in various ways:

- ▶ Automated tests are “change detectors” which warn us about changed functionality (no matter if these changes are intended or – like in our example – unintended)
- ▶ They are transportable and can be run (almost costless) on different systems
- ▶ If they are implemented to confirm bug fixes, they prevent known errors from reappearing
- ▶ They can serve as a confirmation for agreed-upon requirements of the software

These benefits are valid for all kinds of automated tests. When we assume tests that are written by the developers themselves like in this example, there are even some more benefits:

- ▶ The creation of tests can help to shift the focus from the *how* to the *what* and to look at a functionality from different perspectives
- ▶ Well-written tests can serve as code example and documentation for how to use a certain functionality or what it is meant to do
- ▶ Self-tests encourage the creation of “simpler” programming constructs, because they are easier to test. This has a positive impact on the maintainability of the code.

From my point of view, the most important advantage of a solid, automated test base, however, is that it creates the preconditions to improve our own code continuously and confidently in order to do continuous refactoring.

## ANOTHER TEST AND MORE ANNOTATIONS

We already secured one functionality of our view, but the new column, in particular, contains logic of significant complexity. That makes it worthwhile securing the behaviour via an automated self-test – especially if we expect the functionality or implementation to change in future.

We define another test and also use a new utPLSQL annotation:

```
create or replace package ut_v_starwars_
characters as
  -- %suite(View: V_STARWARS_CHARACTERS)

  -- %beforeall
  procedure setup_test_data;

  -- %test(Update character-name via view)
  procedure update_name;

  -- %test(View returns correct list of
  episodes)
  procedure return_list_of_episodes;
end;
```

The procedure following the `%beforeall`-annotation is run once per test-suite, before all of its tests.

utPLSQL provides a variety of annotations which allow you to

decouple setup and cleanup from the actual test:

- `%beforeall`
- `%beforeeach`
- `%beforetest`
- `%aftertest`
- `%aftereach`
- `%afterall`

The benefit of this annotation is revealed when we look at the implementation:

```
create or replace package body ut_v_starwars_
characters as
```

```
function get_view_row
return v_starwars_characters%rowtype
as
  l_result v_starwars_characters%rowtype;
begin
  select * into l_result
  from v_starwars_characters
  where id = -1;
  return l_result;
end;
```

```
procedure setup_test_data
as
begin
  insert into star_wars_characters (id, name)
values (-1, 'Test-Char');
  insert into appearance_in_episode
(character_fk, episode_no)
values (-1, 3 );
  insert into appearance_in_episode
(character_fk, episode_no)
values (-1, 5 );
end;
```

```
procedure update_name
as
begin
  update v_starwars_characters set name =
'Darth utPLSQL' where id = -1;
```

```
ut.expect(get_view_row().name)
.to_equal('Darth utPLSQL');
end;
```

```
procedure return_list_of_episodes
as
begin
  ut.expect(get_view_row().episodes)
.to_equal('3,5');
end;
```

Now `setup_test_data` creates the situation we need to execute both tests: one entry in the table `star_wars_characters` and two entries in the table `appearance_in_episode`. These tables are the source of the view we want to test.

The tests themselves are pretty easy to read and understand – to increase the readability we added the helper function `get_view_row()` which returns the complete view-row. To do the check we once again use the `to_equal` expectation.

Maybe you’ve been wondering what happens to the test data we create?

utPLSQL works with savepoints and rollbacks when operating in standard mode. Before each suite and before each test a savepoint is created to which the session is rolled back after finishing the test or suite. That means that all the data changes we do, including our test data, are automatically undone at the end of a test run (see Figure 2).

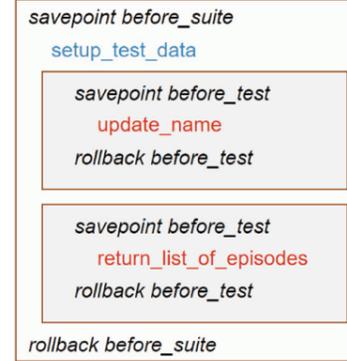


Figure 2: Succession and savepoints of the test-suite

It also means that we can’t test functions in this mode which contain transaction control like `commit` and `rollback` or DDL.

To secure these scenarios with utPLSQL tests we add the annotation `%rollback(manual)` right below `%suite`. Of course, we need to take care to clean up any changes ourselves now (for example via a `%afterall` method).

For all situations that don’t require transaction control or DDL, however, the rollback-mechanism provides a huge simplification of cleanup.

## SECURING EDGE CASES

What we tested and secured so far are the ways in which we expect our application to be normally used. But what happens if we have a *Star Wars* character who doesn’t appear in any of the movies – for example the quite popular Ahsoka Tano from the *Clone Wars* series?

It’s important to think beyond the usual use cases, beyond what we expect or want the user to do. The chances are high that users will use our application in ways we wouldn’t normally expect.

Let’s write another test for the characters without a movie appearance then:

```
-- Package-Header
...
-- %test(View returns row but empty list of
episodes when character has no appearance)
procedure return_empty_list_of_episodes;
...
-- Package-Body
...
procedure return_empty_list_of_episodes
as
begin
  delete from appearance_in_episode where
character_fk = -1;

  ut.expect(get_view_row().episodes)
.to_be_null();
end;
```

If this test is run successfully, we proved two things at once:

- ▶ Even if a character doesn’t appear in the movies, the view returns a row (otherwise a `NO_DATA_FOUND` exception would be thrown)
- ▶ The value of column `EPISODES` is NULL in that case.

We could go on and test whether the view-row still returns the expected name and not `NULL`. This shows a dilemma we face when writing automated self-tests.

## HOW MANY TESTS ARE ENOUGH?

How far should we go with testing? Should we secure every little eventuality with a test – in our case, for example, that inserts are not allowed or that the name cannot be changed to a name that already exists? This is something only you can answer, because only you know the circumstances and risks of your project and your database application. However, the following questions can be helpful when making a decision:

- ▶ How severe are the consequences when a certain functionality doesn’t behave as expected?
- ▶ How likely is it that a certain case happens (e.g. a table-constraint is removed by accident and therefore multiple entries with the same name would become possible)?
- ▶ How likely is the code of this functionality to be changed and how often will it happen?
- ▶ How difficult or costly is it to test with an automated self-test?

Software development is often about compromises and it’s exactly the same with creating automated tests. Whether and to what extent they are valuable depends a lot on your goals, your development process, your company and project circumstances.

Even a few tests which secure existing behaviour on a relatively high level can be very beneficial and sufficient for some projects.

If you develop software which is to be maintained, extended and improved over years, a more detailed base of unit tests that allows continuous refactoring of your code will provide additional value and improve your development velocity and quality.

**My tip:** Start small and experiment. Every time an error occurs or gets reported you have to analyse and try to reproduce it. You can usually do this in a way that can be used as a test setup. When the error is found and fixed, you already have an automated test that prevents this error from happening again.

Every beginning is hard, just keep going! Experiment and observe what helps you. It’s often not the big steps that bring sustainable change, but the small, steady ones that become part of the daily routine. ☒

## MORE INFORMATION AND TOOLS

You can find a lot of information around utPLSQL in the Resources section of [utpls.org](https://utpls.org):

<https://utpls.org/resources>

Meanwhile there is also a comprehensive number of tools around utPLSQL:

- ▶ SQLDeveloper Plugin: <https://www.salvis.com/blog/2019/07/06/running-utpls-tests-in-sql-developer/>
- ▶ Command-line interface: <https://github.com/utPLSQL/utPLSQL-cli>
- ▶ Maven-Plugin: <https://github.com/utPLSQL/utPLSQL-maven-plugin>
- ▶ Demo-Project incl. CI/CD integration in Travis: <https://github.com/utPLSQL/utPLSQL-demo-project>

All code examples including setup can be found here: <http://bit.ly/sithdb-aber-das-hat-gestern-noch-funktioniert>



## ABOUT THE AUTHOR

Samuel Nitsche is a software-developer for Smart Enterprise Solutions and an Oracle ACE with nearly 20 years of experience. He writes regularly about database development and is part of the core development team of utPLSQL. You can read his blog here: [cleandatabase.wordpress.com](http://cleandatabase.wordpress.com)