\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Oracle EMEA Pre-sales – Digital Business Platform v3

***Oracle APEX – Developer Cloud Service Integration Demo***

***----------------------------------------------------------------------------------***

Jules Lane – EMEA Specialist Pre-sales

Edi Vasermann – Cloud Pre-sales

April 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contents

[Demo Attributes 2](#_Toc444850703)

[Executive Summary 2](#_Toc444850704)

[What’s New in this Release 3](#_Toc444850705)

[Storyline and Personas 3](#_Toc444850706)

[Supporting Material 4](#_Toc444850707)

[Prerequisites 4](#_Toc444850708)

[Data used **Error! Bookmark not defined.**](#_Toc444850709)

[Known Issues 4](#_Toc444850710)

[Demo Flow 6](#_Toc444850711)

## Demo Attributes

|  |  |
| --- | --- |
| **Product(s)** | Oracle Application Express 5.0, Oracle Database 12.1, Oracle Database Cloud Service, Oracle Developer Cloud Service 16.1 |
| **Date last updated** | April 19, 2016 |
| **Author(s)** | Jules Lane |
| **GSE Demo Release Version(s)** |  |
| **Demo Title(s)** | Oracle APEX – DevCS Integration |

## Executive Summary

Oracle Application Express (APEX 5) is the fastest and easiest way to develop database-centric web and mobile apps, either in the cloud or on premise. It is included as a standard feature of all Oracle databases and is pre-configured and ready to run in all Oracle Database Cloud Services. It is ideal for very rapid and iterative development methods and for larger projects has traditionally been used in conjunction with 3rd party version control systems.

Oracle Developer Cloud Service is a cloud-based software environment for your application development infrastructure. It provides an open source, standards-based solution to manage and automate the control, collaboration, and deployment applications within Oracle Cloud. It helps you to manage the application development life cycle effectively through integration with the widely used Hudson, Git and Maven tools, as well as issue tracking, and team wikis. DevCS integrates with all appropriate Oracle cloud services as well as on-premise environments.

This demonstration provides an illustration of how the Oracle Developer Cloud Service can be used to manage APEX development artifacts and environments, thus enabling APEX projects to be governed in the same way as other development technologies, such as Java, Python, Ruby, Javascript etc.

Other demonstrations are available to showcase the wider functionality of both APEX and DevCS.

ARCHITECTURE DIAGRAM

## What’s New in this Release

* First Release – xx

## Storyline and Personas

The Demo User IDs and persona roles for this demo are:

|  |  |
| --- | --- |
| **Demo User ID (login)** | **Persona Roles** |
| APEX developer (on prem or cloud) | APEX app developer / administrator |
| DevCS user | DevCS user (in reality the same person) |

We want to use DevCS to manage APEX source code and automate the deployment of new or updated APEX apps to a cloud based test environment.

The scenario is;

1. Run (or build and then run) a simple app in the APEX cloud environment
2. Login to target APEX environment to show that the APEX workspace is not there
3. Export app components to sql and xml files
4. Copy files to local local GIT repository
5. Using Git, upload files into the DevCS GIT repository
6. Explain and show main DevCS capabilities
7. Run a pre-defined set of Hudson jobs to;
   1. Execute a script to provision a new DBaaS environment (optional – takes 30 mins)
   2. Execute a script to import the sql and XML files into the APEX environment in the new DBaaS service in order to create the new APEX Workspace, schema objects, data and APEX app.
8. Run the APEX app in the new DBaaS APEX environment.
9. Repeat 7b and 8 to create and run an updated version of the app.

## Supporting Material

* [APEX](http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html)
* [Database Cloud Service](https://cloud.oracle.com/en_US/database?resolvetemplatefordevice=true)
* [Developer Cloud Service](https://cloud.oracle.com/en_US/developer_service?resolvetemplatefordevice=true).

## Prerequisites

* Any modern browser
* Git installed on your laptop, optionally with a Git client such as SourceTree.
* Database cloud and DevCS accounts – not necessarily in the same identity domain

## Known Issues

* ??

## In advance

* >>> Instructions to set up the DevCS environment are in this document (upload to beehive).
  + - Update Sourcetree with the current GSE password in – Tools 🡪 Options / Authentication
    - Run the DevCS Build *Dropall* to remove workspace JULES from the target environment

## Login / environment details

Replace the following with details of the environments *you* plan to use.

**1/ APEX Development Environment**

* APEX on my local laptop
  + jules / jules / Pitta111
* Local git repository (origin)
  + C:JL\APEXDevCS\git (this was originally cloned from the master on DevCS, put there originally by Edi)

**2/ DevCS environment**

* + url = https://developer.us2.oraclecloud.com/developer73023-paasdem0002/#projects/developer73023-paasdem0002\_jules-1/
  + Cloud Identity Domain = Paasdem0002
  + Cloud User = cloud.admin
  + Cloud pwd = <get this weeks from GSE launchpad>
  + Git Repository (master) = APEX DevCS.git
    - The build *Dropall* should be run *in advance* to remove workspace JULES from the target APEX environment

**3/ APEX target (test) Environment**

* Database Cloud environment (if created in advance, not created during the demo)
  + Cloud Identity Domain = PaasDemo0002
  + Cloud User = cloud.admin
  + Cloud pwd = <get this weeks from GSE launchpad>
  + System / Db pwd = Pitta111\_
  + APEX Admin pwd = Pitta222”
* APEX
  + url = https://129.191.5.61/ords/pdb1/f?p=4550:1
    - internal / admin / Pitta222”
    - jules / jules / Pitta111 (to be created during the demo)

## Demo Flow

* Login to target environment (internal) and show no workspace called JULES (and therefore no schema, no app)
* Optionally also use SQL Dev to show no Schema (User) called JULES in that Db.

**Move entire new app from Dev to Test via DevCS**

1. Run any APEX app – note the workspace and schema names
   1. Optionally make a change to the app.
2. Export – save to file and put in local project repository.
   1. Rename file as buildApexApp.sql (case sensitive)
   2. Move file to C:\JL\APEXDevCS\dbcs\sql – REPLACE (this is in the local git repository – overwrites the old version of the file)
   3. Make any small update to C:\JL\APEXDevCS\Docs\GSE Pwords.txt and save.
3. Sourcetree
   1. Select Log/History tab at the bottom
   2. Select 1st line (Uncommitted changes) – Updated files shows up as yellow (git command line equivalent is “git status”)
   3. Select earlier versions to see changes made then
   4. Click on the text file (or Word file) to see diffs of the content from last version
   5. COMMIT with comment (git command line equivalent is “git commit –m “comment text”)
4. DevCS / Code (= git repository)
   1. Explain structure
   2. Show scripts, that call sql files
   3. End showing \ dbcs \ sql
5. Sourcetree
   1. Push (note the number of changes to be pushed)
   2. Show full Output (quick)
   3. Click on new most recent line, showing commit comment.
6. DevCS / Home
   1. Note recent action
   2. Click on commit number
   3. Shows content changes (may just be the date in the sql file)
   4. Click Code – dbcs – sql. Note commit comment and source user.

**Automated Builds (= Hudson continuous deployment engine)**

1. Show Builds and explain
   1. xx
2. Run CopySQLFiles – which triggers all the others in turn (LoadAPEXData may be queued for some minutes)
3. Watch Build Queue and Job Stats chart

**See results in target environment**

1. Go to target environment and refresh to show new workspace
2. Login to workspace jules / jules / Pitta111 (will need to change pwd – use Pitta222)
3. SQL Workshop to show tables with data.
4. View / run Climbing Logbook app – same as it was in the dev environment
5. Optionally also view new Schema (User) in SQL Dev.

**Update just the App and move to Test**

1. Make a change to the app (switch to a different ThemeRoller style, change text on login page or add a new field to a form etc)
2. Export – save to desktop and rename as updateApexApp.sql (CASE sensitive)
3. Move file to c:\JL\apex-demo\APEXDevCS\dbcs\sql – REPLACE what is there(overwrites the old version)
4. Sourcetree commit and then push
5. View in DevCS code
6. Show and run build updateApexApp
   1. Possible enhancement – make this build step and script pick up the same buildApexApp.sql file as the build process to allow for changes only to the apex app and nothing else.
7. In APEX workspace refresh the app to show change.

**Demo Details**

| **View** | **Click stream** | **Talk stream** |
| --- | --- | --- |
|  | Login to your Dev APEX;  **Note** – Feel free to use your own APEX environment.  . | We log in to the APEX development Environment.  Note the workspace we are using |
|  | Go to the Application Builder |  |
|  | Click on the xxx App |  |
|  | Click Run |  |
|  | Login to the target APEX admin environment.  Internal / admin / <db pwd> |  |
|  | Show that our workspace does not exist yet. | We want to use DevCS to recreate this workspace in a target environment (most likely, test) and then also migrate the app into it. |
|  | Back in our dev environment |  |
|  | Export the App to a file.  Locally rename the file to be; |  |
|  | Scripts  Edit and then download the DDL for tables needed for the app. |  |
|  | Export data for a table to a file.  Show other data files already unloaded. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |