**White Paper**

**A Technical Overview of New prototype for creating ORACLE Databases**

**OR**

**DiOC - Deliver Oracle Database in One Click**

**Abstract**  
  
This white paper describes deploying ORACLE database on Docker platform. Here we will highlight one of the best techniques known today to create database extremely fast with less efforts.

**September 2015**

***TABLE OF CONTENTS***

1. Introduction
2. Audience
3. Technologies Used
4. Architecture Overview
   1. Technical Design
   2. Workflow
   3. Logging
   4. Security
5. Prototype in Action
6. Prototype Source Code
7. Conclusion
8. References

**Introduction**  
  
The need for more sophisticated approach for creating oracle database is becoming increasingly important for organizations. Delivering solution and serving large volume of requests in such short span of time is quite challenging for DBAs. DBAs or Oracle Advanced users can deliver a database more effectively using this technique. This idea will speed up the deployment for the ORACLE infrastructure. This will spin up a Docker container having ORACLE software installed and database created on it automatically with just one click. Yes, I’ve just written, “One Click”. This is just an initial prototype, but it can surely be customized and enhanced based on customer’s requirements.

**Assumptions:**

🗸 Customer needs a database on urgent basis (Almost instantly)

🗸 Customer needs a multiple databases in one go (i.e. 10 database for some classroom/Vendor training)

🗸 Customer needs a dedicated environment (One database instance per container)

🗸 Customer just wants to play around and do some PoC on ORACLE database with high privileges credential (DBA privileges and Linux Administrator privileges) and do not need regular DBA support

🗸 Technical (DBA) team needs to train new resources on ORACLE Database tasks – onboarding/offboarding

**Audience**

This white paper is intended for Oracle Database Administrator (DBAs), Docker Administrators, Linux Administrators, storage administrators, IT architects, and technical managers responsible for designing, creating, and managing ORACLE databases, infrastructures and data centers.

**Technologies Used**

Docker, OracleLinux, PHP, HTML5, Bootstrap

**Architecture Overview**

1. Technical Design

**Docker File:**

It contains a set of instructions to build a container. It is being used to create a base image. It has command set to load the OS image, installing required packages, setting up volumes with appropriate permissions, kernel parameters, user creation, and Oracle software installation.

**Manager Process:**

It feeds end user inputs to all generators and respective generator will build a script to be executed. Once all the scripts have been executed, it creates the container using docker commands. Container will be ready with Oracle Software and Oracle Database on it. It also sends email notification to end user with all the details about container.

For example, Database create generator will generate a code with the required name provided by end user, so once the container will be created, it can be used to create database on it.

**Database Create Generator:**

Manager process invokes this generator with required inputs, and it generates the code to create the database in a container.

**Database listener generator:**

Manager process invokes this generator with required inputs, and it generates the code to create the listener.ora file in a container

**Database init file generator:**

Manager process invokes this generator with required inputs and it generates the code to create the initSID.ora file in a container which in turn will be used by create database script as an argument.

User will have access to all above generated files in a container.

**Sample Notification Email:**

Important information regarding ORACLE database creation:

Host name

Database Name

SSH Port

Listener Port

Volume

TNS Entry

Database Image used

IMPORTANT NOTICE:

Default Credential

For SSH to Linux box

For Database Remote Connection

1. **Work flow**



Enter all required information

Login to Web Interface

Send Notification

Database Init file Generator

Database Listener Generator

Master process

Database Create Generator

Click on Order Now Button

Figure 1

\*\* Master process: Invokes all generators and execute Docker commands to create the container.

1. **Logging**

You can see the details of container which has been created earlier.

1. **Security**

**Conclusion:**

**This will be a huge time saver and also a resource saver as it may cut down on the need for building servers to test with.**