Oracle XE Service Broker Tile

Customers are increasingly interested in using Developmental version of Oracle Database (referred to as Oracle XE) on PCF. They would like to spin up instances of Oracle XE and have its services available on the service marketplace via an associated service broker on CF.

Oracle XE Service Broker

An open source [Oracle service broker](#) written by Vinicius Carvalho allows creation of managed tablespace and profiles on Oracle XE Database instance using CF's service broker interface. This allows users to consume Oracle XE database as a service to applications on CF. When a service instance is created, a new profile is created with limited resource availability (connections, space etc.) on the Oracle XE instance and binding creates the managed user profile for that profile.
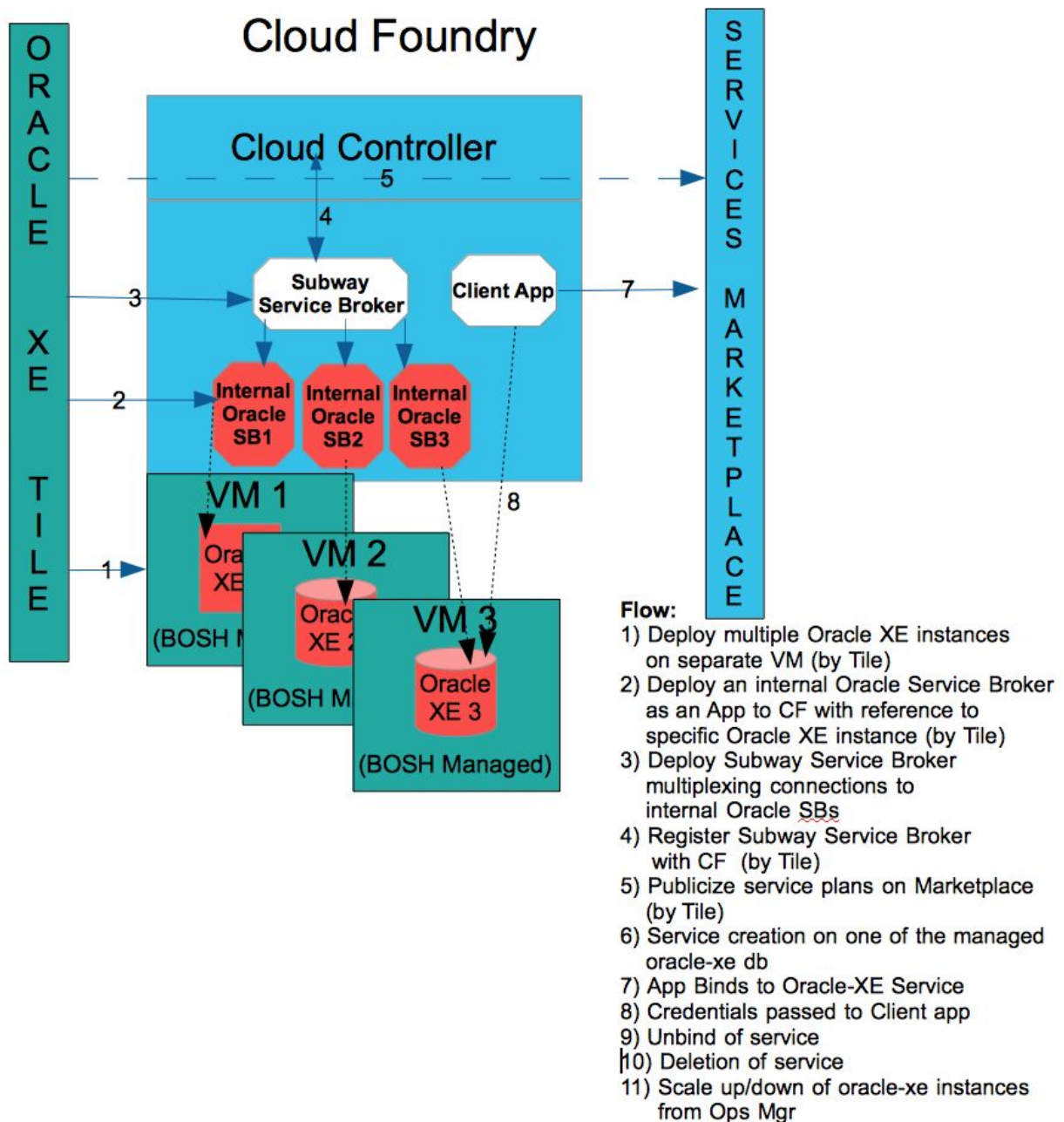
BOSH Release

Oracle XE database instance would need persistent storage and external addressing that is best managed by Bosh. A Bosh release allows install and management of the Oracle XE database instance on Bosh. The release also comprises of Bosh errands to push and register the broker on CF. A one-to-one mapping exists between an Oracle XE instance and its service broker. As Bosh scales up the number of Oracle XE instances, additional service broker instances would be deployed to CF. But this would mean surplus of service brokers on the marketplace. In order to present a single unified version of the Oracle XE service broker, CF Subway Service Broker was used to multiplex requests across multiple underlying Oracle XE Service Brokers. Subway would act as a facade to CF and route actual service creation and binding to the backend XE Service Brokers. A service instance creation would involve creation of a limited user profile on one of the underlying oracle xe instances chosen at random by Subway Service Broker. Subway would attempt to delete the service instance by checking against all of the managed service brokers till it succeeds.

Oracle XE Tile

A Pivotal Tile would wrap the Bosh release composed of the Oracle XE instance and the packaging/deployment of the Service Broker. The Tile would install the combined Subway + Oracle XE Service Brokers as applications on Cloud Foundry and register the Subway as the main Service Broker with CF, while exposing its internal Oracle XE service broker service plans on the Marketplace. Users can choose to go with a single or multiple instance of Oracle XE by configuring the instance count in the Resource Config page.

Figure 1: Components of Oracle XE PCF Tile

Cloud Foundry

ORACLE XE TILE

SERVICES MARKETPLACE

Cloud Controller

5

4

Subway Service Broker

Client App

3

7

Internal Oracle SB1

Internal Oracle SB2

Internal Oracle SB3

2

VM 1

Oracle XE

(BOSH Managed)

VM 2

Oracle XE 2

(BOSH Managed)

VM 3

Oracle XE 3

(BOSH Managed)

1

8

Flow:
1) Deploy multiple Oracle XE instances on separate VM (by Tile)
2) Deploy an internal Oracle Service Broker as an App to CF with reference to specific Oracle XE instance (by Tile)
3) Deploy Subway Service Broker multiplexing connections to internal Oracle SBs
4) Register Subway Service Broker with CF (by Tile)
5) Publicize service plans on Marketplace (by Tile)
6) Service creation on one of the managed oracle-xe db
7) App Binds to Oracle-XE Service
8) Credentials passed to Client app
9) Unbind of service
10) Deletion of service
11) Scale up/down of oracle-xe instances from Ops Mgr

What are the limitations?

- The Oracle XE Database instances are meant for pure development and there is no associated backup, restore or management tools provided. The tile is an experimental one and not supported officially and also subject to change.
- The tile does not bundle the Oracle XE install and users are expected to make it available on some network endpoint and let Bosh download and install it as part of installation of the tile.
- This has been tested against only Oracle XE 11.2 RPM binary install only (made available on Oracle website as the only linux distro flavor). The tile requires Centos-7 Stemcell (available at http://bosh.io/stemcells).