



以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

# Oracle Developer Cloud Service キーワード一覧

## コード開発

- (privateの) Git リポジトリ
- マージ・リクエスト
- IDEから簡単接続
- GitHub連携
- Oracle OTN Mavenリポジトリ

## ビルド (Hudson)

- シェル
- Maven
- Ant
- Gradle
- Selenium
- (privateの) Maven リポジトリ

## 実行環境連携

- Java Cloud Serviceへデプロイ
- Application Container Cloudへデプロイ(予定)

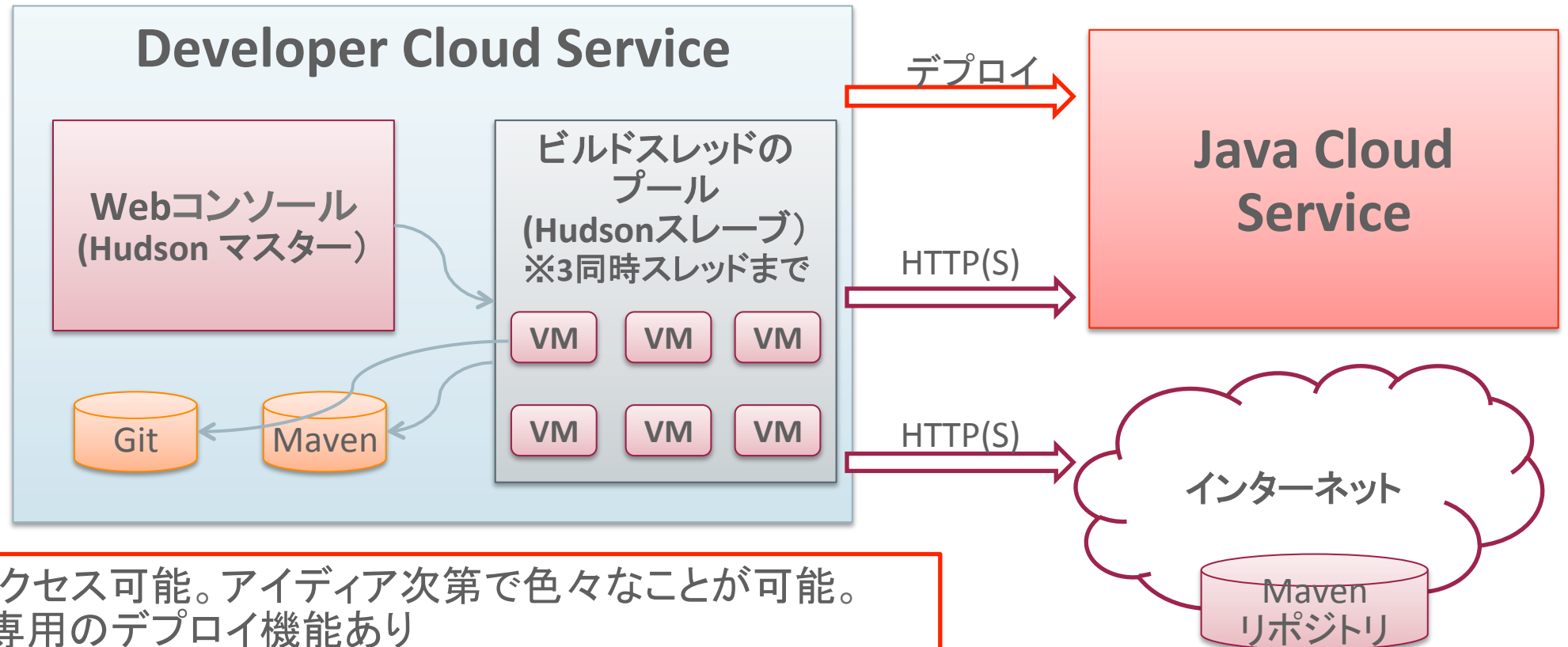
## 便利機能

- Webhooks
- メール通知
- デフォルト・コード・レビューア

## プロジェクトツール

- プロジェクト & メンバー登録
- Wiki
- Issues

# Developer CloudのHudsonビルドのアーキテクチャ



- インターネットアクセス可能。アイデア次第で色々なことが可能。
- Java Cloudへは専用のデプロイ機能あり
- Java CloudへはHTTP(S)が通るのでJava Cloud 上のアプリケーションのテストが可能

# Developer Cloudを活用したJava Cloud Serviceの使い倒しテクニック

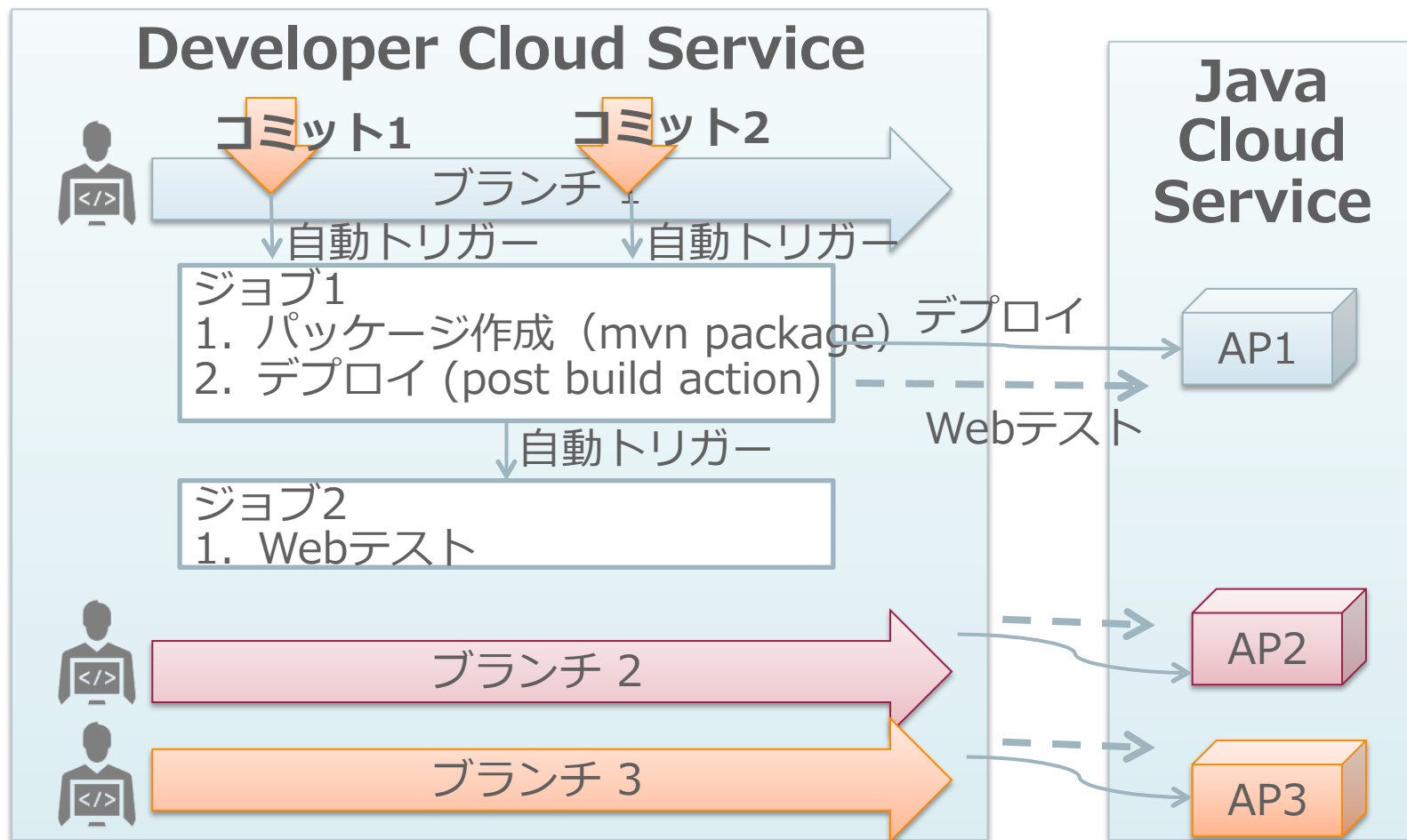
1. 継続的デリバリーとテスト自動化で環境をフル稼働
2. マルチテナントでクラウド実行環境をシェアして使う
3. 開発でJava Flight Recorderをフル活用する方法

# 1. 継続的デリバリーとテスト自動化で 環境をフル稼働

# Developer Cloud Serviceで行う継続的デリバリー コミットした後は自動的にデプロイされた環境での自動テストまで実行

## ポイント

- コミット後はJava Cloudへのデプロイとテストまで自動化可能
- Gitリポジトリ数に制限がないので、アプリケーションのラインは細かく分けられる
- Web上のコードレビューとマージ機能でブランチ開発を促進



# Developer Cloud Serviceのデプロイ機能

## Java Cloud Serviceにデプロイ

- デプロイ設定を作成（複数可）
  - デプロイ名とアプリケーション名
  - デプロイ先のJava Cloud Serviceの管理サーバーのアドレスとポート
  - タイプ：自動デプロイもしくは手動か
  - ジョブ名と手動デプロイの場合はビルド番号
  - ジョブで生成されたアーチファクト
- デプロイアクションでデプロイ実行
  - デプロイ／再デプロイ／アンデプロイ
  - アプリケーションのスタート／ストップ
- ジョブのPost-Build Actionsにデプロイの実行を組み入れ可能

The screenshot displays the 'Deploy' tab of the Developer Cloud Service interface. It features a navigation bar with links for Home, Code, Merge Requests, Issues, Build, Deploy, Wiki, and Administration. A 'New Configuration' button is located in the top right corner. The main content area is titled 'Deployments' and lists three deployment entries:

- WelcomeApp**: Java Service dcsdevteam / jcs2. Configuration WelcomeApp. Job / Build WelcomeApp / On Demand. Artifact ViewController/ deploy/webapp1.war. Last deployment succeeded -- January 14 2015 2:04 AM +0530.
- helloworld**: Java Service dcsdevteam / jcs2. Configuration helloworld. Job / Build hw / On Demand. Artifact target/helloworld.war. Last deployment succeeded -- Wednesday at 11:16 AM +0530.
- WelcomeApp2**: Java Service 129...75 / 9001 / weblogic. Configuration WelcomeApp2. Job / Build WelcomeApp / On Demand. Artifact ViewController/ deploy/webapp1.war. Last deployment failed -- Thursday at 2:26 PM +0530.

On the right side, the 'WelcomeApp: History' section shows a list of deployment events:

- Deployment Succeeded**: WelcomeApp / 4 / ViewController/ deploy/webapp1.war. Logs virus-scan, whitelist, wls-appc, cloud-appc, deploy. January 14 2015 2:04 AM +0530 by Alex Admin.
- Undeploy Succeeded**: WelcomeApp / 4 / ViewController/ deploy/webapp1.war. Logs undeploy. January 14 2015 2:02 AM +0530 by Alex Admin.
- Deployment Succeeded**: WelcomeApp / 4 / ViewController/ deploy/webapp1.war. Logs virus-scan, whitelist, wls-appc, cloud-appc, deploy. January 13 2015 9:43 PM +0530 by Alex Admin.
- Undeploy Succeeded**: WelcomeApp / 4 / ViewController/ deploy/webapp1.war. Logs undeploy. January 13 2015 9:42 PM +0530 by Alex Admin.



# マージ・リクエスト コードレビュー機能

- レビュー
  - ブランチに対してのレビュー
  - Developer Cloud Serviceプロジェクト内のメンバーがレビューアを複数指定
  - ソースコード上にレビューコメントを挿入できる
  - レビューの中でコメントを書いて会話をシステム上で展開
- マージ
  - Webのボタンからトピックブランチをターゲットブランチにマージ
- メール通知をサポート
  - リクエストの作成、承認、却下、クローズ、キャンセル、コメント追加時に通知

## リクエスト一覧

ID	Summary	Status	Repository	Branch	Submitter	Created
22	add dynamic support	OPEN	mywebapplication.git	dynamic-web	Clara Coder	22. 1. 2015
1	<a href="#">initial push to test-branch</a>	OPEN	mywebapplication.git	test-branch	Clara Coder	19. 1. 2015

Page 1 of 1 (1-2 of 2 items) < > 1 >

## コードレビュー

< Merge Requests | [initial push to test-branch](#)

OPEN Clara Coder wants to merge 3+ commits to master from test-branch

Conversation Commits (3+) **Changed Files (21)** Pending Comments (1) +235 -0

```
+12 .classpath MyDynamicApp
+12 .classpath MyDynamicApp
@@ -1,0 +1,12 @@
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
  <classpathentry kind="src" path="src"/>
  <classpathentry kind="con" path="org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.debug.ui.la
  <attributes>
    <attribute name="owner.project.facets" value="java"/>
  </attributes>
</classpath>
<classpathentry kind="con" path="org.eclipse.jst.j2ee.internal.web.container"/>
<classpathentry kind="con" path="org.eclipse.jst.j2ee.internal.module.container"/>
<classpathentry kind="output" path="build/classes"/>
</classpath>
```

Clara Coder - Just now  
need different value

Reviewers (2)  
NO RESPONSE  
Clara Coder  
Don Developer

コード上に  
レビューコメントを追加

# Developer Cloudでもできる自動テストテクニック① Selenium WebDriverを利用

- Selenium IDEを使って記録したWebDriverテストスクリプト
- Developer CloudはXvfbをサポート。GUIなしのFirefoxのSelenium WebDriverテストが可能

**Build Environment**

Start Xvfb before the build, and shut it down after

Xvfb Specific Displayname  ▼ ▲

Timeout In Seconds  ▼ ▲

Xvfb Screen

Xvfb Display Name Offset  ▼ ▲

Xvfb Additional Options

Log Xvfb output

Shutdown Xvfb with whole job, not just with the main build action

< [Jobs Overview](#) [easywebapp\\_selenium](#) #13 [Tests](#) [EasywebappTest](#)

### Test Results

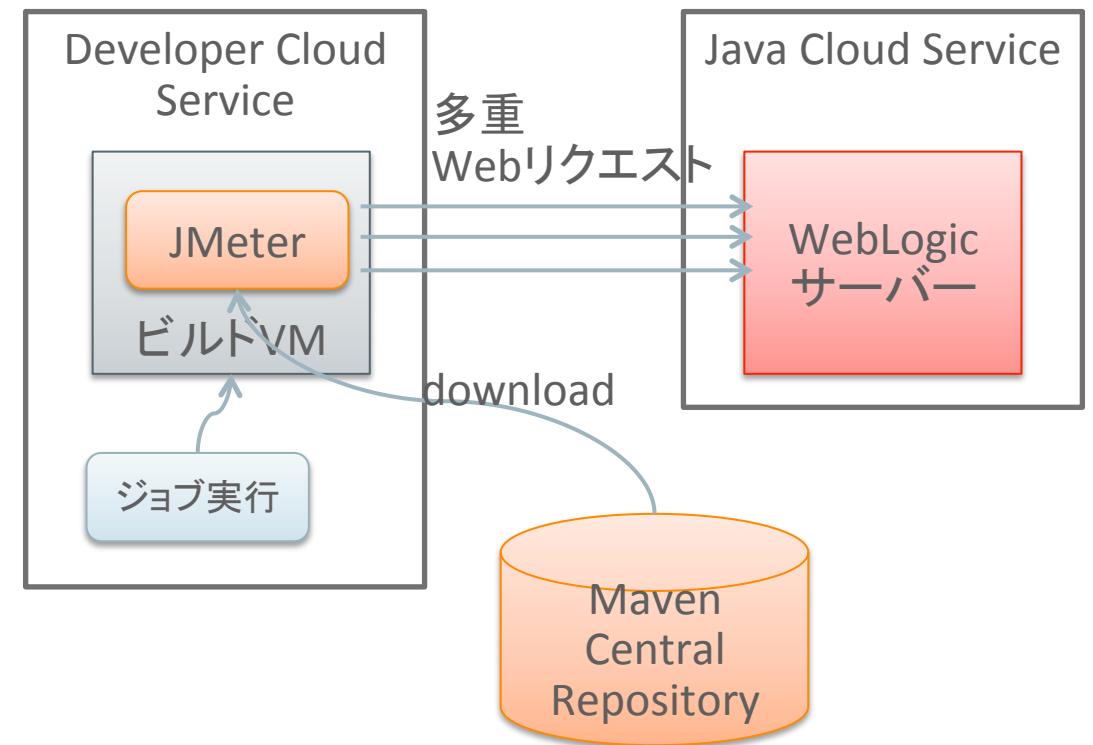
▲ Suite Name  
integrationtest.easywebapp.EasywebappTest

Test Name	Duration	Status
integrationtest.easywebapp.EasywebappTest.testFeatureC	6 s 787 ms	FAILED
integrationtest.easywebapp.EasywebappTest.testHelloWorld	3 s 683 ms	FIXED

# Developer Cloudでもできる自動テストテクニック②

## jmeter-maven-plugin を利用

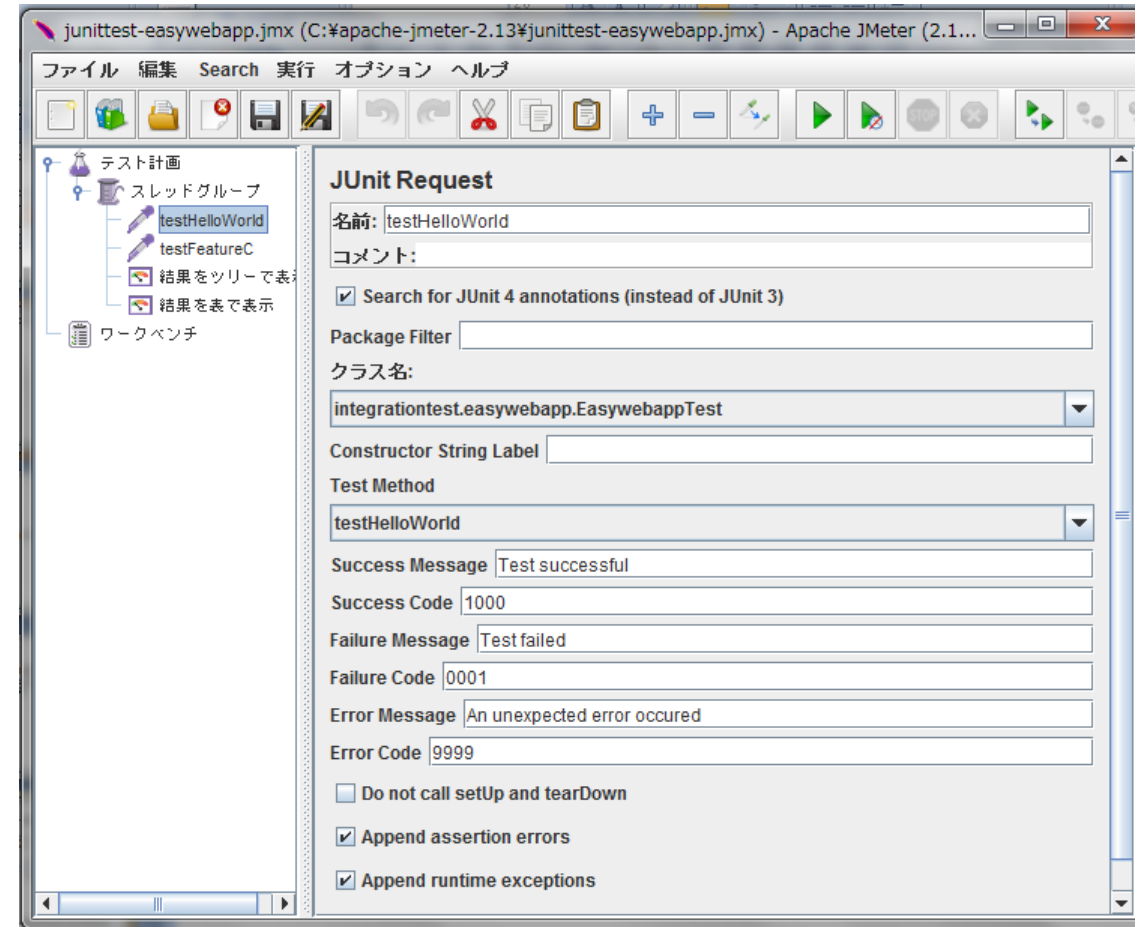
- Apache JMeterを使ってJava Cloud Serviceに簡易な負荷テストを実行
- Developer Cloud Serviceにテストをジョブとして登録し、いつでもテストが実施可能に
- jmeter-maven-plugin (<https://github.com/jmeter-maven-plugin/jmeter-maven-plugin>)を利用すればMavenから簡単にJMeterのインストールと実行が可能
- JMeterの結果ファイルをアーチファクト保存して結果を参照



# Developer Cloudでもできる自動テストテクニック③

## JMeter + Selenium

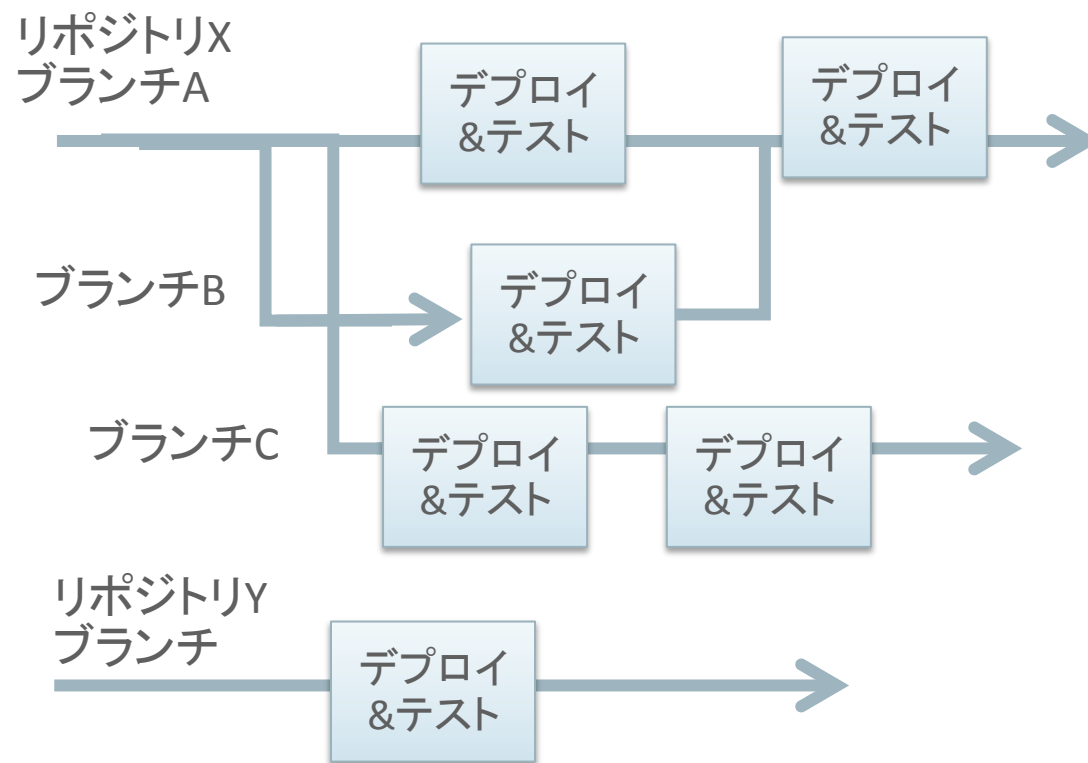
- Seleniumテストで多重度テストを試みたい場合に利用
- JMeterのJUnitサンプラーを利用



## 2. マルチテナントでクラウド実行環境 をチーム内でシェア

# クラウド開発はマルチテナント！

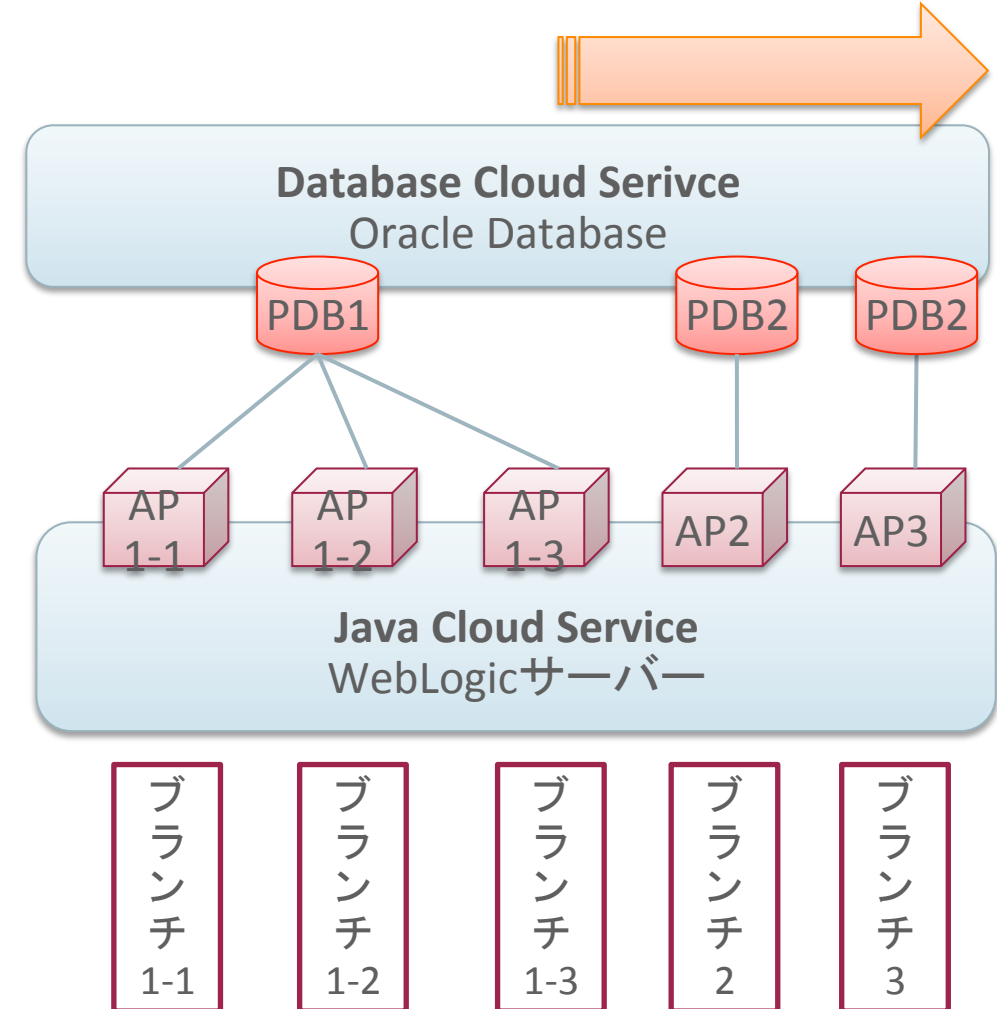
- コードのリポジトリやブランチが多くなると実行環境も増やしたくなる（＝テナント化）。
- テナント毎にデータや設定が独立できることが理想。ただ、一般的には管理が増えて増やすことができない。
- テナント毎にサーバーを用意すると課金コストが高つく。
- オラクル製品のマルチテナント機能とオラクルクラウドの拡張性でソリューションを見出してみます。



# アプリケーション層とデータベース層のマルチテナントの実現方法

- Database CloudはDB12cのマルチテナント機能を利用するのがベスト
  - ※High Performance以上のエディションが必要
  - Database Cloud付属のWebコンソールでPDBが誰でも簡単に追加・削除・クローン可能
- Java Cloudの現状バージョン (WebLogic サーバー12.1.3)のマルチテナント化の方法の一つはコンテキストルートに分ける方法
  - ※ちなみに最新WebLogicサーバー12.2.1ではマルチテナントが機能として登場！
- APとPDBの追加・削除に加え、マッピング（接続設定）の管理をいかに簡単にできるかがカギ

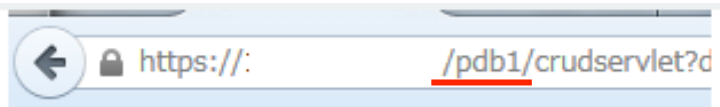
テナントが増えたら環境を拡張可能



# マルチテナントのイメージ

- URLのルートパスでテナントを分ける

## ブラウザ



successfully selected table mytable  
data for pdb1



successfully selected table mytable  
data for pdb2



successfully selected table mytable  
data for pdb3



## Webアプリケーション

デプロイメント

インストール 更新 削除 起動▼ 停止▼

<input type="checkbox"/>	名前
<input type="checkbox"/>	coherence-transaction-rar
<input type="checkbox"/>	+ DMS Application (12.1.3.0.0)
<input type="checkbox"/>	+ em
<input type="checkbox"/>	+ OraJaaSmon
<input type="checkbox"/>	+ <b>pdb1</b>
<input type="checkbox"/>	+ <b>pdb2</b>
<input type="checkbox"/>	+ <b>pdb3</b>



## データベース

```
SQL> show pdbs
```

```
CON_ID CON_NAME
```

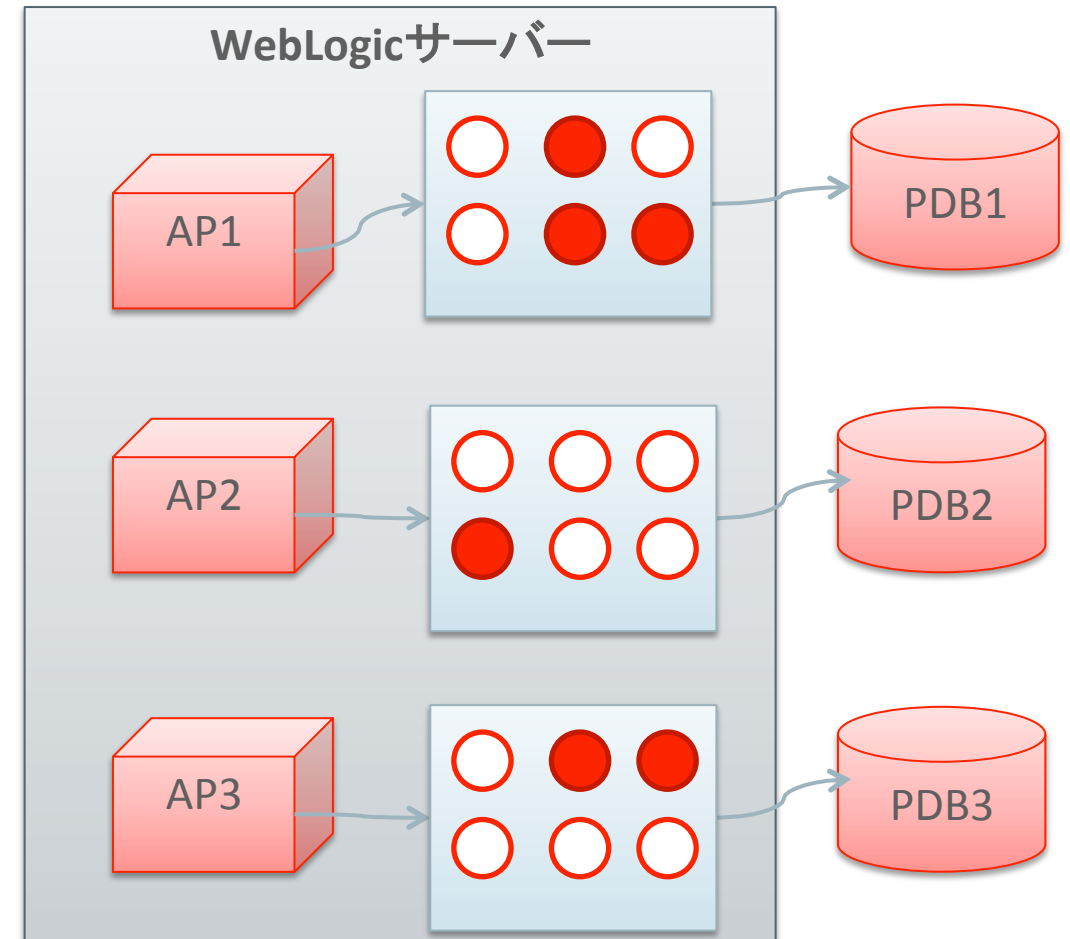
```
-----  
2 PDB$SEED  
3 PDB1  
4 PDB2  
5 PDB3
```



# WebLogicサーバーのPDB接続方法

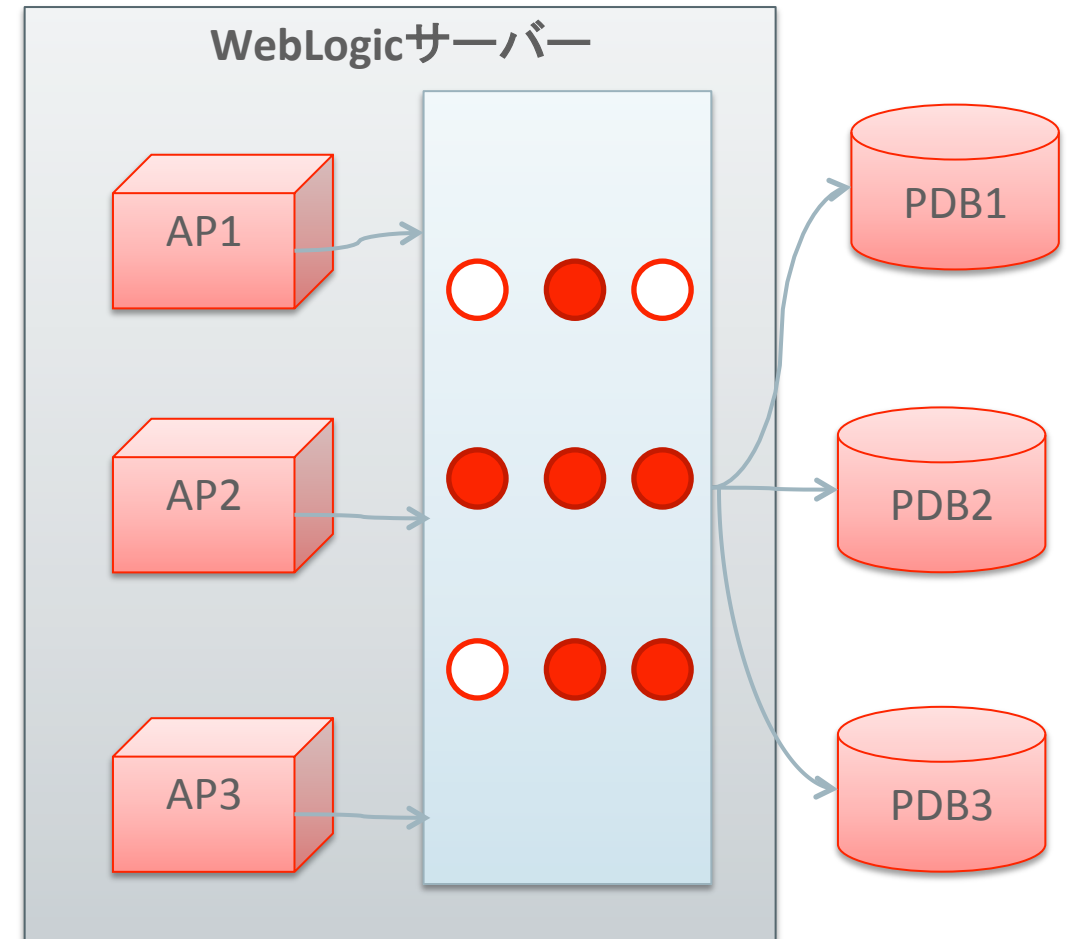
## パターン①：PDB毎にJDBCデータソースを設定

- アプリケーションが増えるにつれ、データソースの数も増えて、使用メモリが増加してしまう
- データソース内の接続は空いていることが多い。
  - それぞれの接続数を個別にチューニングするのは手間がかかる。



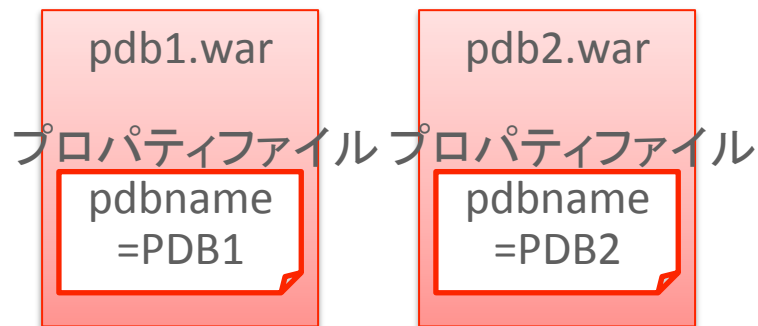
# WebLogicサーバーのPDB接続方法 パターン②：単一のデータソースから複数のPDBにアクセス

- データソースは一つで済む
- 接続数のチューニングも一か所で済む
- WebLogicとUCPの機能を利用したコネクション・ラベリング・コールバックを利用して実現



# UCPコネクション・ラベリング・コールバックの設定方法

- コネクション・ラベリング・コールバックのクラスのjarファイルをWebLogicサーバーの\$DOMAIN\_HOME/libディレクトリーに配置
- データソースを作成
  - 接続設定のユーザーはCDB共通ユーザーを指定
    - CREATE USER c##test\_user1 IDENTIFIED BY xxxxxx CONTAINER=ALL;
  - データソースに対してコールバック関数を登録
- アプリケーション内のプロパティファイルに使用したいPDB名を設定**



```
import oracle.ucp.jdbc.ConnectionLabelingCallback;
import oracle.ucp.jdbc.LabelableConnection;

public class PDBSwitchLabelingCallback implements
ConnectionLabelingCallback {

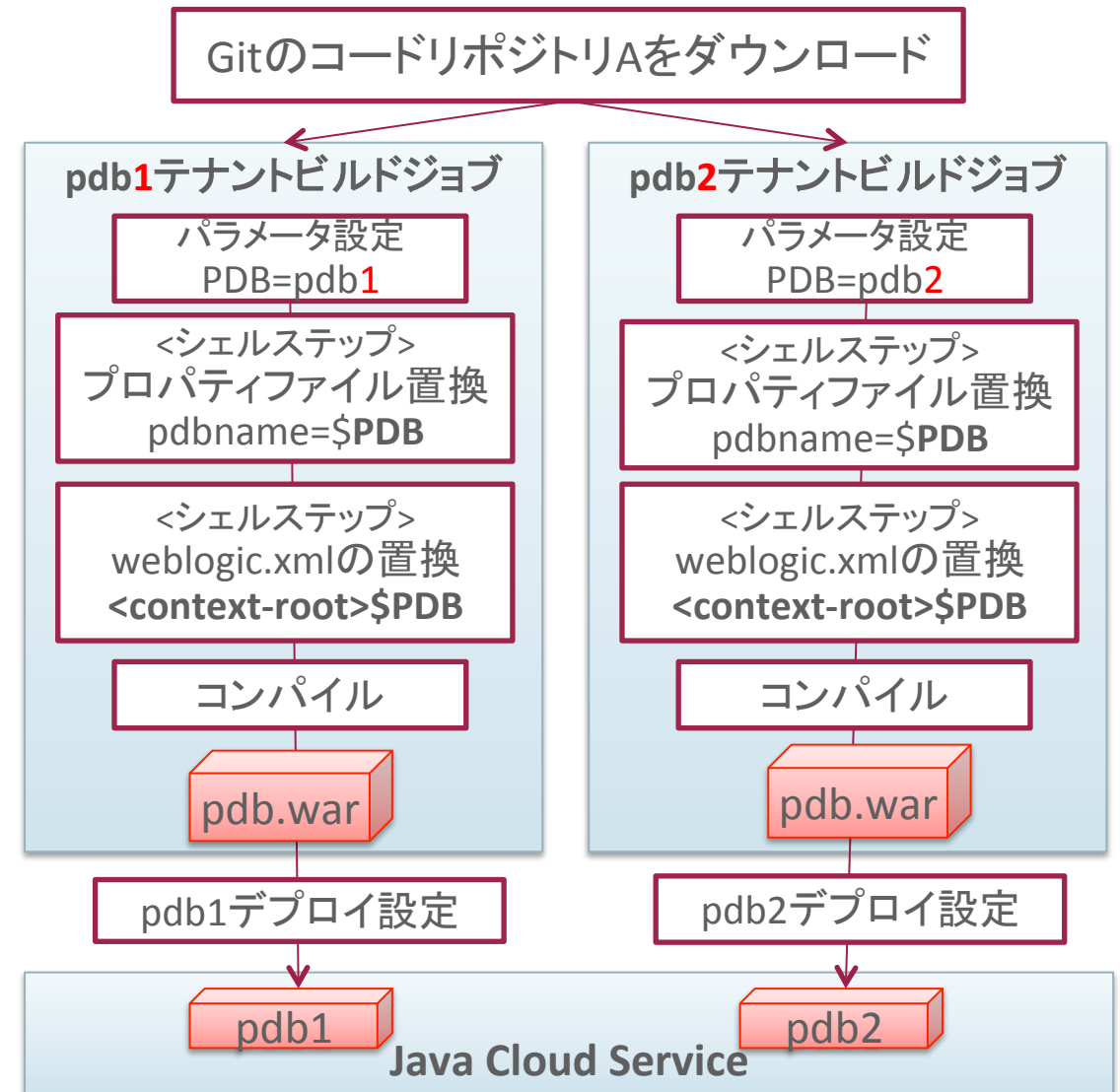
    public boolean configure(Properties reqLabels, Object conn) {
        String valStr = (String) reqLabels.get("pdbname");
        Statement s = ((Connection) conn).createStatement();
        s.executeUpdate("ALTER SESSION SET CONTAINER = " + valStr);
        LabelableConnection lconn = (LabelableConnection) conn;
        lconn.applyConnectionLabel("pdbname", valStr);
        .....
    }

    public Properties getRequestedLabels() {
        Properties labels = new Properties();
        Properties appProp = new Properties();
        appProp.load(Thread.currentThread().getContextClassLoader()
            .getResourceAsStream("app.properties"));
        labels.put("pdbname", appProp.get("pdbname"));
        return labels;
    }
}
```

※紙面の都合でコードは簡略化されています

# Developer Cloudのビルドでアプリケーションのマルチテナント化を楽にする

- テナント毎にビルドジョブを作成
  - 設定を共通化しておけばジョブコピーで簡単に増加可能。
- テナントを分けるための設定箇所
  - weblogic.xmlのcontext-root値でURLのルートパスを設定
  - プロパティファイル内の接続先PDB名
  - デプロイするときのアプリケーション名 (名前は分ける必要あり)



### 3. 開発でJava Flight Recorder(JFR)をフル活用する方法

# 開発で役立つJava Flight Recorder

- Java Flight Recorderで可視化できること（例）
  - 個々のWebリクエストの実行時間
  - 個々のリクエスト中のSQL文と実行時間
  - 時間のかかったJavaメソッド

※Java Flight RecorderのWebLogic Server Plugin (Unsupported)を利用した場合

※Java Flight Recorderの利用にはJava Cloud ServiceのエディションがEnterprise Edition以上である必要があります。

URI別のサーブレット呼出し数

列のフィルタ URI

URI	サンプル・カウ...	平均継続時間	プレッシャー
/pdb1/crudservlet	3	444 ms 956 µs	30.87%
/bea_wls_deployment_internal/DeploymentService	211	5 ms 939 µs	28.98%
/pdb3/crudservlet	3	370 ms 434 µs	25.70%
/pdb2/crudservlet	2	296 ms 910 µs	13.73%
/bea_wls_cluster_internal/psquare/p2	1	11 ms 963 µs	0.28%

ユーザー コネクタ データベース デバッグ デプロイメント EJB JMS JTA サーブレット

## キャプチャされたサーブレットリクエストの統計

SQL文

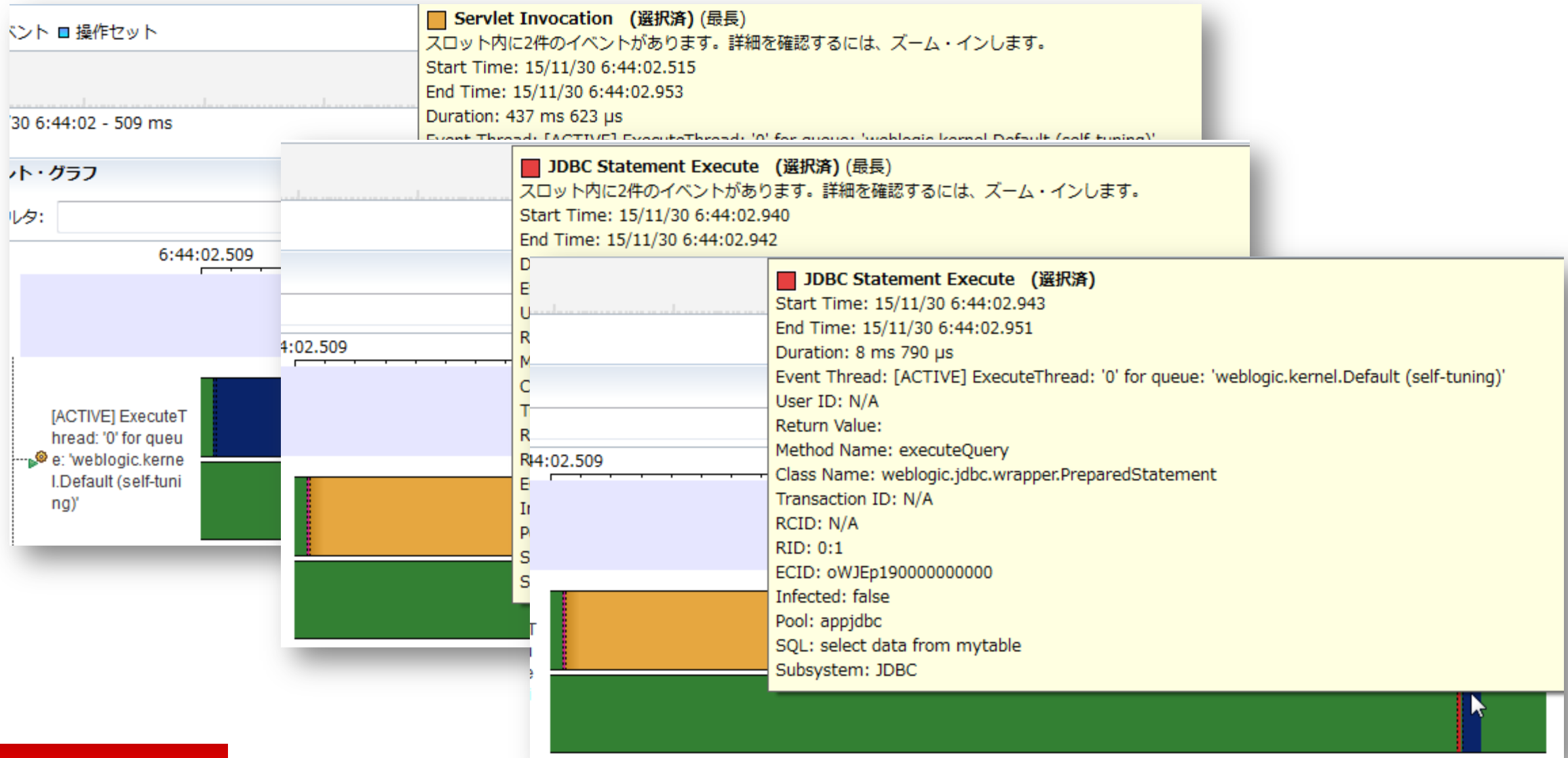
列のフィルタ SQL

SQL	平均継続時間	サンプル・...	プレッシャー
SQL SELECT /*+ INDEX(p MDS_PATHS_N7) USE_NL (p t) */ PATH_...	1 ms 410 µs	248	13.03%
SQL SELECT MAX(TXN_CN) FROM MDS_TRANSACTIONS WHERE TXN_...	1 ms 512 µs	213	12.01%
SQL drop table mytable	307 ms 7...	1	11.47%
SQL SELECT ENTRYID, PARENTDN, RDN FROM JPS_DN WHERE ((PARE...	2 ms 508 µs	96	8.97%
SQL create table mytable ( data varchar2(100) )	101 ms 1...	2	7.54%

ユーザー コネクタ データベース デバッグ デプロイメント EJB JMS JTA サーブレット

## キャプチャされたSQL実行の統計

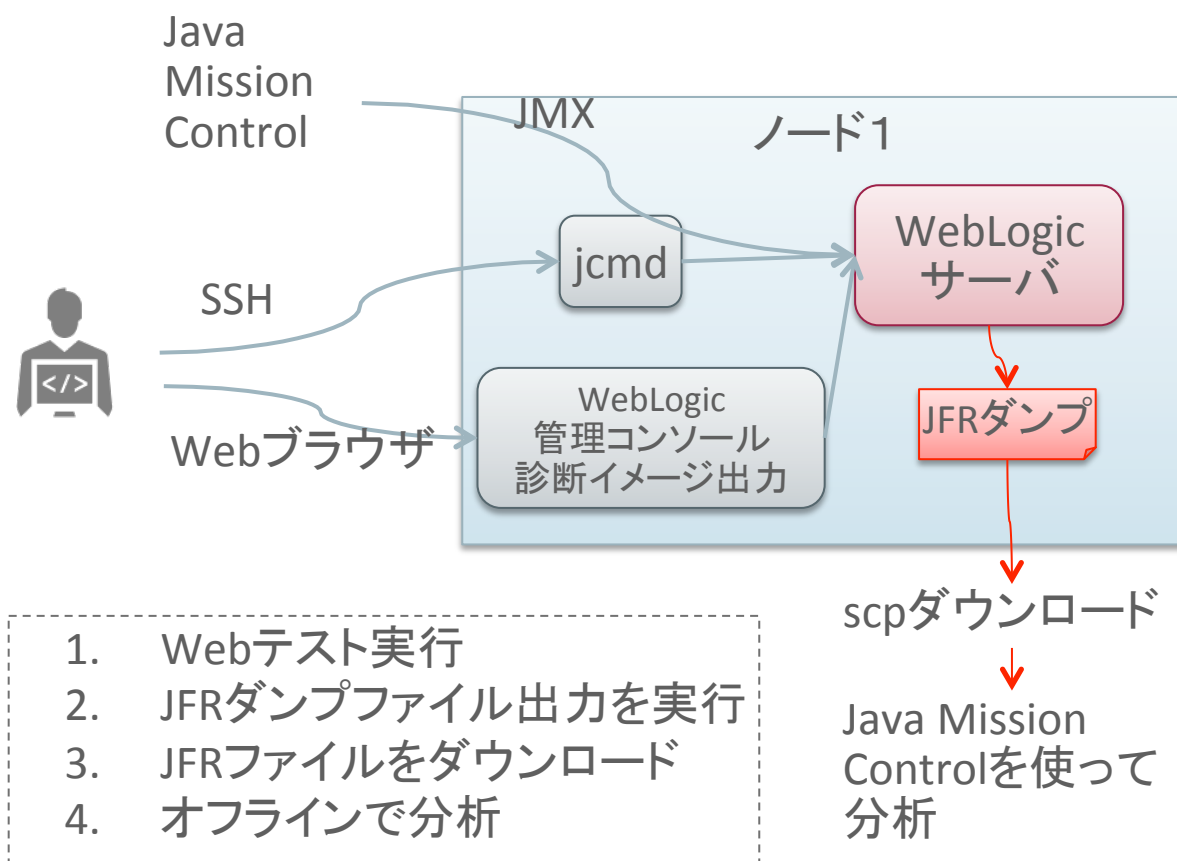
# リクエストの詳細を把握可能



# Java Flight Recorderを開発フローの一部に入れる際の課題

## • 課題

- JFRダンプファイルをサーバーから取得するには慣れが必要
- 慣れている場合でも、ダンプファイルを取得するのに数分かかる（障害解析のときには問題ないが、フローの一部として常時取得するには手間）
- サーバーが複数ある場合、それぞれから取得が必要な場合もある
- 自動テストの結果の一部にダンプファイルを残す方法がない



Java Flight Recorderのダンプ出力時のアクセス



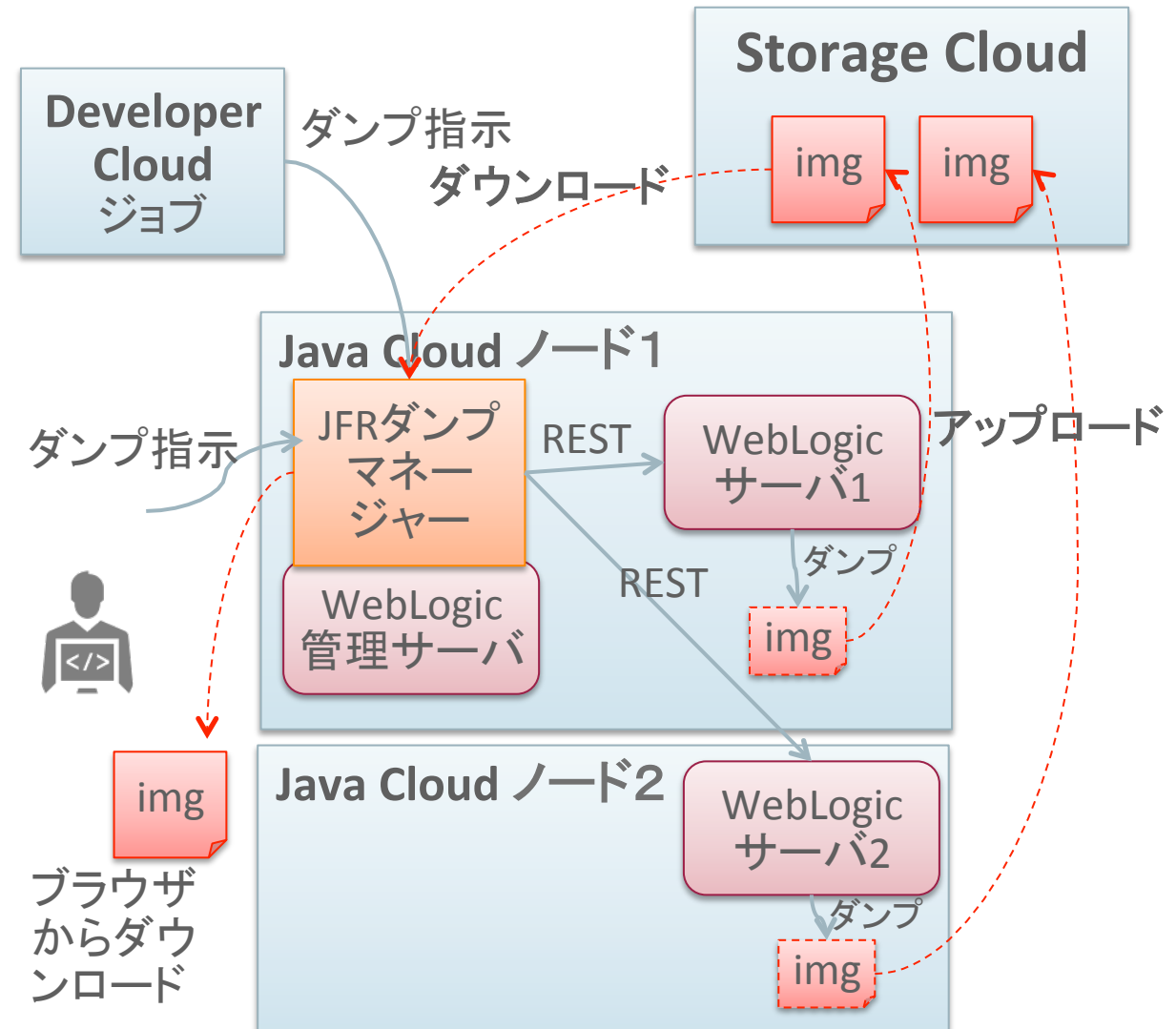
# JFRダンプマネージャー (自作実験ツール) + Oracle Storage Cloud Service

- 簡単・素早くダンプ実行
  - Webブラウザやcurlなどを利用
- Storage Cloudにファイル保存
  - ダンプ実行時に自動的にOracle Storage Cloudにダンプファイルが保存される
- ダウンロードも簡単
  - Storage CloudにあるファイルをJFRダンプマネージャー経由でWebダウンロード
  - ダンプリクエストのレスポンスにダウンロードのリンクが含まれ、自動テスト結果にリンクを残せられる

※Storage Cloudは従量課金で有料です

**GitHubで公開予定**

img ... WebLogic診断イメージファイル(JFRダンプを含む)





クラウド・テクノロジーを語ろう  
**Oracle Cloud Developers**

第1回 Meetup 2016年1月29日 19:00~  
@オラクル青山センター

参加登録はこちら : <http://ora.cl/91T>

```
var community = React.createClass({
  init : function(){
    return {
      date : "2016-1-29",
      location : "OAC",
      goal : [
        "Learn",
        "Connect",
        "Have Fun"
      ]
    };
  }
});
```

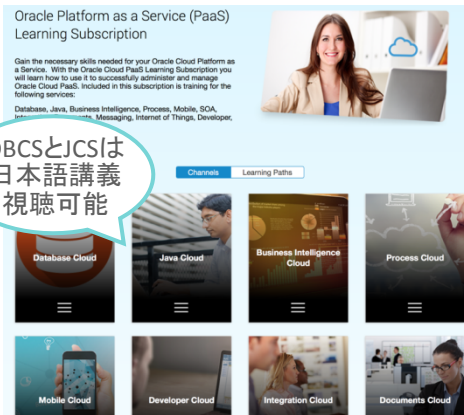


# Oracle Cloud ラーニング・サブスクリプション

Oracle Cloud (PaaS) の導入から運用まで、クラウドを活用するために必要なスキルを身につけた "Cloud Ready" なエンジニアを育成するための学習プラットフォーム

- Oracle Cloud Service の活用に必要なスキルを学習できるビデオ・トレーニング
- 製品アップデートに応じて最新のコンテンツに更新
- 1年間のサブスクリプション形式

## Oracle Platform as a Service ラーニング・サブスクリプション



多彩な Oracle PaaS の活用方法をトータルにカバー。すべての Oracle PaaS 技術者にオススメです！

### 学習内容:

- Oracle Cloud Platform as a Service の使用方法
- Oracle PaaS の多様なサービスを活用する利点
- Oracle PaaS の運用管理
- Oracle PaaS を使ったアプリケーションの開発とデプロイ
- 既存のアプリケーションの Oracle PaaS への移行 など

### 【対応サービス】

Database Cloud, Java Cloud, Business Intelligence Cloud, Process Cloud, Mobile Cloud, Integration Cloud, Documents Cloud, Messaging Cloud, Internet Of Things Cloud, SOA Cloud, Database Backup Cloud, Developer Cloud

定価: 116,856 円 (税込)

50% Off

特別価格: **58,428** 円 (税込)

1ユーザー/1年間利用可能

【ご注意】 Oracle Platform as a Service (PaaS) ラーニング・サブスクリプションの最小購入ユーザー数は 5 です。本特別価格は、2015 年 12 月 31 日までにご購入される方に対して適用されます。また、他の割引契約、またはキャンペーンと併用することはできません。

ただいま 無償体験版公開中！

Oracle Cloud インスタンスの作成やクラウド上の Oracle Database, WebLogic Server の起動方法など、技術者が円滑に Oracle Cloud (PaaS) をはじめるためのポイントを学習できる『Getting Started』を視聴可能

アクセスはこちらから

[education.oracle.co.jp/cls\\_paas](http://education.oracle.co.jp/cls_paas)

オラクルユニバーシティ  
お問い合わせ窓口



TEL 0120-155-092

URL <http://www.oracle.com/jp/education/>



# Integrated Cloud

## Applications & Platform Services

ORACLE®