Windows Azure で作る ボットアプリ？

http://twitter.com/normlian
http://d.hatena.co.jp/waritohutsu
http://www.pixiv.net/member.php?id=147209

自己紹介

- HN　　：割と普通
- 本拠地：横浜近辺
- 肩書き：しがない SI屋
- 趣味　：ソースを書く&絵を描く

絵描きがさぼり気味・・・orz

本セッションの趣旨

- 本セッションの対象者
  - ASP.NETは弄った事あるよ
  - Windows Azureってクラウドだよね？
  - Windows Azureと他のクラウドってどう違うの？

- 本セッションのゴール
  - これでWindows Azure上でボット開発できるぜ！
  - Windows Azureってこういうクラウドなのか…

アジェンダ

- Windows Azure と、他クラウドの違い
- Windows Azure とは何者か
- Windows Azure 上の開発
- Azure Storage を使ってみる
- Twitter ボットを作ってみる
- まとめ

他クラウドとの違い

- Google
  - すべてがクラウド側だけで完結する世界を目指している
  - 現在の技術レベル、利用者では、使い勝手やセキュリティ、障害対応などで問題有り
- Amazon
  - サーバイメージを配置する為、管理作業の多くをユーザーにゆだねる
  - 自由度が高い反面、管 理コストは大きな変化無し
- Microsoft
  - 既存資産流用を考慮にいれ、クラウドサービスという選択の自由を提供

アジェンダ

- Windows Azure と、他クラウドの違い
- Windows Azure とは何者か
- Windows Azure 上の開発
- Azure Storage を使ってみる
- Twitter ボットを作ってみる
- まとめ

Azure Services Platform

マイクロソフトのデータ センターでホストされる、インターネット規模のクラウド サービス プラットフォーム

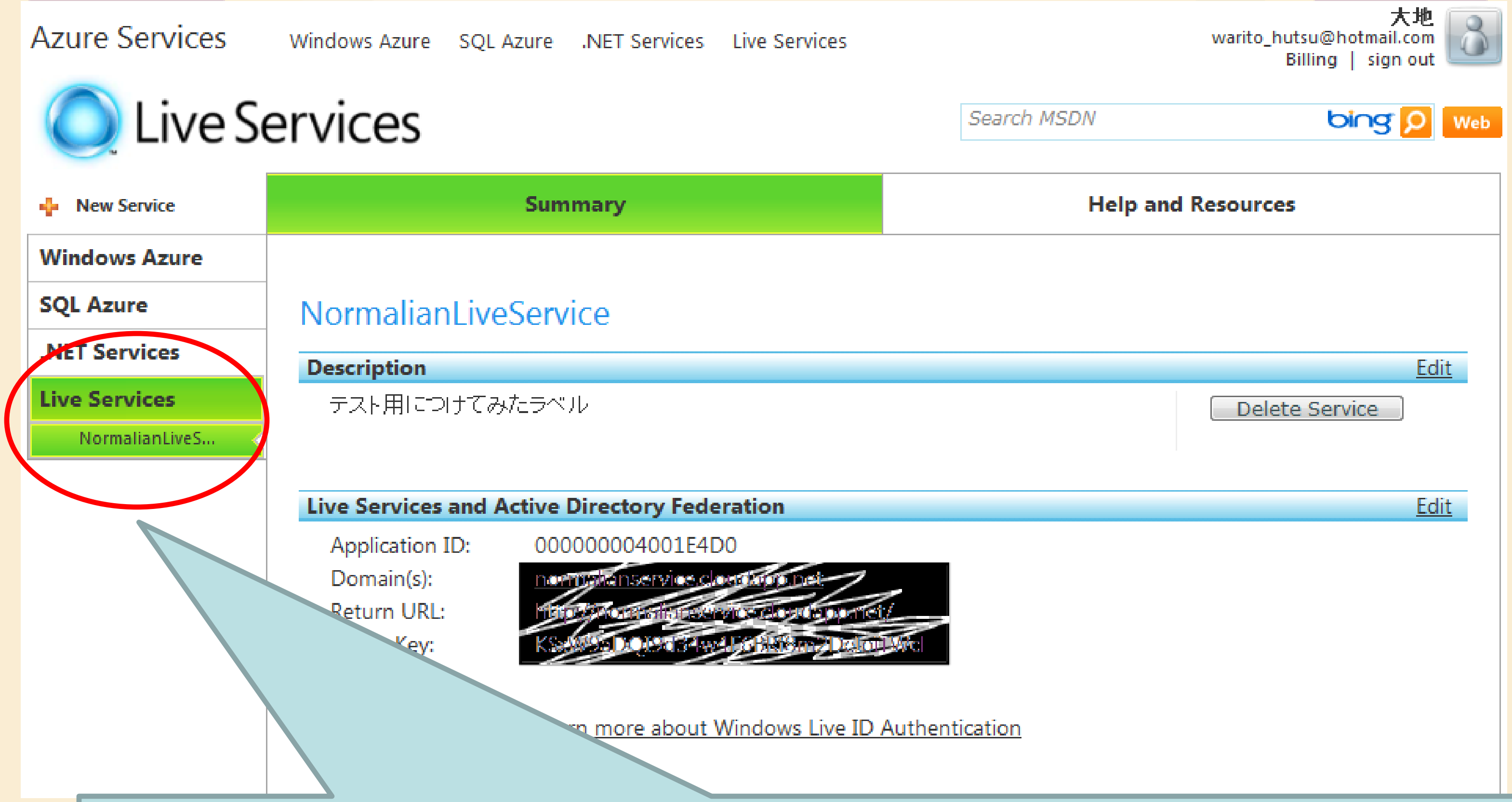
Azure Services Platform

- **Windows Azure** → 大本のシステム
- **.NET Services** → SOAっぽいもの
- **Live Services** → 認証っぽいものとか
- **SQL Azure** → SQL Server on クラウド

Azure Services Platform

- **Windows Azure**
  - 自動管理、ロール、ストレージ
- **.NET Services**
  - サービス×クライアント間、サービス×サービス間の通信を行うISB
- **Live Services（ポータルサイトから消えてる…）**
  - 認証、連絡先、デバイス
- **SQL Azure**
  - データベース、レポーティング等々・・・

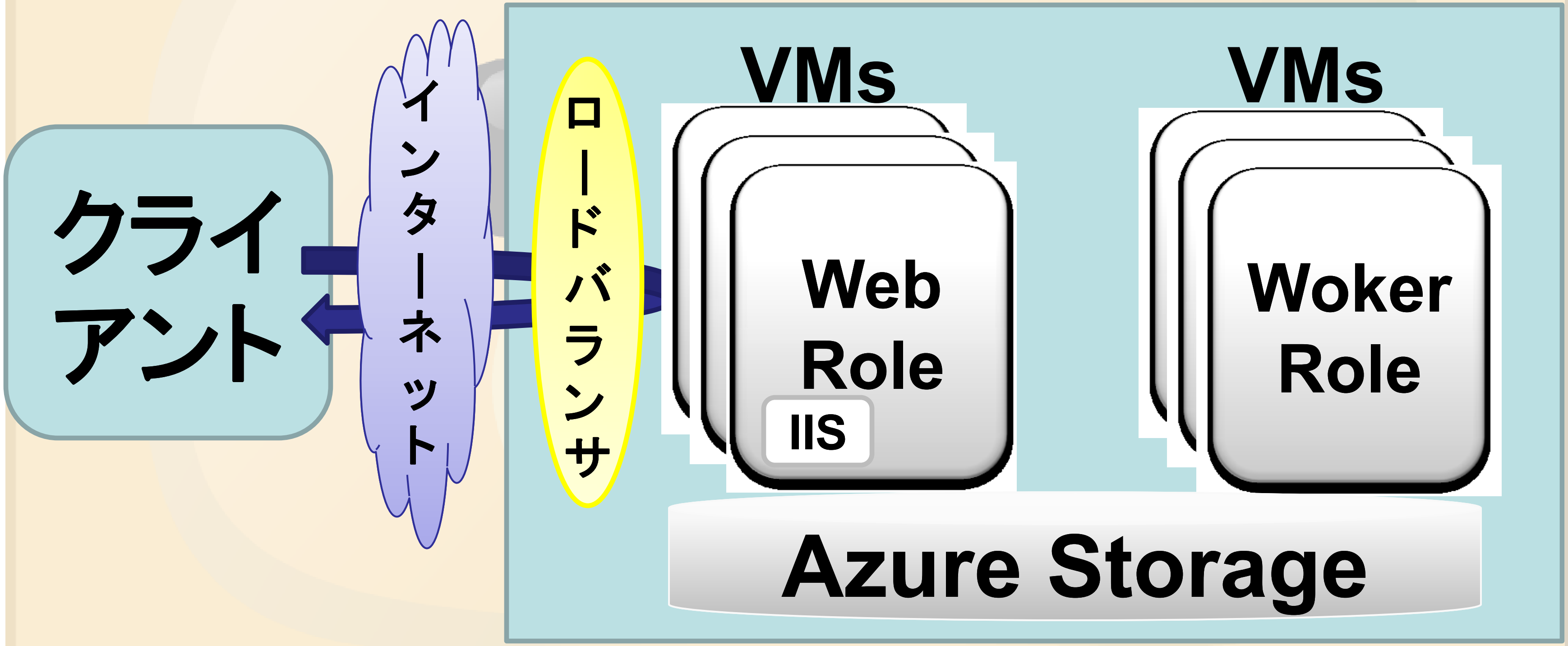
Azure Services Platform



方々なページを行ったり来たりすると、**Live Services**が表示されたりする

Windows Azure とは

- フロントがWeb Role、バックがWorker Roleとしたアーキテクチャ
- インスタンスは複数実行可能



Windows Azure とは

- 自動管理
  - ポータルサイトで ⇒ 実行制御の設定もできる
- ロール
  - Web Role ⇒ ほぼASP.NET
  - Worker Role ⇒ バックエンド処理等
- ストレージ
  - Blob ⇒ 大容量バイナリデータを扱う
  - Queue ⇒ Web RoleとWorker Roleの通信
  - Table ⇒ KeyValue Store

アジェンダ

- Windows Azure と、他クラウドの違い
- Windows Azure とは何者か
- Windows Azure 上の開発
- Azure Storage を使ってみる
- Twitter ボットを作ってみる
- まとめ

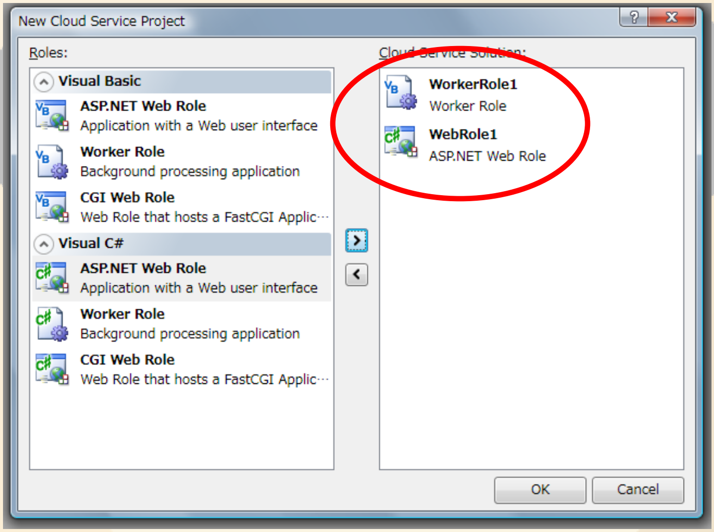
Windows Azureの開発で必要な物

- 必要なOS
  - Windows Vista (Home Premium以上)
  - Windows Server 2008
  - Windows 7
- 必要なVisual Studio
  - Visual Studio 2008 SP1
  - Visual Studio 2010 beta
- SQL Server 2005以上(SQL Azureに繋ぐ為には SQL Server 2008が必要)
- .NET Framework 3.5以上

この環境に
Windows Azure
Tools for
Microsoft
Visual Studio
をインストール

Windows Azureの開発テンプレート

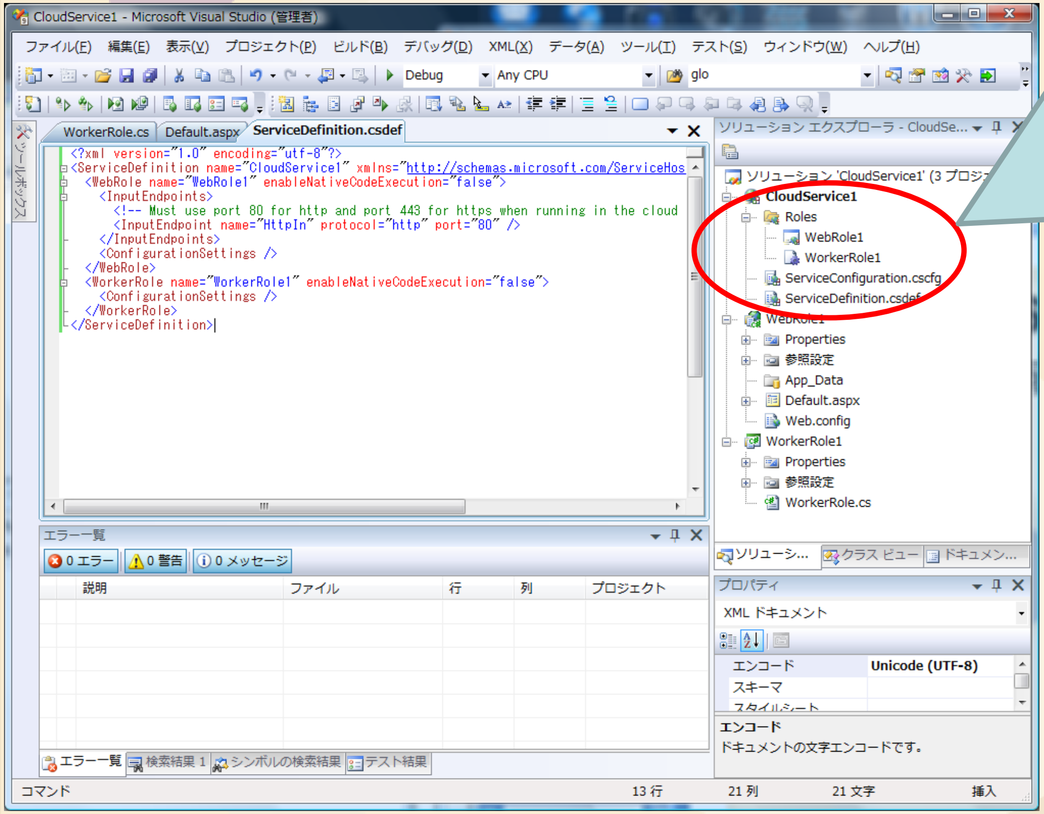
- とりあえずサービスを作ってみる
- [新規作成] → [プロジェクト] →[CloudService] 辺りを選択



上記のダイアログが表示され、開発テンプレートが選択可能（**C#＋VBが併存可能**）

Windows Azureの設定ファイル

- プロジェクト構成＋設定ファイル



- ここに登録したプロジェクトのインスタンスがAzure上に作られる
- 設定ファイル *.cscfgと、*.csdefが存在する
- *.csdef ⇒ キー定義用
- *.cscfg ⇒ 値定義用

Windows Azureの設定ファイル

*.csdefファイル

ServiceDefinition.csdef　ServiceConfiguration.cscfg

```xml
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="NormalianService" xmlns="http://schemas.microso
  <WebRole name="Gottani_WebRole" enableNativeCodeExecution="false">
    <InputEndpoints>
      <!-- Must use port 80 for http and port 443 for https when running
      <InputEndpoint name="HttpIn" protocol="http" port="80" />
    </InputEndpoints>
    <ConfigurationSettings>
      <Setting name="AccountName" />
      <Setting name="AccountSharedKey" />
      <Setting name="BlobStorageEndpoint" />
      <Setting name="QueueStorageEndpoint" />
      <Setting name="TableStorageEndpoint" />
    </ConfigurationSettings>
  </WebRole>
  <WebRole name="WebApplication1" enableNativeCodeExecution="false">
    <InputEndpoints>
      <!-- Must use port 80 for http and port 443 for https when running
      <InputEndpoint name="HttpIn" protocol="http" port="8080" />
    </InputEndpoints>
    <ConfigurationSettings />
  </WebRole>
</ServiceDefinition>
```

*.cscfgファイル

ServiceDefinition.csdef　ServiceConfiguration.cscfg

```xml
<?xml version="1.0"?>
<ServiceConfiguration serviceName="NormalianService" xmlns="http://schemas
  <Role name="Gottani_WebRole">
    <Instances count="1" />
    <ConfigurationSettings>
      <!--
      <Setting name="BlobStorageEndpoint" value="http://127.0.0.1:10000/"
      <Setting name="QueueStorageEndpoint" value="http://127.0.0.1:10001/"
      <Setting name="TableStorageEndpoint" value="http://127.0.0.1:10002/"
      <Setting name="AccountName" value="devstoreaccount1" />
      <Setting name="AccountSharedKey" value="Eby8vdM02xNOcqFlqUwJPLlmEtlC
      -->
      <Setting name="BlobStorageEndpoint" value="http://normalian.blob.cor
      <Setting name="QueueStorageEndpoint" value="http://normalian.queue.c
      <Setting name="TableStorageEndpoint" value="http://normalian.table.c
      <Setting name="AccountName" value="normalian" />
      <Setting name="AccountSharedKey" value="JsA6d8D2imguEWJOThIrPyvXvsl3
    </ConfigurationSettings>
  </Role>
  <Role name="WebApplication1">
    <Instances count="1" />
    <ConfigurationSettings />
  </Role>
</ServiceConfiguration>
```

Microsoft.ServiceHosting.ServiceRuntime.
RoleManager.GetConfigurationSetting(＜キー名＞)

コード上での値取得

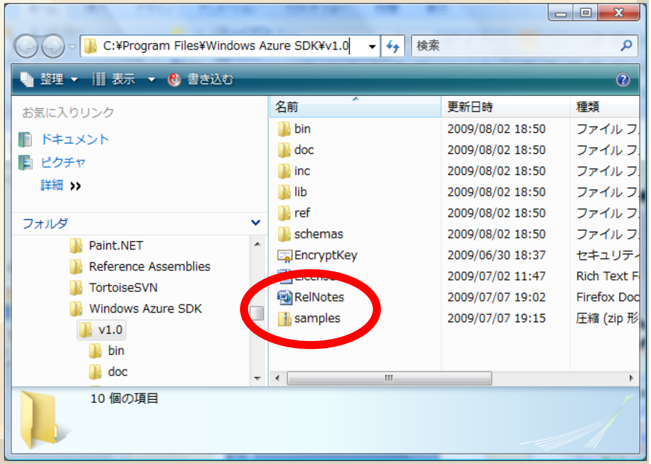
簡単な Hello World アプリのデモ

アジェンダ

- Windows Azure と、他クラウドの違い
- Windows Azure とは何者か
- Windows Azure 上の開発
- Azure Storage を使ってみる
- Twitter ボットを作ってみる
- まとめ

Azure Storageを弄る為に

- Azure Storageを弄るAPIはバイナリで提供されていない（Codeplexには存在する）
- Azure SDKの奥深くに存在するsample.zipの中身から「StorageClient」プロジェクトを引っこ抜く
  - Blob
  - Table
  - Queue



Tableのデモ

＋

Blobのデモ

Tableの開発ノウハウ

- Tableって名前ついてるけど、RDB連想禁止
  - 流行のKeyValueStoreです
  - スケールアウトはPartitionKey、RowKeyの設計次第
- DataContextやらEntityは自分でコーディング
  - 以下を継承したクラスをコーディング
    - Microsoft.Samples.ServiceHosting.StorageClient
      - TableStorageEntity（**Entity**）
      - TableStorageDataServiceContext（**DataContenxt**）
- 集計系のメソッドが存在しない(以下は全滅)
  - **Single、SingleOrDefault、OrderBy、OrderByDescending、Count、Min、Max、Average**

Windows Azureの開発ノウハウ

- まず、サンプルと日本語の解説書を読むこと
- ローカルストレージに繋ぐのでなく、本番ストレージにつないで開発しましょう
- Azure Storageの制御には「Azure Storage Explore」が便利(Codeplexで公開中)
- ローカル環境と本番環境のタイムゾーンの差に注意
- 本番環境にデプロイ＆実行後は、Runningメッセージが出るまで素直に待つ

アジェンダ

- Windows Azure と、他クラウドの違い
- Windows Azure とは何者か
- Windows Azure 上の開発
- Azure Storage を使ってみる
- Twitter ボットを作ってみる
- まとめ

Worker Roleを見ると・・・

```
using System;
using System.Collections.Generic;
using System.Threading;
using System.Linq;
using System.Text;
using Microsoft.ServiceHosting.ServiceRuntime;

namespace WorkerRole1
{
    public class WorkerRole : RoleEntryPoint
    {
        public override void Start()
        {
            // This is a sample worker implementation. Replace with your logic.
            RoleManager.WriteToLog("Information", "Worker Process entry point called");

            while (true)
            {
                Thread.Sleep(10000);
                RoleManager.WriteToLog("Information", "Working");
            }
        }

        public override RoleStatus GetHealthStatus()
        {
            // This is a sample worker implementation. Replace with your logic.
            return RoleStatus.Healthy;
        }
    }
}
```

これはどう見ても Twitterボット作れと言ってるだろjk

Twitter にどうやってポストする？

- LINQ to Twitterが存在する
  - http://www.codeplex.com/LinqToTwitter
  - 一通りのAPIは用意されてるので便利

```
var twitterContext = new TwitterContext(
        RoleManager.GetConfigurationSetting("bot_username"),
        RoleManager.GetConfigurationSetting("bot_password"));
twitterContext.UpdateStatus("発言するよー♪" + DateTime.Now);
```

これだけで Twitterにポストできる

Twitter ボット

デモ

Twitter ボット開発ノウハウ

- TwitterのAPI制限には注意する事
  - 同じ発言数回でも引っかかる
  - 一定時間内に発言しすぎても引っかかる
    - http://sourceforge.jp/projects/tween/wiki/API制限について
- DLLのローカルコピーはTrueにする事
  - LinqToTwitter.dllは本番環境にないので、ローカルコピーをTrueにする必要がある
  - 上記を設定しないとDr.Watsonと会えます

アジェンダ

- Windows Azure と、他クラウドの違い
- Windows Azure とは何者か
- Windows Azure 上の開発
- Azure Storage を使ってみる
- Twitter ボットを作ってみる
- まとめ

まとめ

- Windows Azureは面白いけど発展途上
- Windows Azureのコツは「自分を責めない」
- 2009/11 のPDCで公開される情報に注目
- Training Kit は最初に目を通しましょう
- 本番環境とローカル環境との違いに注意
- Codeplexの動向には注意しよう

参考

- 「.NET開発テクノロジ入門」
  - Bingって確認してください♪
- 「Windows Training KIT」
  - http://windows-azure.jp/community/TrainingKit.aspx
- 「Azureの鼓動」 砂金氏 ブログ
  - http://blogs.itmedia.co.jp/isago/
- 「Codeplex」公式で Azureと検索
  - http://www.codeplex.com/
- 非常に普通なブログをAzureで
  - http://d.hatena.ne.jp/waritohutsu/searchdiary?word=*[Windows%20Azure]