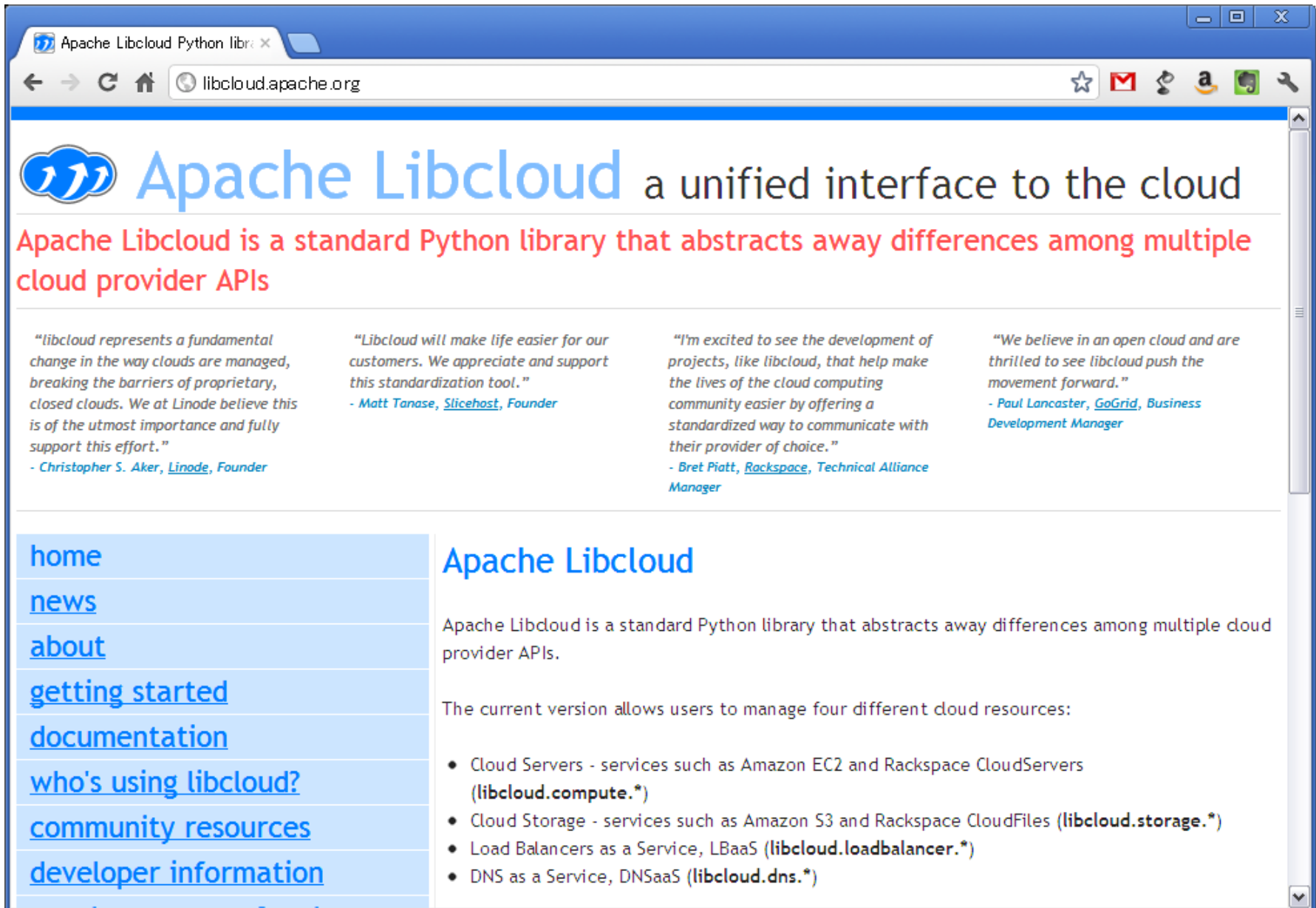


異なるベンダーの クラウドAPIを使ってみる

2012年4月20日

キヤノンソフト情報システム

上村準也



The screenshot shows a web browser window with the URL `libcloud.apache.org`. The page features the Apache Libcloud logo and the tagline "a unified interface to the cloud". A red headline states: "Apache Libcloud is a standard Python library that abstracts away differences among multiple cloud provider APIs". Below this, there are four testimonials from industry leaders. On the left, a navigation menu lists: home, news, about, getting started, documentation, who's using libcloud?, community resources, and developer information. The main content area includes the title "Apache Libcloud" and a brief description: "Apache Libcloud is a standard Python library that abstracts away differences among multiple cloud provider APIs." It also lists the current version's capabilities: "The current version allows users to manage four different cloud resources:" followed by a bulleted list: Cloud Servers (libcloud.compute.*), Cloud Storage (libcloud.storage.*), Load Balancers as a Service (libcloud.loadbalancer.*), and DNS as a Service (libcloud.dns.*).

Apache Libcloud Python libr x

libcloud.apache.org

Apache Libcloud a unified interface to the cloud

Apache Libcloud is a standard Python library that abstracts away differences among multiple cloud provider APIs

"libcloud represents a fundamental change in the way clouds are managed, breaking the barriers of proprietary, closed clouds. We at Linode believe this is of the utmost importance and fully support this effort."
- Christopher S. Aker, [Linode](#), Founder

"Libcloud will make life easier for our customers. We appreciate and support this standardization tool."
- Matt Tanase, [Slicehost](#), Founder

"I'm excited to see the development of projects, like libcloud, that help make the lives of the cloud computing community easier by offering a standardized way to communicate with their provider of choice."
- Bret Piatt, [Rackspace](#), Technical Alliance Manager

"We believe in an open cloud and are thrilled to see libcloud push the movement forward."
- Paul Lancaster, [GoGrid](#), Business Development Manager

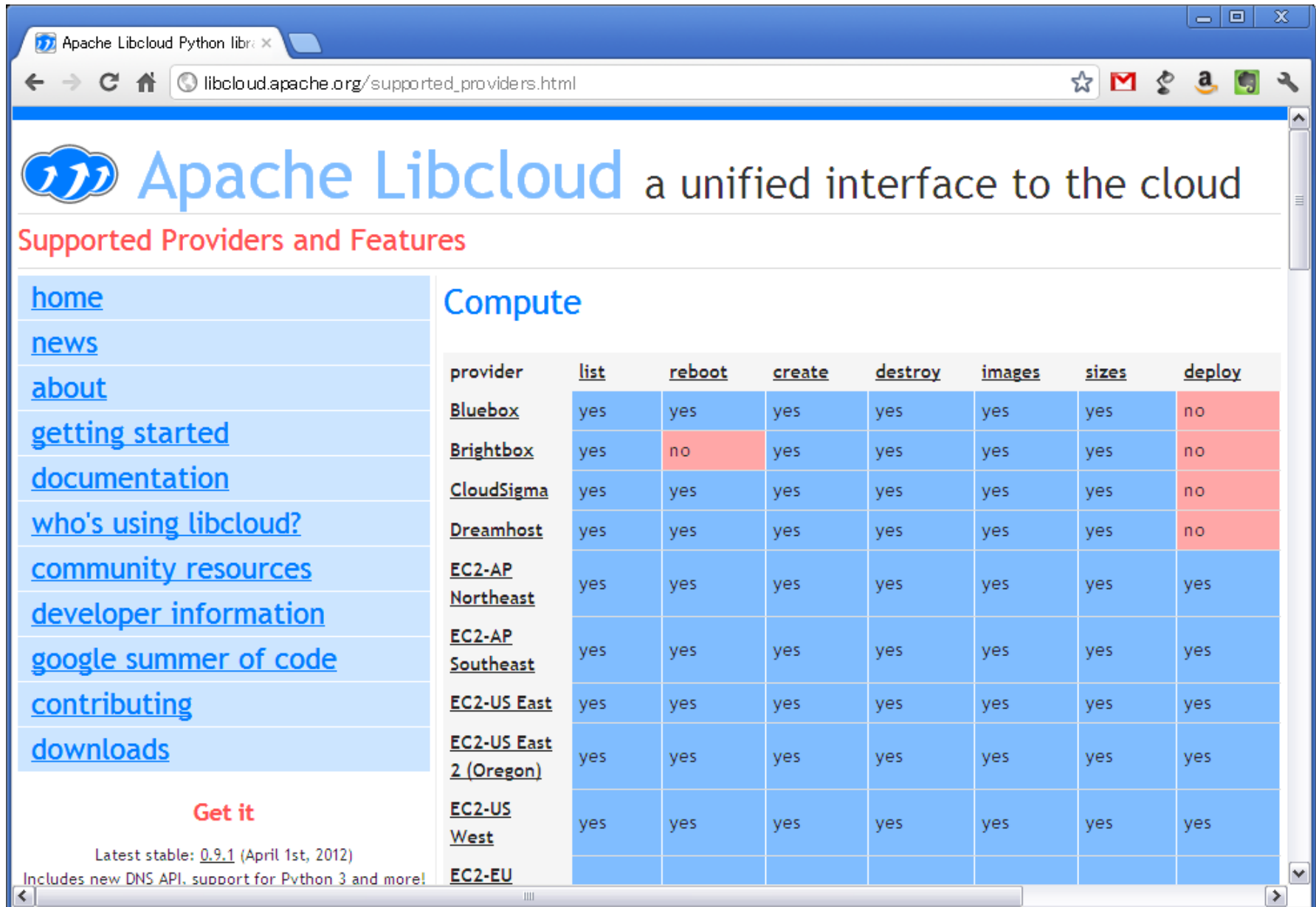
- [home](#)
- [news](#)
- [about](#)
- [getting started](#)
- [documentation](#)
- [who's using libcloud?](#)
- [community resources](#)
- [developer information](#)

Apache Libcloud

Apache Libcloud is a standard Python library that abstracts away differences among multiple cloud provider APIs.

The current version allows users to manage four different cloud resources:

- Cloud Servers - services such as Amazon EC2 and Rackspace CloudServers (`libcloud.compute.*`)
- Cloud Storage - services such as Amazon S3 and Rackspace CloudFiles (`libcloud.storage.*`)
- Load Balancers as a Service, LBaaS (`libcloud.loadbalancer.*`)
- DNS as a Service, DNSaaS (`libcloud.dns.*`)



Apache Libcloud Python libr... x

libcloud.apache.org/supported_providers.html

Apache Libcloud a unified interface to the cloud

Supported Providers and Features

- [home](#)
- [news](#)
- [about](#)
- [getting started](#)
- [documentation](#)
- [who's using libcloud?](#)
- [community resources](#)
- [developer information](#)
- [google summer of code](#)
- [contributing](#)
- [downloads](#)

Compute

provider	list	reboot	create	destroy	images	sizes	deploy
Bluebox	yes	yes	yes	yes	yes	yes	no
Brightbox	yes	no	yes	yes	yes	yes	no
CloudSigma	yes	yes	yes	yes	yes	yes	no
Dreamhost	yes	yes	yes	yes	yes	yes	no
EC2-AP Northeast	yes	yes	yes	yes	yes	yes	yes
EC2-AP Southeast	yes	yes	yes	yes	yes	yes	yes
EC2-US East	yes	yes	yes	yes	yes	yes	yes
EC2-US East 2 (Oregon)	yes	yes	yes	yes	yes	yes	yes
EC2-US West	yes	yes	yes	yes	yes	yes	yes
EC2-EU							

Get it

Latest stable: [0.9.1](#) (April 1st, 2012)
Includes new DNS API, support for Python 3 and more!

デモ用サーバ [Jenkins]
← → ↻ ↕ 🔍

172.20.1.10:8000/view/デモ用サーバ/
☆ 📧 🔄 📄 🔍

Jenkins

検索

Jenkins > デモ用サーバ
自動リロード on

- 新規ジョブ作成
- 開発者
- ビルド履歴
- ビューの変更
- ビューの削除
- プロジェクト相関関係
- ファイル指紋チェック
- Jenkinsの管理

デモ用サーバの操作

ニフテクラウド上に構築された仮想マシンです。使わない時は終了しましょう。

基本的な操作手順は以下のとおりです：

- 「デモ用サーバの起動」でまず起動してください。数分かかりますが、左側に表示されている [www.sakura.ne.jp](#) がオンラインになるはずですが。
- 「デモ用サーバで差分を適用」を実行するだけで、「[www.sakura.ne.jp](#) v1.02-01」の成果物である ISO イメージが転送され、デプロイされます。勝手に再起動したりします。

以上で使えるようになるはずですが。

デモ用サーバにアクセスするには以下の URL へ

- <http://www.sakura.ne.jp/~sakura/aij/>
- <http://www.sakura.ne.jp/~sakura/aij/v1.02-01/>
- <http://www.sakura.ne.jp/~sakura/aij/v1.02-01/iso/>

ビルドキュー
 なし

ビルド実行状態

#	master	
1	待機中	cmmci1
1	待機中	cmmci2
1	待機中	www.sakura.ne.jp
1	待機中	www.sakura.ne.jp

all components install-media migration selenium-test さらVPS デモ用サーバ 開発用サーバ +

S	W	名前 ↓	最新の成功ビルド	最新の失敗ビルド	ビルド所要時間
		デモ用サーバで差分を適用	3ヶ月前 (#30)	3ヶ月前 (#29)	5分 22秒
		デモ用サーバの終了	15時間前 (#241)	1ヶ月前 (#185)	3分 16秒
		デモ用サーバの起動	1時間 53分前 (#194)	3ヶ月前 (#118)	3分 14秒

アイコン: [S](#) [M](#) [L](#)

[凡例](#)
[RSS 全ビルド](#)
[RSS 失敗ビルド](#)
[RSS 最新ビルドのみ](#)

Help us localize this page
更新: 2012/04/19 10:55:03
Jenkins ver. 1.444

さくらのクラウド

```
Java - cc4j/src/sakura.groovy - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
sakura.groovy x
cc4j > src > (default package) > sakura > 1 >
#!/usr/bin/env groovy
import groovy.json.*

Properties sakura = new Properties()
new File('sakura.properties').withInputStream { sakura.load(it) }

String server = sakura.getProperty('server')
String apikey = sakura.getProperty('apikey')
String secret = sakura.getProperty('secret')

Authenticator.default = new Authenticator() {
    public PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication (apikey, secret.toCharArray())
    }
}

def text = new URL(server).getText()
println JsonOutput.prettyPrint(text)

def root = new JsonSlurper().parseText(text)
println "Total=${root.Total}"
```

```
Java - Eclipse
File Edit Navigate Search Project Run Window Help

Console X
<terminated> sakura [Groovy Script] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (2012/04/19 23:04:12)
{
  "From": 0,
  "Count": 1,
  "Total": 1,
  "Servers": [
    {
      "Index": 0,
      "ID": "112400194737",
      "Name": "centos5",
      "HostName": "centos5",
      "Description": "適当に起動してみただけのサーバ。",
      "CreatedAt": "2012-04-19T23:00:15+09:00",
      "Icon": {
        "ID": "112300511981",
        "URL": "https://secure.sakura.ad.jp/cloud/api/cloud/0.2/icon/112300511981.png",
        "Name": "CentOS",
        "Scope": "shared"
      },
      "ServerPlan": {
        "ID": 1,
        "Name": "プラン1",
        "CPU": 1,
        "MemoryMB": 2048,
        "ServiceClass": "cloud/plan/1",
        "Availability": "available"
      },
      "Zone": {
        "ID": 31001,
        "Name": "is1a",
        "Location": "Tokyo"
      }
    }
  ]
}
```

```
Java - cc4j/src/nifty.groovy - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
nifty.groovy x
cc4j src (default package) nifty.groovy cc4j/src/nifty.groovy
#!/usr/bin/env groovy
import groovy.inspect.*
import com.nifty.cloud.sdk.*
import com.nifty.cloud.sdk.auth.*
import com.nifty.cloud.sdk.server.*
import com.nifty.cloud.sdk.server.model.*

Properties nifty = new Properties()
new File('nifty.properties').withInputStream { nifty.load(it) }

String accessKey = nifty.getProperty('accessKey')
String secretKey = nifty.getProperty('secretKey')

Credentials credential = new BasicCredentials(accessKey, secretKey)
NiftyServerClient client = new NiftyServerClient(credential)

DescribeInstancesRequest request = new DescribeInstancesRequest()
DescribeInstancesResult result = client.describeInstances(request)

new Inspector(result).getPropertyInfo().each { println it }
```

```
Java - Eclipse
File Edit Navigate Search Project Run Window Help
Console X
<terminated> nifty [Groovy Script] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (2012/04/19 23:32:39)
[GROOVY, public, n/a, Class, class, class com.nifty.cloud.sdk.server.model.DescribeInstancesRes
[GROOVY, public, n/a, String, requestId, '024f1a64-2346-49e1-ad92-6de14115ab02']
[GROOVY, public, n/a, String, responseXml, '<?xml version="1.0" encoding="UTF-8" standalone="y
[GROOVY, public, n/a, int, status, 200]
[GROOVY, public, n/a, String, proxy, null]
[GROOVY, public, n/a, String, requestHeader, 'User-Agent: NIFTY Cloud API Java SDK/1.9\nHost: c
[GROOVY, public, n/a, List, reservations, [[reservationId=, ownerId=, requesterId=null, groupIc
[GROOVY, public, n/a, String, userAgent, 'NIFTY Cloud API Java SDK/1.9']
[GROOVY, public, n/a, String, reseponseHeader, 'Date: Thu, 19 Apr 2012 14:32:34 GMT\nContent-Ty
[GROOVY, public, n/a, String, queryString, 'AccessKeyId=AKIAJ56QSTGGYCTM0VSTMF&Action=DescribeIns
[GROOVY, public, n/a, String, statusText, 'OK']
[GROOVY, public, n/a, String, url, 'https://cp.cloud.nifty.com/api/']
```



```
Java - Eclipse
File Edit Navigate Search Project Run Window Help

Console X
<terminated> nifty_soap [Groovy Script] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (2012/04/19 23:41:11)
2012/04/19 23:41:14 jp.co.canon_js.soap.LogHandler inbound
情報: <?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <DescribeInstancesResponse xmlns="https://cp.cloud.nifty.com/api/1.3/">
      <requestId>cc1deee0-12b3-4221-8bf6-5a4937aa40b6</requestId>
      <reservationSet>
        <item>
          <reservationId/>
          <ownerId/>
          <groupSet/>
          <instancesSet>
            <item>
              <instanceId>cmmnf</instanceId>
              <imageId>CentOS 5.3 32bit Plain</imageId>
              <instanceState>
                <code>16</code>
                <name>running</name>
              </instanceState>
              <privateDnsName><del>XXXXXXXXXX</del></privateDnsName>
              <dnsName><del>XXXXXXXXXX</del></dnsName>
              <reason/>
              <keyName><del>XXXXXXXXXX</del></keyName>
              <amiLaunchIndex/>
              <productCodes>
                <item>
                  <productCode/>
                </item>
              </productCodes>
            </item>
          </instancesSet>
        </item>
      </reservationSet>
    </DescribeInstancesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Cloud(n)

The image displays a multi-layered screenshot of a cloud management interface. The background is a web browser window showing the 'Product Portal' for 'Cloud(n)'. The URL is https://myportal2.securesites.com/product-portal/?locale=ja_JP. The page features a navigation menu with buttons for 'ダッシュボード', 'ユーザー管理', 'ドメイン管理', '利用アラート管理', '料金カタログ', '請求予定額', and 'クラウドコンソール'. Below this, a '請求予定額' (Billing Forecast) section is visible.

Overlaid on this is a 'Cloud Console - Google Chrome' window. It shows a login page for 'mycloud2.securesites.com' with a command line: `https://mycloud2.securesites.com/client/?command=login&username=Administrator1&domainid=461×tamp=1334847110794&signature=43RPZQcrUxham12dpw`. The console displays details for a virtual machine named 'i-496-1271-VM (uemuraj-win1)'. The details include:

- ID: 1271
- Zone: zone 0
- 名前: i-496-1271-VM
- ハイパーバイザ: KVM
- テンプレート: Windows Serv
- OSタイプ: Windows Serv
- 仮想サーバープラン: Plan vQ
- HA有効: Yes
- ホスト:
- アタッチされたISO: No
- グループ: Windows
- ドメイン: [redacted]
- アカウント: [redacted]
- 作成日付: 19 Apr 2012 14

In the foreground, a Windows desktop environment is shown. A '日付と時刻' (Date and Time) dialog box is open, displaying the current date and time: 2012年4月19日 (April 19, 2012) at 15:05:12. The time zone is set to '(UTC) モンロピア、レイキャビク' (UTC) Monrovia, Reykjavik. The dialog box includes options to change the date and time, and the time zone. The desktop taskbar shows the 'スタート' (Start) button and system tray icons for network, volume, and power. The system clock in the bottom right corner shows 15:05 on 2012/04/19.

Cloud(n)

```
Java - Eclipse
File Edit Navigate Search Project Run Window Help

Console X
<terminated> cloudn [Groovy Script] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (2012/04/20 0:14:57)
Constructing API call to host = 'https://mycloud2.securesites.com/client/api' with API command
Sorted Parameters: [apikey=2448021142...
sorted URL : apikey=2448021142...
final URL : https://mycloud2.securesites.com/client/api?command=listVirtualMachines&response=json
{
  "listvirtualmachinesresponse": {
    "count": 1,
    "virtualmachine": [
      {
        "id": 1271,
        "name": "i-100-1271-VM",
        "displayname": "uemuraj-win1",
        "account": "uemuraj",
        "domainid": "100",
        "domain": "uemuraj",
        "created": "2012-04-19T10:54:15-0400",
        "state": "Running",
        "haenable": true,
        "groupid": "100",
        "group": "Windows",
        "zoneid": 1,
        "zonename": "zone 0",
        "templateid": 293,
        "templatename": "Windows Server 2008 R2 64-bit (Japanese)",
        "templatedisplaytext": "Windows Server 2008 R2 64-bit (Japanese) *charged",
        "passwordenabled": true,
        "serviceofferingid": 53,
        "serviceofferingname": "Plan vQ",
        "cpunumber": 1,
        "memory": 1024
      }
    ]
  }
}
```

- APIレベルの互換性は何にせよ実現するように思われます
- 当面の問題は解決しそうです
- 本格的な業務となると、ここから先にまだ考慮することあり…
- 楽をするならやはりIaaSでなくPaaSのレベルかと思えます
- Cloud FoundryやJava EE7に期待しています