From 1 RPM to 10,000 RPM

ALM in the As-A-Service Age

Steven G. Harris, SVP Products
@stevengharris
steven.g.harris@cloudbees.com
The Problem

• I know how to deliver “packaged software”, but I’m moving to an as-a-service model or mobile.
  – Is it really that different?
  – How should I change my processes?

• I’m delivering functionality as a hosted service.
  – How can I improve and fine-tune what I’m offering?
  – How can I connect more intelligently with my users?

• When I deliver everything as a service and consume everything as a service, what kind of things should the “platform” be doing for me?
CloudBees Is a Platform, Delivered As A Service

It’s just Software as a Service. But the software is your *platform* to create, deploy, and manage applications.
What You Interact With When You Use CloudBees

- CLI/SDK
- GrandCentral Web Console
- IDE (eclipse, IntelliJ IDEA, Codenvy)
What We Deliver Continuously

Grand Central Web Console
CloudBees API
Forge Repositories
Git
SVN
mvn

Test
Jenkins
Master
Stage
Build

Code
Jenkins
Executor

Application
Session Clustering
Router
MySQL

Partner Ecosystem

Session
Clustering

Identity
Provisioning
Monitoring
Auditing

AnyCloud Message Bus
A5 Agent
DB Agent
Agents

AnyCloud

Shared Services

©2013 CloudBees, Inc. All rights reserved
We Use... Jenkins in the Cloud!

![Jenkins Dashboard](https://example.com/jenkins-dashboard.png)

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Last Success</th>
<th>Last Failure</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudbees-security-groups</td>
<td>10 min</td>
<td>4 days</td>
<td>58 sec</td>
</tr>
<tr>
<td>Internal Maven Repository</td>
<td>1 hr 33 min</td>
<td>1 mo 19 days</td>
<td>1.5 sec</td>
</tr>
<tr>
<td>base-ci-server-plugin</td>
<td>18 hr</td>
<td>N/A</td>
<td>4 min 42 sec</td>
</tr>
<tr>
<td>base-ci-plugs</td>
<td>18 hr</td>
<td>1 mo 15 days</td>
<td>6 min 54 sec</td>
</tr>
<tr>
<td>base-ci-plugin</td>
<td>10 hr</td>
<td>6 days 17 hr</td>
<td>8 min 6 sec</td>
</tr>
<tr>
<td>Send RUN status by email</td>
<td>18 hr</td>
<td>N/A</td>
<td>6.5 sec</td>
</tr>
<tr>
<td>base-ci</td>
<td>23 hr</td>
<td>6 hr 48 min</td>
<td>1 min 54 sec</td>
</tr>
<tr>
<td>Nexusci provider</td>
<td>1 day 1 hr</td>
<td>N/A</td>
<td>4 min 29 sec</td>
</tr>
<tr>
<td>CloudBees RUN Analytics</td>
<td>1 day 1 hr</td>
<td>N/A</td>
<td>5 min 13 sec</td>
</tr>
<tr>
<td>run-hadoop</td>
<td>1 day 2 hr</td>
<td>7 days 14 hr</td>
<td>14 min</td>
</tr>
<tr>
<td>distributed-tests</td>
<td>1 day 2 hr</td>
<td>23 days</td>
<td>4 min 1 sec</td>
</tr>
<tr>
<td>cloudbees-server-plugin</td>
<td>1 day 13 hr</td>
<td>N/A</td>
<td>1 min 25 sec</td>
</tr>
<tr>
<td>base-ci-conf-gui-plugin</td>
<td>1 day 18 hr</td>
<td>18 hr</td>
<td>1 min 52 sec</td>
</tr>
<tr>
<td>base-ci-di-plugin</td>
<td>1 day 18 hr</td>
<td>18 hr</td>
<td>1 min 47 sec</td>
</tr>
<tr>
<td>base-ci-app-plugin</td>
<td>1 day 18 hr</td>
<td>18 hr</td>
<td>4 min 48 sec</td>
</tr>
<tr>
<td>jenkins-service-provider</td>
<td>1 day 20 hr</td>
<td>1 day 1 hr</td>
<td>7 min 2 sec</td>
</tr>
</tbody>
</table>
Continuous Cloud Delivery at CloudBees

Unit Tests → Cross Service Tests → Stage → UAT → Prod

As needed, multiple times per day

Staging

Production
Cross-Service Tests and Promoted Builds

API Service
- API Tests
- OAuth Tests
- SDK Tests
- Deployer Tests
- Cloud Foundry
- App Engine

Promote

Web Console
- Partner Embeds
- RUN Console
- DEV Console
Continuous Cloud Delivery at CloudBees

GrandCentral Release
SDK

Identity
Provisioning
Monitoring
Messaging
Services Platform

Unit Tests
Cross Service Tests
Stage
UAT
Prod

As needed, every couple of days

Staging
Production

©2013 CloudBees, Inc. All rights reserved
What’s So Different About As-A-Service?
“Mental picture” of most of today’s software vendors

We use SaaS extensively. It makes us sooooo productive...

... at delivering packaged software to our customers!

#FAIL

©2013 CloudBees, Inc. All rights reserved
Do Unto Others...

• Your customers are no different than you
• They don’t like...
  – Shelf-ware, setup, patches, upgrading releases, etc.
• They do like...
  – Fast time-to-market, pay-as-you-go, elasticity, mobility, etc.
Drivers & Acceleration

• Drivers are accelerating this transition to Cloud-based services:
  – Mobile, Mobile, Mobile, Social, Big Data, Etc.

• Time-to-market is more important than ever
  – “I need it yesterday!”
  – Can’t afford 18m cycles
  – Discovery-mode
    • i.e., Don’t spend 18m building what I am not sure I really need

• And SaaS is already pulling more data into the cloud
  – Slowly redefines what data is “remote”
  – On-premise is the new legacy
Software vs. Cloud Service

- A cloud service is **not** just “hosted software”
  - i.e., take your current software, host it, and keep doing what you’ve been doing to date
- Delivering as a service affects…
  - The way you create, test, and deploy software
  - The opportunities you have to experiment cheaply
  - The cost of developing and maintaining software
  - The “short cuts” you can take compared to packaged software
  - The way you engage with your customers
  - The quality of information you have about your customers
To Be Or Not To Be...

Packaged Software

- Release v1.0
- Gather requirements for v2.0
- Work on v2.0
- Release v2.0
  - don’t screw it up: once it is out, it is out.
- Have your customer upgrade
  - Support v1.0 for years
- Support and patch dozen of “branches” in parallel
- Gather requirements for v3.0
- Work on v3.0

As-A-Service

- Release v0.1
- Implement micro-feature, deploy, measure success, keep or kill
- Implement micro-feature, deploy, measure success, keep or kill
- Patch a bug
- Implement micro-feature, deploy, measure success, keep or kill
- Implement micro-feature, deploy, measure success, keep or kill
- Implement micro-feature, deploy, measure success, keep or kill
Lower Risk and Cheap Experimentation

Packaged Software

As-A-Service

Risk/Cost

Time

v1.0 Release

v1.0.x Continuous Releases

v2.0 Release

Risk/Cost

Time

Same basic value prop as agile, but code is being released continuously

©2013 CloudBees, Inc. All rights reserved
Use Case: Lifecycle Marketing

Marketing!? Wait, aren’t I at EclipseCon?
Lifecycle Marketing in the Packaged Software World

- Lead
- Nurturing
- Consideration
- Sale
- Post Sale

Most systems in house: Email marketing, customer lists

They noticed us!
This is why we are special
Seems like they might buy
Sold!
Upsell!
Fairly static customer lists

Marketing

©2013 CloudBees, Inc. All rights reserved
Lifecycle Marketing in the Cloud Era ( Freemium Model )

Lead
Nurturing
Consideration
Free Tier
Base Tier
Add ons
Enterprise Tier
Post Sale

Marketing
Evangelist
“Growth Hacker”
Customer Success Manager

Short cycle, as users can try the free tier and make the determination on product feasibility themselves.

Users can move freely up and down the tiers.

Most systems external: Hosted, best-of-breed services.
Continuous Delivery

What does that have to do with marketing?
Packaged Software vs Cloud Software Delivery

- Development Methodology Varies
- Bonus: “We use CI”

- Version 1
  - Shipped May 2011

- Version 2
  - Shipped May 2012

- Version 3
  - Shipped May 2013

- Unachievable without Continuous Integration and Continuous Delivery

- Update CSS
  - 1st Feb

- Move button
  - 1st Feb

- New tab
  - 2nd Feb

- Overhaul UI
  - 13th Feb

- Bug fix
  - 14th Feb
Capturing Events of Interest at CloudBees

**Subscribed to a service**
- Jenkins
- RUN@cloud (Deployment PaaS)
- Forge (Git, SVN)
- DB service
- Partner service
  - MongoHQ
  - NewRelic
  - PaperTrail
  - ...

**Performed an action**
- Web UI login
- Jenkins
  - Created a Job, Installed a plugin, Ran a job...
- Deployment PaaS
  - Deployed an app, restarted app, app stopped, app hibernated...
- Others
  - Canceled account, added a paid DB, added a user
Continuous Integration, Delivery and... Engagement

• Continuous improvements on events
  – Continuously add new “finer” grained events based on usage feedback
• Rapid deployments of new events using Continuous Deployment
Details – User Engagement
Traditional Lifecycle Marketing

Product Signup

Tip of the Week for ‘x’ weeks
As-A-Service LifeCycle Marketing
Tracking “For Free” User Engagement Stages

- Time and activity based
- Grade users based on risk profile
- Separate onboarding from established users
- Offer hands-on help for established users

<table>
<thead>
<tr>
<th>Time</th>
<th>Sign-Up</th>
<th>Eden</th>
<th>Tenured</th>
<th>OldSpace</th>
<th>PermSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 wk</td>
<td>Use? Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>2 wks</td>
<td>Use? Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>3 wks</td>
<td>Use? Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>4 wks</td>
<td>Use? Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

Onboarding

Established
As-A-Service LifeCycle Marketing
Keeping Paying Users Happy

- Track number and types of activity over time
  - Scale and adjust “smart” communication over time
- Users can become inactive and then reactivate
- Technical, not sales communication
Using Other Cloud Services

The dance to make all this happen
Segmenting Users for Engagement

1. Deliver new event to prod
2. Log events during usage
3. Set up appropriate active list
4. Sync to SF
5. Pull into “smart lists”
6. Communicate w/user
7. Monitor effectiveness
8. Go to 1
Getting Useful Information To Your Users

1. Deliver new event to prod
2. Log events during usage
3. Set up appropriate active list
4. Pull list
5. Massage data
6. Generate usage graphs
7. Expose as web service
8. Pull into “smart lists”
9. Insert “webhooks”
10. Communicate w/ user
11. Monitor effectiveness
12. Go to 1
John Doe used “CloudBees Deployer” to deploy app from Jenkins on CloudBees to RUN@cloud

```java
public void deploymentSuccess(String scope, String account, String email) {
    track(account, email, "Deployment successful", scope);
}

private synchronized void track(String org, String user, String activity, String module) {
    Request request = new RequestBuilder("GET")
        .setUrl("http://sdr.totango.com/pixel.gif/")
        .addQueryParameter("sdr_s", key)
        .addQueryParameter("sdr_o", org)
        .addQueryParameter("sdr_u", user)
        .addQueryParameter("sdr_a", activity)
        .addQueryParameter("sdr_m", module).build();
    Singletons.asyncHttpClient(context).executeRequest(request);
}
```

Track: http://sdr.totango.com/pixel.gif/?sdr_s=<Key>@sdr_o=<Org>&sdr_u=<User>&sdr=<Activity>&sdr_m=<Module>
Lists to Track Activity

Collect: Weekly-active: All users who have used any module in the last week

Filter Conditions:

- Used module: 7 selected
  - Select all
  - Unselect all
  - AppEngine Gateway
  - DEV@ Forge
  - DEV@ Jenkins
  - GrandCentral
  - GrandCentralAPI
  - Jenkins Enterprise
  - RUN@cloud

Activate list: When you Activate the list, Totango will periodically evaluate your account.

Entry Trigger: Perform these actions whenever an account matches the list.

Add tag: active_this_week
Setting Up Campaigns, Service Integrations

TOTANGO

©2013 CloudBees, Inc. All rights reserved
Customized Processing of Lists

**Question:** What were active users doing on the platform last week?

**Access:**
```
(def totango-api-endpoint "https://app.totango.com/api/v1/accounts/active_list/.../current.json")
```

**Pull Raw Data**
```
defn totango-get-data [site token params]
  (let [user-data
    (client/get site
     {:headers {"Authorization" token}
     :query-params params
     :as :json-string-keys})]
    (user-data :body)))
```

**Extract Map**
```
; Get a map account->list of non-CB-emails
(defn map-account-to-users [accounts]
 ; account = n * users
  (map get-user-account-id accounts))
```
Multiplying Continuous Delivery Benefits
In The As-A-Service World

If only I had access to...
I think we can expose that!

From feature request to usage in CloudBees was 2 weeks!
Not possible without Continuous Delivery.

#FTW
What's up with this new framework!

What's up with this new framework!

Wow!

Wow!

Sampling Exhaust Fumes

cloudbees.com

Partner Service

Bind

App

Bind

Deploy

At your service

What's up with this new framework!

Wow!
Sampling Exhaust Fumes

- What is interesting to users?
- What languages and frameworks are they using?
Data Driven Product Investment

- Popularity of Play! is very interesting
- Strength of traditional Java stacks
  - Mixed with JS
- Strong showing of node
Futures

What’s the meaning of “platform” in an as-a-service world?
Platform Evolution To Cloud Services

http://netflix.github.com/
http://techblog.netflix.com/2013/03/introducing-first-netflixoss-recipe-rss.html

©2013 CloudBees, Inc. All rights reserved
Example – Platform Role in the As-A-Service Age

For each service, you set up:

* .jar => lib
* .properties
  ... more ...

And use them by writing code like this:

```java
ms1 = MyService1Factory(...);
ms1.auth(token);
/* ... more service-specific boilerplate ... */
ms1.doSomethingInteresting(...);
result1 = ms1.getResult(...);

ms2 = MyService2Factory(...);
ms2.setCredentials(key, secret);
/* ... more service-specific boilerplate ... */
result2 = ms2.getStuff(result1, ...);

/* do something useful */
myAppResult = Mashup.things(result1, results);
```

But the platform knows...
- OAuth2 tokens
- keys, secrets
- Service-specific setup
So why not let the platform do the work?

```java
@Inject MyService1 ms1;
ms1.doSomethingInteresting(...);
result1 = ms1.getResult(...);

@Inject MyService2 ms2;
result2 = ms2.getStuff(result1, ...);

/* do something useful */
myAppResult = Mashup.things(result1,result2);
```

Platform can handle...
- Service-specific setup
- Secure connections
- Wiring with app, like datasources
Conclusions

• Creating and delivering software as a service is fundamentally different than packaged software
• Continuous delivery, preferably in the cloud, is a necessity to match the as-a-service model
• Realtime feedback, driven by developers, is key to improving user experience and product investments
• The platform’s role is changing in the as-a-service world
Thank You!

steven.g.harris@cloudbees.com
@stevengharris

#cdcloud