Infrastructure in the Cloud Era

Adam Jacob - Co-founder & CTO, Opscode
Ezra Zygmuntowicz - Co-Founder & Mad Scientist Engine Yard

Opscode
Please Join our Ad-Hoc Cloud

If you have a Mac with Developer Tools installed...

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
What are you guys talking about?

- Bootstrapping
- Configuration
- Command and Control

```bash
curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
```
Many Different Tools

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
Bootstrapping

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
## Bootstrapping Approaches

<table>
<thead>
<tr>
<th></th>
<th>Good</th>
<th>Bad</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corp Approvals</strong></td>
<td>Known Costs, No Variation. Anything you want, as long as IT pre-approved it.</td>
<td>High Waste (Hoarding) Red Tape Expensive ($/Time) Long lead time</td>
<td>6-8w</td>
</tr>
<tr>
<td><strong>Agile Corp Approvals</strong></td>
<td>Known Costs. Total Hardware Control. Trivial Approvals.</td>
<td>Lower Waste Less Red Tape Still slow Expensive ($/Time) Shorter lead time</td>
<td>2-4w</td>
</tr>
<tr>
<td><strong>Cloud</strong></td>
<td>Variable Costs. Highly Adaptable. Minimal lead time. Trivial approvals. No humans needed.</td>
<td>Variable Costs. No control over hardware. Must re-train.</td>
<td>5-10m</td>
</tr>
</tbody>
</table>

```
curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
```
Why Time Matters

Total Bootstrapping Time in Weeks

- Corp Approvals
- Agile Approvals
- Cloud

Best Time

Worst Time

Wednesday, April 8, 2009

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
Tools for Bootstrapping

PXE Boot
FAI
System Imager
Kickstart
Cobbler

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
Bootstrapping Demo

Engine Yard Solo

Now would be a totally sweet time to pull out your Mac and run...

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh

Wednesday, April 8, 2009
curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
## Configuration Approaches

<table>
<thead>
<tr>
<th></th>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Manual</strong></td>
<td>You can do anything. Results in an intimate knowledge of the details.</td>
<td>Slow. Error Prone (Bus Error!) Non-repeatable. Difficult knowledge transfer.</td>
</tr>
</tbody>
</table>

```bash
curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
```
Tools for Configuration

**Chef, Puppet, Cfengine, Bcfg2**

**Shell Scripts, Perl, Python, Ruby, etc.**

**Operating System tools**
Your favorite text editor

```bash
curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
```
Infrastructure as Code

Manage configuration as idempotent Resources.
Put them together in Recipes.
Track it like source code.
Configure your servers.

You can learn more about Chef at http://wiki.opscode.com/display/chef/Home

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
You need an API

You can learn more about Chef at http://wiki.opscode.com/display/chef/Home

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
Command and Control

You are running out of time to join the Ad-Hoc Cloud Party!

curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
# Command and Control

<table>
<thead>
<tr>
<th></th>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
</table>
| **Meatcloud** | Super flexible.  
Can do almost anything.  
Always easy to find someone to blame.  
Free will.         | Error Prone.  
Slow.  
Expensive to Scale.  
Not repeatable.  
Free will.         |
| **Ad-Hoc**    | More repeatable.  
Easier to scale.  
Less error prone (hopefully!) | One-off by necessity.  
Tooling sprawl.  
Hard to share solutions.  
Much higher learning curve. |
| **Framework** | One system to learn.  
Scales well.  
Paint by numbers.  
Repeatable.  
Trades depth of knowledge for ease of use. |

*Meatcloud appears in this presentation courtesy of Andrew Shafer - [http://is.gd/Ega](http://is.gd/Ega)*

**Last Chance!** - curl -O [https://cloud.engineyard.com/w2e.sh](https://cloud.engineyard.com/w2e.sh) && sh w2e.sh
Tools for Command and Control

Nanite, Capistrano, Control Tier

Shell Scripts, Perl, Python, Ruby, etc.

You
Your co-workers

Last Chance! - curl -O https://cloud.engineyard.com/w2e.sh && sh w2e.sh
You can learn more about Nanite at http://github.com/ezmobius/nanite/tree/master
What else can you do?

- Application Deployment
- Starting and stopping services
- Maintenance
- You name it.
1. Bring on capacity as traffic ramps up
2. Take down capacity as it ramps down
3. 10-15 Minutes on either side, fully unattended

Graphs in this portion of the presentation taken from Theo Schlossnagle
http://omniti.com/seeds/dissecting-todays-internet-traffic-spikes
Atypical Load

1. Hope you know it is coming.
2. Increase capacity in advance.
3. Take down capacity as it ramps down.

However, you are still better off!

No way around Capacity Planning

Graphs in this portion of the presentation taken from Theo Schlossnagle
http://omniti.com/seeds/dissecting-todays-internet-traffic-spikes

Wednesday, April 8, 2009
1. Signal Command and Control
2. Analyze Responses
3. Rinse and Repeat
4. Profit
1. Your infrastructure is disposable.
2. Make sure you can turn each layer off.
3. New efficiencies often lead to new hard problems.
Q & A

Brought to you by...

Opscode

You can learn more about Chef at http://wiki.opscode.com/display/chef/Home
You can learn more about Nanite at http://github.com/ezmobius/nanite/tree/master